HP 64000
Software Status Bulletin

Part Number: 5958-6018
E0289

PRINTED IN U.S.A. FEBRUARY 1989

HP STARS II

64000 SOFTWARE STATUS BULLETIN

Logic Systems Division

Issue 2906.89.04A * ALL PRODUCTS

APRIL, 1989

This document supersedes all previously dated SSBs.

HEWLETT
PACKARD

READER COMMENT SHEET

STARS II SSB (STARS B)

Issue _____._____ DATE _____/_____/_____

We welcome your evaluation of this bulletin. Your comments and suggestions help us to improve our publications. Please use additional pages if necessary.

Is this bulletin technically accurate?	Yes [ ] No [ ]	(If no, explain under Comments, below.)
Are the concepts and wording easy to understand?	Yes [ ] No [ ]	(If no, explain under Comments, below.)
Is the format of this bulletin convenient in size, arrangement and readability?	Yes [ ] No [ ]	(If no, explain or suggest improvements under Comments, below.)

Comments:

****

Date: _____

FROM:

Name _____

Company _____

Address _____

_____

_____

STARS B
Printed in U.S.A.

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, CO.

POSTAGE WILL BE PAID BY ADDRESSEE

STARS Administration
Hewlett-Packard Company
Logic Systems Division (LSD)
P.O. Box 617
Colorado Springs, Colorado. 80901-0617

FOLD FOLD

P R E F A C E

This Software Status Bulletin (SSB) documents all known problems in the software product line designated on the cover page. The SSB is derived from Known Problem Reports (KPR) which result from Service Requests (SR) submitted by users of these products. The SSB is provided as a benefit of Hewlett-Packard's Account Management Support, Response Center Support, Software Materials Subscription, and Software Notification Service.

Not all SR's submitted to HP are listed in the SSB. Ones which involve problems that cannot be duplicated, requests for enhancements and misunderstandings about an application or a feature are not listed in the SSB. SR's which refer to a previously documented problem are cross referenced within the report which originally identified the problem. Every SR verified by an HP Systems Engineer and sent to the factory is assigned a unique identifier and acknowledged by letter to the submitter. When the SR is classified as a documentation problem or a software design fault, a KPR is written for it. The KPR then appears in the next issue of the SSB. After the problem has been corrected and signed off by Product Assurance, the fact is noted in the KPR with the following statement: "Date fix signed off : mm/dd/yy Rel: uu.ff.". When a new software release is made for the product line, all problems that were corrected in that release are reported in the Software Release Bulletin for that release and the KPR's are removed from the SSB.

The SSB is distributed in complete form once every calendar quarter. Between quarterly issues, monthly issues containing only those problems documented since the last SSB issue are distributed. This means, that to have a complete list of all outstanding problems you must have the last quarterly issue and all monthly issues since that quarterly.

Of the five sections contained in the SSB, only the last (known problem reports) has page numbers. The product, KPR number and keyword indices all reference these page numbers to direct the user to a particular area or individual detailed report. The five sections are described below:

SOFTWARE RELEASE CONTENTS

This section gives you the release ID of the current software release and the previous software release for the product line. Additionally, the current and previous update/fix levels are given for each product contained in the product line.

PRODUCT INDEX

The monthly issues have one product index. The quarterly issue has two; the first referencing the problems that were documented since the last monthly issue, and the second referencing problems that were reported in a previous issue of the SSB. Each unique product name/number has an entry listing the page number where the KPR's against that product begin.

KPR NUMBER INDEX

A sequential list of KPR numbers with the corresponding page number where the KPR can be found.

KEYWORD INDEX

This index is sorted by product name, keyword, product number (including update/fix level) and by KPR number in that order. Along with the sort items, each entry has a brief (72 character) description and the page number where the KPR can be found. In the quarterly issue, entries that are new since the last update are denoted by an asterisk (*) immediately following the KPR number.

KNOWN PROBLEM REPORTS

Each report contains all the available information relevant to the problem.

Product name	Product number	uu.ff	prev
1000L ASSEMB	64852	01.00	00.00
1802 ASSEMB	64848	01.00	00.00
2A03 EMUL	64209	01.00	00.00
6301V EMULATION 300	64206S004	01.20	01.10
6301V/03R EMUL	64206	01.01	00.00
6301X EMULATION 300	64207S004	01.10	01.00
6301X/03X EMUL	64207	01.00	00.00
6301Y EMULATION 300	64208S004	01.10	01.00
6301Y/03Y EMUL	64208	01.00	00.00
64000 UX GENERIC	64003S001	01.00	00.00
64000 UX GENERIC	64003S004	01.00	00.00
64000-UX OP-ENV 300	64801S004	02.60	02.40
64180 ASSEMB	64864	01.00	00.00
64180 ASSEMB 300	64864S004	01.01	01.01
64180 EMUL	64180	01.10	01.00
64180 EMUL	64180S004	01.00	00.00
64340 P.V.	64340	01.00	00.00
650X ASSEMB	64843	01.80	00.00
650X ASSEMB 300	64843S004	01.80	00.00
650X ASSEMB 500	64843S001	01.80	01.20
650X ASSEMB DOS	64843S006	01.80	00.00
650X ASSEMB VAX	64843S003	01.90	01.80
6800 C	64821	02.20	02.10
6800 C 300	64821S004	02.20	02.10
6800 C 500	64821S001	02.20	02.10
6800 C VAX	64821S003	02.30	02.20
6800 EMULATION	64212	01.05	00.56
6800 INTERFACE 300	64672S004	01.00	00.00
6800 PASCAL	64811	02.00	01.90
6800 PASCAL 300	64811S004	02.00	01.90
6800 PASCAL 500	64811S001	02.00	01.90
6800 PASCAL VAX	64811S003	02.10	02.00
6800 PRE-PROCESSOR	64672	00.53	00.00
6800/2 ASSEMB	64841	01.80	01.14
6800/2 ASSEMB 300	64841S004	01.80	01.00
6800/2 ASSEMB 500	64841S001	01.80	01.30
6800/2 ASSEMB DOS	64841S006	01.80	00.00
6800/2 ASSEMB VAX	64841S003	01.90	01.80
68000 12MHZ EMUL FW	64742	00.05	00.04
68000 12MHZ FUI DOS	64742S006	01.02	01.01
68000 12MHZ PTUI 300	64742S004	01.00	00.00
68000 16MHZ EMUL FW	64743	00.01	00.00
68000 16MHZ FUI DOS	64743S006	00.01	00.00
68000 16MHZ PTUI 300	64743S004	00.01	00.00
68000 ASSEMB	64845	02.20	02.10
68000 ASSEMB 300	64845S004	02.20	02.10
68000 ASSEMB 500	64845S001	02.20	02.10
68000 ASSEMB DOS	64845S006	02.20	02.11
68000 ASSEMB VAX	64845S003	02.20	02.10
68000 BBA 300	64380S004	01.10	01.00
68000 BBA 800	64380S005	01.10	00.00
68000 C	64819	02.20	02.10
68000 C 300	64819S004	02.20	02.10
68000 C 500	64819S001	02.20	02.10
68000 C VAX	64819S003	02.30	02.20

Product name	Product number	uu.ff	prev
68000 DQ EMUL 300	64243S004	01.30	01.20
68000 DQ SW ANAL	64331B	01.02	01.01
68000 DQ SW ANALYZER	64341G	01.03	01.02
68000 EMUL 12.5 MHZ	64243	01.01	01.01
68000 EMULATION	64242	01.07	01.06
68000 EMULATION 300	64242S004	01.00	00.00
68000 INTERFACE 300	64674S004	01.00	00.00
68000 MONITOR	64742-11001	00.04	00.00
68000 PASCAL	64815	02.00	01.90
68000 PASCAL 300	64815S004	02.00	01.90
68000 PASCAL 500	64815S001	02.00	01.90
68000 PASCAL VAX	64815S003	02.10	02.00
68000 PRE-PROCESSOR	64670	01.00	00.56
68000 PRE-PROCESSOR	64674	01.00	00.00
68000 SW ANAL	64331	02.03	02.02
68000 SW ANALYZER	64341B	02.02	02.01
68000/20ASMB CONT300	5010-2766	01.30	01.00
68000/20ASMB CONT800	64870-13801	01.00	00.00
68000C AXLS COMP 300	64902S004	02.10	02.00
68000C AXLS COMP 800	64902S005	02.10	02.00
68008 EMULATION	64244	01.01	01.01
68008 EMULATION 300	64244S004	01.30	01.10
68008 INTERFACE 300	64673S004	01.00	00.00
68008 PRE-PROCESSOR	64673	00.65	00.00
68008 SW ANAL	64337	01.02	01.01
6801/3 EMULATION	64256	01.04	00.70
6801/3 EMULATION 300	64256S004	01.00	00.00
68010 16MHZ EMUL FW	64745	00.01	00.00
68010 16MHZ FUI DOS	64745S006	01.02	01.01
68010 16MHZ PTUI 300	64745S004	01.00	00.00
68010 DQ SW ANAL	64334B	01.02	01.01
68010 DQ SW ANALYZER	64341I	01.02	01.01
68010 EMUL 12.5 MHZ	64245	01.01	01.01
68010 EMUL 12.5M 300	64245S004	01.30	01.20
68010 G.P. EMUL 300	64249S004	01.00	00.00
68010 G.P. EMULATOR	64249	01.02	01.01
68010 SW ANAL	64334	02.03	02.02
68010 SW ANALYZER	64341D	02.02	02.01
6802 EMULATION	64213	01.05	00.56
68020 ASSEMB 300	64870S004	01.00	00.00
68020 ASSEMB 800	64870S005	01.00	00.00
68020 BBA 300	64381S004	01.20	01.10
68020 BBA 800	64381S005	01.20	01.00
68020 EMUL 300	64410S004	02.10	02.00
68020 EMUL 300	64416S004	02.00	01.00
68020 INV ASSEMB	64675	01.00	00.00
68020 INV ASSEMB 300	64675S004	01.00	00.00
68020C AXLS COMP 300	64903S004	02.11	02.10
68020C AXLS COMP 800	64903S005	02.11	02.00
6805 E EMUL 300	64195S004	01.10	01.00
6805 G EMUL	64194	01.07	01.05
6805 G EMUL 300	64194S004	01.10	01.00
6805 P EMUL	64193	01.07	01.05
6805 P EMUL 300	64193S004	01.10	01.00
6805 U&R EMUL	64192	01.07	01.06

Product name	Product number	uu.ff	prev
6805 U&R EMUL 300	64192S004	01.10	01.00
6805/9 ASSEMB	64844	01.90	01.10
6805/9 ASSEMB 300	64844S004	01.90	01.00
6805/9 ASSEMB 500	64844S001	01.90	01.30
6805/9 ASSEMB DOS	64844S006	01.90	00.00
6805/9 ASSEMB VAX	64844S003	02.00	01.90
6809 C	64822	01.90	01.80
6809 C 300	64822S004	01.90	01.80
6809 C 500	64822S001	01.90	01.80
6809 C VAX	64822S003	02.00	01.90
6809 EMULATION	64215	01.08	00.56
6809 EMULATION 300	64215S004	01.10	01.00
6809 PASCAL	64813	01.70	01.60
6809 PASCAL 300	64813S004	01.70	01.60
6809 PASCAL 500	64813S001	01.70	01.60
6809 PASCAL VAX	64813S003	01.80	01.70
6809/E INTERFACE 300	64671S004	01.00	00.00
6809/E PRE-PROCESSOR	64671	00.49	00.00
6809E EMULATION	64216	01.08	00.56
6809E EMULATION 300	64216S004	01.10	01.00
680XX DEBUG/SIM 300	64360S004	01.00	00.00
68HC11 EMUL	64265	01.01	01.00
68HC11 EMUL 300	64265S004	01.10	01.00
68HCII ASSEMB	64865	01.40	01.30
68HCII ASSEMB 300	64865S004	01.40	01.30
68HCII ASSEMB 500	64865S001	01.40	01.30
68HCII ASSEMB DOS	64865S006	01.40	01.30
68HCII ASSEMB VAX	64865S003	01.50	01.40
70108 EMUL	64295	01.00	00.00
70108 EMUL 300	64295S004	01.10	01.00
70108 SW ANAL	64339	01.00	00.00
70108 SW ANALYZER	64342B	01.00	00.00
70116 EMUL	64294	01.00	00.00
70116 EMUL 300	64294S004	01.10	01.00
70116 SW ANAL	64338	01.00	00.00
70116 SW ANALYZER	64342A	01.00	00.00
70208 EMUL	64297	01.00	00.00
70208 EMUL	64297S004	01.00	00.00
70216 EMUL	64296	01.00	00.00
70216 EMUL	64296S004	01.00	00.00
78310/12 ASSEMB	64866	01.02	01.00
78310/12 ASSEMB 300	64866S004	01.00	01.00
80186 EMUL FW	64764	00.02	00.00
80186 FUI DOS	64764S006	01.02	01.01
80186 PTUI 300	64764S004	01.00	01.00
80186 EMULATION	64224	01.05	01.04
80186 EMULATION 300	64224S004	01.30	01.20
80186 INTERFACE 300	64658S004	01.00	00.00
80186 MONITOR	64764-11001	00.01	00.00
80186 PRE-PROCESSOR	64658	00.57	00.00
80186 SW ANAL	64335	02.03	02.02
80186 SW ANALYZER	64341E	02.02	02.01
80188 EMUL FW	64765	00.02	00.00
80188 FUI DOS	64765S006	01.02	01.01
80188 PTUI 300	64765S004	01.00	00.00

Product name	Product number	uu.ff	prev
80188 EMULATION	64225	01.03	01.02
80188 EMULATION 300	64225S004	01.30	01.20
80188 MONITOR	64765-11001	00.01	00.00
80188 SW ANAL	64336	02.04	02.03
80188 SW ANALYZER	64341F	01.02	01.01
80196 EMUL FW	64771	00.02	00.01
80196 FUI DOS	64771S006	01.00	00.00
80196 PTUI 300	64771S004	01.00	00.00
80196 MONITOR	64771-11001	00.01	00.00
80286 EMULATION	64228	01.02	01.01
80286 INTERFACE 300	64657S004	01.00	00.00
80286 PRE-PROCESSOR	64657	00.67	00.00
80286 UDE	64227	01.00	00.00
80286B ASSEMB	64859	01.50	01.40
80286B ASSEMB 300	64859S004	01.50	01.40
80286B ASSEMB 500	64859S001	01.50	01.40
80286B ASSEMB DOS	64859S006	01.50	01.40
80286B ASSEMB VAX	64859S003	01.60	01.50
80386 EMUL	64420S004	01.00	00.00
8048 ASSEMB	64846	01.80	00.00
8048 ASSEMB 300	64846S004	01.80	00.00
8048 ASSEMB 500	64846S001	01.80	01.20
8048 ASSEMB DOS	64846S006	01.80	00.00
8048 ASSEMB VAX	64846S003	01.90	01.80
8048 EMULATION	64262	01.07	01.06
8051 ASSEMB	64855	01.80	01.07
8051 ASSEMB 300	64855S004	01.80	01.10
8051 ASSEMB 500	64855S001	01.80	01.40
8051 ASSEMB DOS	64855S006	01.80	00.00
8051 ASSEMB VAX	64855S003	02.00	01.80
8051 EMULATION	64264	01.04	01.02
8051 EMULATION 300	64264S004	01.00	00.00
8080 EMULATION	64202	01.07	01.06
8080/5 ASSEMB	64840	01.80	00.00
8080/5 ASSEMB 300	64840S004	01.80	00.00
8080/5 ASSEMB 500	64840S001	01.80	01.20
8080/5 ASSEMB DOS	64840S006	01.80	00.00
8080/5 ASSEMB VAX	64840S003	01.90	01.80
8080/5 INTERFACE 300	64655S004	01.00	00.00
8080/5 PRE-PROCESSOR	64655	00.56	00.00
8085 B PASCAL	64825	02.00	01.90
8085 B PASCAL 300	64825S004	02.00	01.90
8085 B PASCAL 500	64825S001	02.00	01.90
8085 B PASCAL VAX	64825S003	02.10	02.00
8085 C	64826	02.20	02.10
8085 C 300	64826S004	02.20	02.10
8085 C 500	64826S001	02.20	02.10
8085 C VAX	64826S003	02.30	02.20
8085 EMULATION	64203	01.08	01.07
8085 EMULATION 300	64203S004	01.40	01.30
8085 PASCAL	64810	00.70	00.00
8086 EMUL FW	64762	00.02	00.00
8086 FUI DOS	64762S006	01.02	00.01
8086 PTUI 300	64762S004	01.00	00.00
8086 ASSEMB 300	64871S004	01.00	00.00

Product name	Product number	uu.ff	prev
8086 BBA 300	64382S004	01.00	01.10
8086 MONITOR	64762-11001	00.02	00.00
8086 DQ EMUL 300	64220S004	01.40	01.30
8086 DQ EMULATION	64220	01.01	00.00
8086 DQ SW ANAL	64332B	01.03	01.01
8086 EMUL 300	64222S004	01.00	00.00
8086 EMULATION	64222	01.07	01.06
8086 PLM A,LLL 500	64891S001	01.00	00.00
8086 PLM ASM LLL	64891S003	01.10	00.00
8086 PLM C,A,LLL 500	64890S001	01.00	00.00
8086 PLM CMP ASM LLL	64890S003	01.10	00.00
8086 SW ANAL	64332	02.03	02.02
8086 SW ANALYZER	64341A	01.02	01.01
8086 SYMBOL CONV	64892S003	01.10	00.00
8086 SYMBOL CONV 500	64892S001	01.00	00.00
8086/8 ASSEMB	64853	02.80	02.70
8086/8 ASSEMB 300	64853S004	02.80	02.70
8086/8 ASSEMB 500	64853S001	02.80	02.70
8086/8 ASSEMB DOS	64853S006	02.80	02.70
8086/8 ASSEMB VAX	64853S003	03.00	02.80
8086/8 C	64818	03.90	03.80
8086/8 C 300	64818S004	03.90	03.80
8086/8 C 500	64818S001	03.90	03.80
8086/8 C VAX	64818S003	03.90	03.80
8086/8 INTERFACE 300	64653S004	01.00	00.00
8086/8 PASCAL	64814	03.70	03.60
8086/8 PASCAL 300	64814S004	03.70	03.60
8086/8 PASCAL 500	64814S001	03.70	03.60
8086/8 PASCAL VAX	64814S003	03.70	03.60
8086/8 PRE-PROCESSOR	64653	00.10	00.00
8088 FUI DOS	64763S006	01.02	00.00
8088 DQ SW ANALYZER	64341C	01.02	01.02
8088 SW ANAL	64333	02.03	02.02
8088 DQ EMUL 300	64221S004	01.30	01.20
8088 DQ EMULATION	64221	01.01	00.00
8088 DQ SW ANAL	64333B	01.03	01.01
8088 EMULATION	64226	01.08	01.07
8088 EMULATION 300	64226S004	01.00	00.00
8096 ASSEMB	64860	01.80	01.70
8096 ASSEMB 300	64860S004	01.80	01.70
8096 ASSEMB 500	64860S001	01.80	01.70
8096 ASSEMB DOS	64860S006	01.90	01.80
8096 ASSEMB VAX	64860S003	01.90	01.80
9900/0 ASSEMB	64847	01.80	00.46
9900/0 ASSEMB 300	64847S004	01.80	01.00
9900/0 ASSEMB 500	64847S001	01.80	01.30
9900/0 ASSEMB DOS	64847S006	01.80	00.00
9900/0 ASSEMB VAX	64847S003	01.90	01.80
DIAG/CS 80 EXER/XFER	64934	01.04	01.03
EBPP	64304	01.03	00.70
F8/3870 ASSEMB	64849	00.01	00.00
F9450 EMUL 300	64286S004	01.00	00.00
F9450 EMULATION	64286	01.05	01.04
FILE XFER UT 2&3 VAX	64893S003	01.10	00.00
FILE-XFER UT 2&3 500	64893S001	01.00	00.00

Product name	Product number	uu.ff	prev
GENERIC ANALYSIS FW	64740	01.00	00.03
GENERIC EMULATION FW	64700	01.00	00.05
HI SPD RS422 INTF	64037	00.02	00.01
HOST PASCAL	64817	01.04	00.46
HOST SOFTWARE 300	64883	01.30	01.10
HOST SOFTWARE 500	64880	02.00	01.90
HOST SOFTWARE VAX	64882	02.50	02.40
HP TEAMWORK 300	64711S004	03.00	02.30
HP-IB INTERFACE 300	64695S004	01.00	00.00
HP-IB PRE-PROCESSOR	64695	01.00	00.00
IMB EXTENDER	64303	01.01	00.55
INVERSE ASSEMB	64856	01.01	00.00
MICRO ASSEMB	64861	01.01	01.00
MS1750A ASSEMB	64857	02.00	01.90
MS1750A ASSEMB 300	64857S004	02.00	01.90
MS1750A ASSEMB 300	64857S006	01.00	00.00
MS1750A ASSEMB 500	64857S001	02.00	01.90
MS1750A ASSEMB VAX	64857S003	02.00	01.90
NETWORK TRANSFER 300	64887S004	01.40	01.00
NETWORK TRANSFER 500	64887S001	01.00	00.00
NETWORK TRANSFER VAX	64887S003	01.10	00.00
NSC800 EMULATION	64292	01.03	01.02
NSC800 INTERFACE 300	64690S004	01.00	00.00
NSC800 PRE-PROCESSOR	64690	00.48	00.00
OPERATING SYSTEM	64100	02.11	02.10
P1750 EMUL	64288	01.00	00.00
P1750 EMUL 300	64288S004	01.10	01.00
PROM PROGRAMMER	64501	01.11	01.10
PROM PROGRAMMER 300	64501S004	01.50	01.30
ROM EMULATION	64272	01.04	01.03
RS-232 TRANSFER 300	64885	01.50	01.30
RS-232 TRANSFER 500	64884	01.50	01.40
RS-232 TRANSFER VAX	64886	01.70	01.60
SOFTKEY EDITOR 300	64790S004	02.10	01.00
SOFTKEY EDITOR 500	64790S001	02.10	01.10
STATE 80386	64659	01.00	00.00
STATE ANALYZER	64620	00.71	00.00
STATE ANALYZER	64621	01.07	00.71
STATE ANALYZER 300	64620S004	01.30	01.10
STATE_25MHZ	64320	01.01	01.00
STATE_25MHZ	64321	01.01	01.00
SUBMIT TO 64711S004	64702S004	00.00	00.00
SUBMIT TO 64711S004	64710S004	00.00	00.00
SUBMIT TO 64711S004	64712S004	00.00	00.00
SUBMIT TO 64711S004	64713S004	00.00	00.00
SUBMIT TO 64711S004	64714S004	00.00	00.00
SUBMIT TO 64711S004	64715S004	00.00	00.00
SUBMIT TO 64711S004	64716S004	00.00	00.00
SUBMIT TO 64711S004	64717S004	00.00	00.00
SUBMIT TO 64711S004	64718S004	00.00	00.00
SUBMIT TO 64711S004	64720S004	00.00	00.00
SUBMIT TO 64711S004	B1400A0A6	00.00	00.00
SUBMIT TO 64711S004	B1400A0AE	00.00	00.00
SUBMIT TO 64711S004	B1402A	00.00	00.00
SUBMIT TO 64711S004	B1403	00.00	00.00

Product name	Product number	uu.ff	prev
SUBMIT TO 64711S004	B1403A	00.00	00.00
SUBMIT TO 64711S004	B1404A	00.00	00.00
SUBMIT TO 64711S004	B1405A	00.00	00.00
SUBMIT TO 64711S004	B1406A	00.00	00.00
SUBMIT TO 64711S004	B1412A	00.00	00.00
SW PERF ANALYZER	64310	01.11	01.10
SW PERF ANALYZER 300	64310S004	01.20	01.10
TIMING	64601A	01.04	01.03
TIMING	64601B	01.04	00.00
TIMING ANALYSIS DOS	64741S006	00.01	00.00
TIMING ANALYZER	64600	00.26	00.00
TIMING ANALYZER 300	64610S004	01.60	01.00
TIMING/STATE	64610	01.00	00.00
TMS 320 ASSEMB	64858	01.80	00.00
TMS 320 ASSEMB 300	64858S004	01.80	00.00
TMS 320 ASSEMB 500	64858S001	01.80	01.20
TMS 320 ASSEMB DOS	64858S006	01.80	00.00
TMS 320 ASSEMB VAX	64858S003	01.90	01.80
TMS 32010 MODULES	64285	01.02	01.01
TMS 32020/25 ASMB300	64867S004	01.00	00.00
TMS 32020/25 ASMBDOS	64867S006	01.00	00.00
TMS32020 EMUL FW	64786	01.02	00.00
TMS32020 FUI DOS	64786S006	01.02	00.00
TMS320C25 EMUL FW	64787	00.01	00.00
TMS320C25 FUI DOS	64787S006	01.02	00.00
TMS320C25 PTUI 300	64787S004	01.00	00.00
TMS320C25 MONITOR	64787-11001	00.01	00.00
UPROG	64276	02.00	01.01
USER DEF ASSEMB	64851	00.70	00.00
USER DEF ASSEMB 300	64851S004	02.20	02.10
USER DEF ASSEMB 300	64861S004	02.10	00.00
USER DEF ASSEMB 500	64851S001	02.20	02.10
USER DEF ASSEMB 500	64861S001	02.10	00.00
USER DEF ASSEMB DOS	64851S006	02.20	02.11
USER DEF ASSEMB VAX	64851S003	02.30	02.20
USER DEF ASSEMB VAX	64861S003	02.30	02.00
USER DEF EMUL 300	64274S004	01.20	01.10
USER DEF EMULATION	64274	01.06	01.05
USER DEF INV ASM 300	64856S004	01.00	00.00
USER INTERFACE 300	64808S004	02.10	01.20
USER INTERFACE 500	64808S001	02.10	01.40
UTILITIES PKG 300	64888S004	01.60	01.30
UTILITIES PKG 500	64888S001	01.20	01.00
UTILITIES PKG VAX	64888S003	01.50	01.10
Z8 ASSEMB	64850	00.01	00.00
Z80 EMUL FW	64753	00.01	00.00
Z80 FUI DOS	64753S006	01.02	01.01
Z80 PTUI 300	64753S004	00.01	00.00
Z80 MONITOR	64753-11001	00.01	00.00
Z80 ASSEMB	64842	01.90	01.11
Z80 ASSEMB 300	64842S004	01.90	01.00
Z80 ASSEMB 500	64842S001	01.90	01.30
Z80 ASSEMB DOS	64842S006	01.91	01.90
Z80 ASSEMB VAX	64842S003	02.00	01.90
Z80 EMULATION	64252	01.05	00.56

Product name		Product number	uu.ff	prev
Z80 EMULATION	300	64252S004	01.00	00.00
Z80 INTERFACE	300	64683S004	01.00	00.00
Z80 PASCAL		64812	00.70	00.00
Z80 PRE-PROCESSOR		64683	00.56	00.00
Z80/NSC800 C		64824	02.20	02.10
Z80/NSC800 C	300	64824S004	02.20	02.10
Z80/NSC800 C	500	64824S001	02.20	02.10
Z80/NSC800 C	VAX	64824S003	02.30	02.20
Z80/NSC800PASCAL		64823	02.00	01.90
Z80/NSC800PASCAL	300	64823S004	02.00	01.90
Z80/NSC800PASCAL	500	64823S001	02.00	01.90
Z80/NSC800PASCAL	VAX	64823S003	02.10	02.00
Z8000 C		64820	02.20	02.10
Z8000 C	300	64820S004	02.20	02.10
Z8000 C	500	64820S001	02.20	02.10
Z8000 C	VAX	64820S003	02.30	02.20
Z8000 PASCAL		64816	02.00	01.90
Z8000 PASCAL	300	64816S004	02.00	01.90
Z8000 PASCAL	500	64816S001	02.00	01.90
Z8000 PASCAL	VAX	64816S003	02.10	02.00
Z8001 EMUL	300	64232S004	01.00	00.00
Z8001 EMULATION		64232	02.00	01.07
Z8001 INTERFACE	300	64680S004	01.00	00.00
Z8001 PRE-PROCESS		64680	00.56	00.00
Z8001/2 ASSEMB		64854	01.80	00.00
Z8001/2 ASSEMB	300	64854S004	01.80	00.00
Z8001/2 ASSEMB	500	64854S001	01.80	01.20
Z8001/2 ASSEMB	DOS	64854S006	01.80	00.00
Z8001/2 ASSEMB	VAX	64854S003	01.90	01.80
Z8002 EMUL	300	64233S004	01.00	00.00
Z8002 EMULATION		64233	02.01	02.00
Z8002 INTERFACE	300	64681S004	01.00	00.00
Z8002 PRE-PROCESS		64681	00.56	00.00
Z80H EMULATION		64253	01.02	01.01
Z80H EMULATION	300	64253S004	01.00	00.00

KPR number	page	KPR number	page	KPR number	page	KPR number	page
1650004499	347	5000150292	220	5000225748	264	5000275305	325
1650006544	420	5000151050	144	5000226530	155	5000275693	30
1650006700	180	5000151290	339	5000226563	146	5000275727	268
1650008128	351	5000152819	404	5000229245	296	5000278127	296
1650016618	339	5000153981	393	5000231571	364	5000280958	423
1650020396	219	5000160671	420	5000235150	97	5000281261	13
1650033720	343	5000161182	180	5000238337	318	5000283077	292
1650041624	411	5000162651	269	5000238543	399	5000283937	82
1650042655	277	5000163048	153	5000239012	193	5000285536	277
1650044016	264	5000163303	331	5000239921	340	5000285742	208
1650047167	406	5000163808	195	5000239939	404	5000291294	208
1650047340	406	5000164012	220	5000240259	267	5000291427	394
1650048355	253	5000166983	145	5000240580	126	5000291765	30
1650048652	30	5000169250	181	5000240929	272	5000293530	98
1650049650	13	5000169698	336	5000240937	350	5000293779	128
1650051649	235	5000169995	272	5000240952	363	5000294181	384
1650058925	360	5000170191	415	5000241976	346	5000294199	297
1650061572	294	5000171470	272	5000241984	345	5000294207	221
1650061580	369	5000173815	153	5000242032	193	5000296541	30
1650064923	170	5000176891	295	5000242818	261	5000398396	284
1650069583	93	5000180323	340	5000243048	151	5000401349	369
1650069765	404	5000181131	269	5000244343	267	5000401372	369
1650069773	403	5000181545	421	5000244392	310	5000402214	297
1650071050	219	5000182824	356	5000246983	422	5000403584	31
1650072553	54	5000183475	276	5000247437	151	5000403915	31
1650074567	101	5000183913	182	5000247783	292	5000406058	159
1650074849	93	5000184317	235	5000251322	380	5000406348	298
2700005769	360	5000184374	170	5000251363	261	5000407197	159
2700005918	402	5000188813	354	5000254730	291	5000409094	215
5000089359	377	5000191361	295	5000255752	146	5000409102	384
5000093708	246	5000191544	336	5000258590	151	5000409821	236
5000117002	145	5000191767	324	5000258616	439	5000413161	394
5000122374	373	5000192054	154	5000264481	156	5000415786	147
5000123497	430	5000194951	367	5000264523	14	5000417014	90
5000129023	280	5000196428	183	5000264986	405	5000417063	106
5000131029	319	5000206458	275	5000267468	394	5000417790	384
5000131573	438	5000211359	353	5000269381	356	5000417808	147
5000132662	271	5000211375	368	5000269407	157	5000419440	384
5000134916	430	5000211557	264	5000269415	158	5000420208	119
5000135285	294	5000214858	296	5000269779	127	5000421230	16
5000138941	310	5000219220	279	5000270637	152	5000422394	9
5000139535	404	5000219865	128	5000271957	158	5000422485	17
5000141150	319	5000220418	155	5000272021	311	5000422782	76
5000141747	269	5000220764	397	5000273250	267	5000423129	35
5000143628	220	5000220772	397	5000273268	267	5000423236	93
5000146829	310	5000221200	144	5000273458	146	5000423681	91
5000149724	339	5000222489	401	5000273474	147	5000429126	11
5000150151	431	5000225078	324	5000274506	21	5000429795	103

KPR number	page	KPR number	page	KPR number	page	KPR number	page
D200095992	207	D200098319	12	D200099416	75	D200102814	105
D200096008	410	D200098327	22	D200099424	88	D200102822	80
D200096016	439	D200098335	26	D200099432	78	D200102830	80
D200096024	201	D200098343	27	D200099440	75	D200102913	95
D200096032	278	D200098350	112	D200099523	4	D200103077	99
D200096040	255	D200098368	115	D200102194	95	D200103085	41
D200096057	392	D200098376	23	D200102368	6	D200103093	40
D200096065	217	D200098384	57	D200102376	60	D200103101	39
D200096172	251	D200098392	87	D200102384	60	D200103119	58
D200096180	251	D200098400	99	D200102392	60	D200103127	1
D200096198	251	D200098418	32	D200102400	61	D200103135	2
D200096206	392	D200098426	59	D200102418	61	D200103143	3
D200096214	102	D200098434	59	D200102426	61	D200103150	44
D200096594	293	D200098566	28	D200102434	62	D200103168	45
D200096610	251	D200098574	16	D200102442	62	D200103176	73
D200096727	255	D200098582	34	D200102459	63	D200103184	83
D200096735	244	D200098632	89	D200102467	63	D200103192	74
D200096834	86	D200098640	42	D200102475	63	D200103200	53
D200096859	94	D200098657	91	D200102483	64	D200103218	54
D200097071	109	D200098665	104	D200102491	64	D200103226	84
D200097204	24	D200098707	42	D200102509	65	D200103234	113
D200097212	7	D200098764	19	D200102517	65	D200103242	114
D200097238	51	D200098772	35	D200102525	66	D200103259	15
D200097626	107	D200098780	37	D200102533	66	D200103267	12
D200097659	42	D200098798	37	D200102541	66	D200103275	22
D200098020	91	D200098806	18	D200102558	67	D200103283	26
D200098038	89	D200098814	19	D200102566	68	D200103291	27
D200098137	41	D200098855	8	D200102574	68	D200103309	112
D200098145	40	D200098863	25	D200102582	69	D200103317	115
D200098152	39	D200098871	52	D200102590	70	D200103325	23
D200098160	58	D200098897	10	D200102608	70	D200103333	57
D200098178	1	D200098905	29	D200102616	71	D200103341	46
D200098186	2	D200098913	10	D200102624	71	D200103358	108
D200098194	3	D200098921	29	D200102632	72	D200103366	33
D200098202	44	D200098939	10	D200102640	52	D200103374	56
D200098210	45	D200098947	29	D200102772	91	D200103523	73
D200098244	74	D200099028	78	D200102780	59	D200103531	83
D200098277	84	D200099036	54	D200102798	92	D200103549	53
D200098301	14	D200099390	111	D200102806	92	D200103556	54

- 6301V EMULATION -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64206S004	01.20	Emulation core dumps when run in a small window.	D200095539	117
	64206S004	01.20	"Copy to Read-Only files", fails to deliver an error message to screen	D200095828	118
	64206S004	01.20	Monitor commands may not complete execution correctly with target sys.	D200098178*	1
	64206S004	01.20	Emulation hangs on shell commands that run quickly.	D200103127*	1
	64206S004	00.00	Processes sometimes left running after parent has stopped.	D200082057	116
	64206S004	00.00	Tracelist symbols dissappear.	D200085878	116
	64206S004	00.00	Using simio, then continuing , may not be possible	D200086264	116
	64206S004	00.00	"end" softkey after HP-IB error does not clear command line	D200088229	117
	64206S004	00.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090688	117

- 6301V/03R EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64206	01.01	symbol tables get corrupt.	5000420208	119
	64206	01.01	6301V/03R module cannot be accessed with HP-UX 6.01	D200088088	119
	64206	01.01	Illegal opcode error occur when displaying memory repetetively	D200092122	119

- 6301X EMULATION -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64207S004	01.10	Emulation core dumps when run in a small window.	D200095547	121
	64207S004	01.10	"Copy to Read-Only files", fails to deliver an error message to screen	D200095836	122
	64207S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098186*	2
	64207S004	01.10	Emulation hangs on shell commands that run quickly.	D200103135*	2
	64207S004	00.00	Processes sometimes left running after parent has stopped.	D200082065	120
	64207S004	00.00	Tracelist symbols dissappear.	D200085886	120
	64207S004	00.00	Using simio, then continuing , may not be possible	D200086272	120
	64207S004	00.00	"end" softkey after HP-IB error does not clear command line	D200088237	121
	64207S004	00.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090696	121

- 6301Y EMULATION -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64208S004	01.10	Emulation core dumps when run in a small window.	D200095554	123
	64208S004	01.10	"Copy to Read-Only files", fails to deliver an error message to screen	D200095844	123
	64208S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098194*	3
	64208S004	01.10	Emulation hangs on shell commands that run quickly.	D200103143*	3
	64208S004	00.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090704	123

- 6301Y/03Y EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64208	01.00	Emulator can't work when external clock is selected and E clock = 160khz	D200081596	124

- 64000-UX OP-ENV -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64801S004	02.40	'edbuild' involked from within a product may execute private command.	D200099523*	4
	64801S004	01.80	EDB problems with scoping of locals from new com/asm/linker	D200090472	125

- 64HP-UXVMS8086/8 A M - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
MANUAL	64853-90908	02.03	Need Manual change to explain AC and PH phase errors.	5000240580	126

- 650X ASSEMB - SSB ISSUE DATE: 02/03/89

LINKER	64843	01.00	LNK does load NOLOAD files.	5000269779	127
PROBLEM ON 9000/S300	64843	01.00	LNK does load NOLOAD files.	5000269779	127

- 6800 C - SSB ISSUE DATE: 02/03/89

********none********	64821	02.10	Switch statement causes infinite loop.	5000293779	128
	64821	02.10	Type cast of constant to (char *) in pointer expression error	D200085787	134
	64821	02.10	SHORT_ARITH OFF for some short experssions used as conditional branch	D200085803	135
	64821	01.07	USE OF MANY FUNCTION CALLS WITH CONSTANT PARAMATERS MAY CAUSE ERR #1007	D200074989	131
	64821	01.07	Real variable used as a test condition cause error.	D200081539	133
	64821	01.06	Illegal initialization causes error 1113.	D200068197	130
PASS 3	64821	01.06	Conditional compile fails if it suceeds a fixed parm function call.	D200069823	130
PROBLEM ON 9000/S300	64821	01.20	Libraries cause write to ROM	5000219865	128
	64821	01.07	If condition is tested with a CMP D1,D1	D200079624	132

- 6800 PASCAL - SSB ISSUE DATE: 02/03/89

********none********	64811	02.00	Type casting the ADDR function to SET for masking may cause error	D200093534	138
	64811	02.00	Large Sets may produce invalid results for elements outside set range	D200093542	139
	64811	01.90	Compare using var pointer to first record item fails.	D200093682	141
	64811	01.90	Asignment of string to double dereference string pointer causes error	D200093708	141
	64811	01.90	Pointer dereference of VAR pointer to structure as a parameter fails.	D200093716	142
	64811	01.09	Compiler $FAR ON$, creates incorrect data offsets in listing	D200059980	138

- 6800-03 ASSM - SSB ISSUE DATE: 02/03/89

********none********	64841-90905	01.15	Mask pseudo works incorrectly in certain cases.	5000151050	144
MANUAL	64841-90905	01.15	Support OIM, AIM, EIM, TIM	5000221200	144

- 6800/2 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64841	01.80	JMP to EQU'd label fails.	5000436360*	5
	64841	01.15	External MASKS are not handled properly by the assembler.	5000166983	145
	64841	01.13	Comments are listed in the xref table when not delimited by a ;	5000117002	145
	64841	01.10	Assembler allows the inst. "LDA A". "LDA A" isn't a valid instruction.	5000273474	147
MACROS	64841	01.80	MACRO expansion of constants fails.	5000430900*	5
PROBLEM ON 9000/S300	64841	01.80	XREF missing some labels.	5000415786	147
	64841	01.80	XREF not properly generated.	5000417808	147
	64841	01.80	MACRO expansion of constants fails.	5000430900*	5
	64841	01.10	6301 AIM instruction with ".NT." operator causes LR error.	5000273458	146

- 6800/2 ASSEMB - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
PROBLEM ON 9000/S500	64841	01.40	Xref table is not listing all symbol references.	5000226563	146
	64841	00.00	Very long file causes problems with xref listing on a 2563B	5000255752	146

- 68000 12MHZ EMUL -F SSB ISSUE DATE: 02/03/89

********none********	64742	00.05	Entering the command "cf mon=bg" many times produces an error message.	D200102368*	6
	64742	00.04	Slow Clock interferes with configuring monitor... Poor error messages.	D200089631	149

- 68000 12MHZ FUI -D SSB ISSUE DATE: 02/03/89

********none********	64742S006	01.02	Bringing up multiple windows can cause demon timeout.	D200097212*	7
	64742S006	01.01	All states requested from emtrdata should be valid	D200091587	150
	64742S006	01.01	emtrdata() does not work correctly if upload too big linearray	D200092452	150

- 68000 12MHZ PTUI 300 - SSB ISSUE DATE: 02/03/89

********none********	64742S004	01.00	Deamon hangs when unable to communicate to 64700 pod	D200098855*	8

- 68000 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64845	02.10	Absolute Long code is not generated correctly.	5000422394*	9
	64845	01.13	Math operators not working on 64100.	5000258590	151
	64845	01.12	Size qualifiers in cross reference.	5000247437	151
PROBLEM ON 9000/S300	64845	02.10	No A5 prompt when non-existient .R file specified.	5000270637	152
	64845	02.10	TITLE directive inserting garbage control characters.	D200086678	152
	64845	01.10	Missing whitespace is not flagged.	5000243048	151

- 68000 BBA - SSB ISSUE DATE: 02/03/89

********none********	64380S004	01.10	Misc. severe syntax errors cause "Fatal error in ...bbacpp"	D200098897*	10
	64380S004	01.10	Comments do not force seperator	D200098913*	10
CODE GENERATOR	64380S004	01.10	Struct field and typedef cannot have the same name	D200098939*	10

- 68000 C - SSB ISSUE DATE: 02/03/89

********none********	64819	02.10	Bad code is generated when a char var is compared to a negative number	5000269415	158
	64819	02.10	Cannot combine 'shift' with '&' in same statement for structured var.	5000406058	159
	64819	02.10	"variable = variable = constant" causes failure.	5000407197	159
	64819	02.10	unsigned long i; gets EXT.L when i = 0x8000;	5000429126*	11
	64819	02.10	Zlongreal_sub library error.	D200093294	169
	64819	01.10	Address is not incremented past 0xFFFF for data areas > 32k.	5000220418	155
	64819	01.10	Real variable used as a test condition cause error.	5000226530	155
	64819	01.10	Address comparisons for variables located on negative base-page may fail	D200076513	164

- 68000 C - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64819	01.10	Libraries generate incorrect code 68010 processor.	D200081505	166
	64819	01.10	SHORT_ARITH OFF use of mixed short int in conditionals may not work	D200085373	166
	64819	01.10	SHORT_ARITH OFF with unsigned short int in conditional branch error	D200085399	168
	64819	01.09	Illegal initialization causes error 1113.	5000173815	153
	64819	01.09	Fields of a structure are dereferenced incorrectly (if fields are big).	5000192054	154
	64819	01.09	Shift of wrong sized value in register.	D200063115	161
	64819	01.09	An "if" statement may cause the compiler to go astray.	D200065193	162
	64819	01.09	Libraries load constants into the data area	D200071829	163
	64819	01.08	Pass 3 error 1113 flagged.	5000163048	153
	64819	01.07	Compiler uses MSB of word containing char value rather than LSB.	D200032045	160
CODE GENERATOR	64819	01.07	Bad code using $OPTIMIZE$ and successive uses of the same pointer.	D200014399	160
	64819	00.21	Multiple assignments may cause compiler to reuse an overwritten reg.	D200004929	159
PASS 3	64819	01.09	Conditional compile fails if it suceeds a fixed parm function call.	D200069674	163
PROBLEM ON 9000/S300	64819	02.10	The EXT.L command does not work properly.	5000269407	157
	64819	02.10	Problem with EXT.L command.	5000271957	158
	64819	01.20	Problem with Type Name cast - causes Pass 1 error.	5000264481	156
	64819	01.10	If condition is tested with a CMP D1,D1	D200079590	165

- 68000 C - SSB ISSUE DATE: 02/03/89

********none********	64819-90902	01.09	error 1113	1650064923	170
	64819-90902	01.09	List library link range in manuals.	5000184374	170

- 68000 DQ EMUL - SSB ISSUE DATE: 02/03/89

********none********	64243S004	01.30	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095182	171
	64243S004	01.30	Emulation core dumps when run in a small window.	D200095679	171
	64243S004	01.30	"Copy to Read-Only files", fails to deliver an error message to screen	D200095968	171
	64243S004	01.30	Monitor commands may not complete execution correctly with target sys.	D200098319*	12
	64243S004	01.30	Emulation hangs on shell commands that run quickly.	D200103267*	12

- 68000 EMUL 12.5 MHZ - SSB ISSUE DATE: 02/03/89

********none********	64243	01.01	68000 inverse assembler doesn't recognize the opcode "ILLEGAL"	1650049650*	13
	64243	01.01	Modify memory does not generate correct IEEE format value	5000281261*	13
	64243	01.01	State IA generates wrong instruction for Adr Reg. Indirect w/Indexing	D200081372	172

- 68000 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64242	01.07	Load of more than 1 abs. to targ. mem. not allowed when restricted to RT	D200067637	173

- 68000 EMULATION -3 SSB ISSUE DATE: 02/03/89

********none********	64242S004	01.00	The emulation monitor program may hide target system bus errors.	5000264523*	14

- 68000 EMULATION -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64242S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069484	174
	64242S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080606	174
	64242S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080903	174
	64242S004	01.00	Tracing on status int_ack does not work.	D200081679	175
	64242S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081885	175
	64242S004	01.00	Processes sometimes left running after parent has stopped.	D200082180	175
	64242S004	01.00	Memory breaks during stepping are not detected	D200082594	176
	64242S004	01.00	Loading a trace file from a different processor may cause core dump	D200083196	176
	64242S004	01.00	Tracelist symbols dissappear.	D200085993	177
	64242S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090811	177
	64242S004	01.00	Emulation core dumps when run in a small window.	D200095661	178
	64242S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200095950	178
	64242S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098301*	14
	64242S004	01.00	Emulation hangs on shell commands that run quickly.	D200103259*	15
BREAKPOINT	64242S004	01.00	Software breakpoint in target memory will hang system.	D200082610	176

- 68000 HL SOFT ANAL M - SSB ISSUE DATE: 02/03/89

********none********	64331-90902	01.00	Tracing a variable declared as a pointer to a function doesn't work in C	D200013110	179

- 68000 PASCAL - SSB ISSUE DATE: 02/03/89

********none********	64815	02.00	Type casting the ADDR function to SET for masking may cause an error.	D200093450	191
	64815	01.12	Bad code when taking ADDR of record element when using WITH.	5000161182	180
	64815	01.12	Subrange parameter not passed properly when function returning integer	D200076562	190
	64815	01.11	Declaring a boolean array may cause an out of bounds error.	5000169250	181
	64815	01.11	Casting address to int and adding a signed_16 var generates bad code.	5000183913	182
	64815	01.11	Bytes sign extened in a case statement.	5000196428	183
	64815	01.11	The WARN option cannot be turned off.	D200065045	187
	64815	01.11	Libraries load constants into the data area	D200071696	188
	64815	01.11	Problems with routine STRWRITE & $BASE_PAGE$ mode with ASPIOLIB	D200073007	189
	64815	01.10	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060103	185
	64815	01.10	Compiler generates a LEA instruction with an illegal source operand.	D200060343	186
	64815	00.00	Immediate operand's value is altered when doing a logical and.	1650006700	180
CODE GENERATOR	64815	01.09	Bad code using $RANGE$ or $DEBUG$ with $CALL_PC_LONG$ or LIB_PC_LONG	D200014332	184
	64815	01.08	Compiler generates incorrect code for set inclusion check.	D200013359	184
PASS 3	64815	01.10	Compiler $FAR ON$, creates incorrect data offsets in listing	D200049882	185

- 68000/08/10 ASM - SSB ISSUE DATE: 02/03/89

********none********	64845-90904	01.00	Wrong offset calculated when using PC+index reg+ offset mode of addr.	D200045864	193
	64845-90904	01.00	Include support for BHS and BLO.	D200065565	194
MANUAL	64845-90904	01.10	Alter all assembler manuals to reflect new syntax.	5000239012	193
	64845-90904	01.10	Manual indicates EXT is a legal psuedo for an external declaration.	5000242032	193

- 68000/10 RT S-ANAL M - SSB ISSUE DATE: 02/03/89

********none********	64341-90903	02.00	Non-adjacent symbols not traceable in some conditions.	5000163808	195

- 68000C AXLS COMP - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64902-90901	01.00	HP64000 variable cannot be set to a \net directory. Manual fix requested	5000421230*	16
	64902-90901	01.00	Manual needs to have more info on libraries.	D200098574*	16

- 68000C AXLS COMP 300 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64902S004	02.10	Division with mixed types can generate incorrect code with -O.	5000435669*	17
	64902S004	02.00	-u causes erroneous warning to be generated.	5000422485*	17
	64902S004	02.00	Division of unsigned variable by a constant.	5000439216*	18
	64902S004	02.00	"LOCAL" pseudo not useable inside #pragma ASM.	D200098806*	18

- 68000C AXLS COMP 800 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64902S005	02.00	-u causes erroneous warning to be generated.	D200098764*	19
	64902S005	02.00	"LOCAL" pseudo not useable inside #pragma ASM.	D200098814*	19

- 68008 EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64244	01.01	Unknown software bp error message after software bp is set.	5000274506*	21

- 68008 EMULATION -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64244S004	01.30	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095190	196
	64244S004	01.30	Emulation core dumps when run in a small window.	D200095687	196
	64244S004	01.30	"Copy to Read-Only files", fails to deliver an error message to screen	D200095976	196
	64244S004	01.30	Monitor commands may not complete execution correctly with target sys.	D200098327*	22
	64244S004	01.30	Emulation hangs on shell commands that run quickly.	D200103275*	22

- 6801/3 EMULATION 300 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64256S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069567	197
	64256S004	01.00	State inverse assembler for 6801 does not work	D200077545	197
	64256S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080671	197
	64256S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080978	198
	64256S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081927	198
	64256S004	01.00	Processes sometimes left running after parent has stopped.	D200082255	198
	64256S004	01.00	Under certain conditions the 6801 may not work correctly with SPA	D200082727	199
	64256S004	01.00	Loading a trace file from a different processor may cause core dump	D200083261	199
	64256S004	01.00	Tracelist symbols dissappear.	D200086058	199
	64256S004	01.00	Using simio, then continuing , may not be possible	D200086397	200
	64256S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088351	200
	64256S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090886	200
	64256S004	01.00	Emulation core dumps when run in a small window.	D200095737	201
	64256S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200096024	201
	64256S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098376*	23

- 6801/3 EMULATION 300 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64256S004	01.00	Emulation hangs on shell commands that run quickly.	D200103325*	23

- 68010 16MHZ FUI -D SSB ISSUE DATE: 02/03/89

********none********	64745S006	01.02	Can not run with foreground monitor and standard SSP	D200097204*	24

- 68010 16MHZ PTUI 300 - SSB ISSUE DATE: 02/03/89

********none********	64745S004	01.00	Deamon hangs when unable to communicate to 64700 pod	D200098863*	25

- 68010 EMUL 12.5M 300 - SSB ISSUE DATE: 02/03/89

********none********	64245S004	01.30	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095208	202
	64245S004	01.30	Emulation core dumps when run in a small window.	D200095695	202
	64245S004	01.30	"Copy to Read-Only files", fails to deliver an error message to screen	D200095984	202
	64245S004	01.30	Monitor commands may not complete execution correctly with target sys.	D200098335*	26
	64245S004	01.30	Emulation hangs on shell commands that run quickly.	D200103283*	26

- 68010 G.P. EMUL -3 SSB ISSUE DATE: 02/03/89

********none********	64249S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069534	203
	64249S004	01.00	Incorrect breakpoint behaviour on continuing emulation.	D200072496	203
	64249S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080648	203
	64249S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080945	204
	64249S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081893	204
	64249S004	01.00	Processes sometimes left running after parent has stopped.	D200082222	204
	64249S004	01.00	Memory breaks during stepping are not detected	D200082776	205
	64249S004	01.00	Loading a trace file from a different processor may cause core dump	D200083238	205
	64249S004	01.00	Tracelist symbols dissappear.	D200086025	205
	64249S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090852	206
	64249S004	01.00	Emulation core dumps when run in a small window.	D200095703	207
	64249S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200095992	207
	64249S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098343*	27
	64249S004	01.00	Emulation hangs on shell commands that run quickly.	D200103291*	27
BREAKPOINT	64249S004	01.00	Software breakpoint in target memory will hang system.	D200087288	206

- 68020 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64870S004	01.00	68000 AXLS compiler generates x_ref to _strcpy when it isn't needed.	5000291294	208
	64870S004	01.00	Using asm psued END with numeric expression causes linker error.	D200089276	209
	64870S004	01.00	Ar68k can not handle long list in command line options.	D200089714	209
	64870S004	01.00	PLEN directive does not work properly	D200089722	211
	64870S004	01.00	LLEN directive does not work properly with tab characters	D200089730	211
	64870S004	01.00	Temporary files should be created in /tmp directory	D200089748	211

- 68020C AXLS COMP 800 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64903S005	02.00	"LOCAL" pseudo not useable inside #pragma ASM.	D200098798*	37

- 6805 E - SSB ISSUE DATE: 02/03/89

********none********	64195S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098152*	39
	64195S004	01.10	Emulation hangs on shell commands that run quickly.	D200103101*	39

- 6805 G EMUL - SSB ISSUE DATE: 02/03/89

********none********	64194S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098145*	40
	64194S004	01.10	Emulation hangs on shell commands that run quickly.	D200103093*	40

- 6805 P EMUL - SSB ISSUE DATE: 02/03/89

********none********	64193S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098137*	41
	64193S004	01.10	Emulation hangs on shell commands that run quickly.	D200103085*	41

- 6805 U&R EMUL - SSB ISSUE DATE: 02/03/89

********none********	64192S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200097659*	42
	64192S004	01.10	Registers will not display on term with more than 58 lines	D200098640*	42
	64192S004	01.10	Emulation hangs on shell commands that run quickly.	D200098707*	42

- 6805/9 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64844	01.11	LR error flagged for legal expression of the form 'label-value'.	1650020396	219
	64844	01.11	LEAX [WORD] fails.	1650071050	219
	64844	01.11	HEX pseudo causes byte counter to quit incrementing in certain cases.	5000150292	220
	64844	01.11	Arithmetic expression is not being evaluated correctly.	5000164012	220
	64844	01.11	NT operator not operating consistiently.	D200063164	221
	64844	01.11	BEXT address is not calculated correclty.	D200076950	222
	64844	01.10	Label in IF stmnt. does not appear in XREF	5000143628	220
CODE GENERATOR	64844	01.40	BRSET range not checked.	5000294207	221
PROBLEM ON 9000/S300	64844	01.40	BRSET range not checked.	5000294207	221

- 6809 C - SSB ISSUE DATE: 02/03/89

********none********	64822	01.80	Compare error using address of local variable on right of expression	D200086611	231
	64822	01.80	SHORT_ARITH OFF expressions in branches may not work as K&R	D200086629	232
	64822	01.80	Switch statement using unsigned int values 0 and 0xFFFF creates error	D200093575	234
	64822	01.08	Some C programs using pointer & structure dereferences cause error #1006	D200075036	227
	64822	01.08	Programs with duplicate goto labels may fail in Pass 3 on VAX&HPUX C	D200075663	228
	64822	01.08	Real variable used as a test condition cause error.	D200081547	230

- 6809 C - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64822	01.07	Illegal initialization causes error 1113.	D200068239	224
	64822	01.07	Conditional compile fails if it suceeds a fixed parm function call.	D200069864	224
	64822	01.07	Use of address (&) stack vars on right side of conditional expression	D200073171	225
PROBLEM ON 9000/S300	64822	01.08	If condition is tested with a CMP D1,D1	D200079632	229

- 6809 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64215S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098202*	44
	64215S004	01.10	Emulation hangs on shell commands that run quickly.	D200103150*	44

- 6809 PASCAL - SSB ISSUE DATE: 02/03/89

********none********	64813	01.70	Type casting the ADDR function to SET for masking may cause an error.	D200093468	241
	64813	01.70	Large Sets may produce invalid results for elements outside set range	D200093526	242
	64813	01.70	Write statement causes ERROR 1006.	D200096735	244
	64813	01.60	Incorrect test generated for more than the 256th label.	5000409821	236
	64813	01.30	If >39 functions declared; following funcs may include bad code.	1650051649	235
	64813	01.11	With statements used in FOR loops on records may cause error #1006	D200075010	240
	64813	01.11	Compiler incorrectly assumes the value of a var is in the D register.	D200082446	241
	64813	01.10	Records of pointers to text not handled correctly.	5000184317	235
	64813	01.10	ADDR function for stack relative variables in right side conditionals	D200073155	237
PASS 3	64813	01.09	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060020	236

- 6809 PASCAL - SSB ISSUE DATE: 02/03/89

********none********	64813-90903	00.02	Parameter passing thru the registers has changed.	5000093708	246

- 6809E EMULATION -3 SSB ISSUE DATE: 02/03/89

********none********	64216S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098210*	45
	64216S004	01.10	Emulation hangs on shell commands that run quickly.	D200103168*	45

- 680XX DEBUG/SIM -3 SSB ISSUE DATE: 02/03/89

********none********	64360S004	00.00	Pressing <shift><home> in high level code window does not work.	D200094599	247
	64360S004	00.00	Section pragma causes unpredictable behavior.	D200094870	247
	64360S004	00.00	Illegal use of Expression Monitor Value corrupts array without warning	D200094888	247
	64360S004	00.00	Floating pt values are rounded and/or displayed as integral values.	D200094896	247
	64360S004	00.00	Comment lines echoed to journal window with Command Echo OFF.	D200094904	248
	64360S004	00.00	Debugger will not break on access of inport address.	D200094938	248
	64360S004	00.00	No error when writing to a file opened for read.	D200094961	250
	64360S004	00.00	Macros can not have parameter names that have underscores	D200094979	250
	64360S004	00.00	If push Step_Over after trying Step_Into printf, pgm runs to completion	D200095117	250
	64360S004	00.00	Too many memory map commands yields erroneous breakpoint error.	D200096172	251

- 680XX DEBUG/SIM -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64360S004	00.00	"pi" register is not maintained across Save_state and Load_state.	D200096180	251
	64360S004	00.00	Unsetting TERM in the environment causes core dump.	D200096198	251
	64360S004	00.00	These do not evalueate to the same result: "1+(&test)" and "(&test)+1"	D200096610	251

- 68HC11 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64265S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088377	254
	64265S004	01.10	Code disp. with trace not right if code changed w/o ending emul. session	D200090902	254
	64265S004	01.10	Emulation core dumps when run in a small window.	D200095752	255
	64265S004	01.10	"Copy to Read-Only files", fails to deliver an error message to screen	D200096040	255
	64265S004	01.10	Target system resets on "display memory" command.	D200096727	255
	64265S004	01.10	Emulation hangs on shell commands that run quickly.	D200103341*	46
	64265S004	01.00	68HC11 will work alone as a measurement system.	1650048355	253
	64265S004	01.00	Processes sometimes left running after parent has stopped.	D200082271	253
	64265S004	01.00	Loading a trace file from a different processor may cause core dump	D200083287	253
	64265S004	01.00	Tracelist symbols dissappear.	D200086066	253
	64265S004	01.00	Using simio, then continuing , may not be possible	D200086413	254

- 70108 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64295S004	01.10	Processes sometimes left running after parent has stopped.	D200082313	256
	64295S004	01.10	Tracelist symbols dissappear.	D200086108	256
	64295S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088393	256
	64295S004	01.10	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095240*	47

- 70116 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64294S004	01.10	Processes sometimes left running after parent has stopped.	D200082305	258
	64294S004	01.10	Tracelist symbols dissappear.	D200086090	258
	64294S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088385	258
	64294S004	01.10	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095216*	48

- 70208 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64297S004	01.00	Processes sometimes left running after parent has stopped.	D200082339	260
	64297S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088419	260
	64297S004	01.00	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095281*	49

- 70216 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64296	01.00	V50 Disassembler generates "illegal" opcode for "POP PS" instruction	5000242818	261
	64296	01.00	Can not specify needed trigger specification.	5000251363	261
	64296S004	01.00	Processes sometimes left running after parent has stopped.	D200082321	262
	64296S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088401	262

- 70216 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64296S004	01.00	Open of file pvxxxxfile_asmb causes pv failure on long file name sys.	D200095265*	50

- 80186 - SSB ISSUE DATE: 02/03/89

********none********	64764S006	01.02	Bringing up multiple windows can cause demon timeout.	D200097238*	51

- 80186 - SSB ISSUE DATE: 02/03/89

********none********	64764S004	01.00	Deamon hangs when unable to communicate to 64700 pod	D200098871*	52
	64764S004	01.00	Performance measurements do not work with non zero segments.	D200102640*	52

- 80186 EMUL FW - SSB ISSUE DATE: 02/03/89

********none********	64764-90901	01.00	The Manual says that step is not allowed in real time mode.	D200089847	263
	64764-90901	01.00	The "stty" command doesn't work correctly for baud rate <= 1200.	D200090167	263

- 80186 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64224	01.05	LODS instructions with segment override not properly disassembled.	5000225748	264
	64224	01.04	"disp. memory mnemonic" shows incorrect inv. assembly for JMP NEAR inst	5000211557	264
	64224	00.00	"run from <addr>", "modify reg <reloc>" generates 16 extra I/O writes.	1650044016	264

- 80186 EMULATION -3 SSB ISSUE DATE: 02/03/89

********none********	64224S004	01.20	Open of file pvxxxxfile_asmb causes pv failure on long file name systems	D200095463	265
	64224S004	01.20	Emulation hangs on shell commands that run quickly.	D200103200*	53
	64224S004	01.20	Running w/o monitor will corrupt the offset value on INT 0 vector.	D200103549*	53

- 80188 EMULATION - SSB ISSUE DATE: 02/03/89

USER MEMORY	64225	01.03	Emulator would not recover from errors during display memory repetitive.	D200065805	266

- 80188 EMULATION -3 SSB ISSUE DATE: 02/03/89

********none********	64225S004	01.20	Unable to download large amounts of code to target memory	D200099036*	54
	64225S004	01.20	Emulation hangs on shell commands that run quickly.	D200103218*	54
	64225S004	01.20	Running w/o monitor will corrupt the offset value on INT 0 vector.	D200103556*	54
	64225S004	01.02	Incorrect reads and writes to odd target system memory locations	1650072553*	54

- 80286 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64228	01.02	trace only <Odd Address> data 0: analyzer doesn't qualify properly.	5000240259	267

- 80286 EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64228	01.02	80286 emul. fails to run programs mapped as user memory at the target.	5000244343	267
	64228	01.02	80286 Emulator may not display proper Interrupt Type number.	5000273250	267
	64228	01.02	trace abt addr 0:0E0H status rd mem triggers on addresses 0E0h, 0C0H.	5000273268	267
	64228	01.02	"list printer memory" command gives wrong addresses using seg:offset.	5000275727	268
	64228	01.02	First PV cycle shows failure with some 64155B cards, if PV'd 1st on 228.	D200080127	268

- 80286 UDE - SSB ISSUE DATE: 02/03/89

DISASSEMBLER	64227	01.00	Incorrect data is returned on a trace about an I/O port.	5000181131	269
DISPLAY MEMORY	64227	01.00	Inverse assembler does not work properly during display memory mnemonic.	5000141747	269
	64227	01.00	The IDIV instruction is not correct during a display memory mnemonic.	5000162651	269
INSTRUCT. EXECUTION	64227	01.00	Single step function does not work after a software breakpoint.	D200046714	269

- 80286B ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64859	01.40	MODULE pseudo generates random relocation type	D200092734	270
	64859	01.02	Aliases not allowed in the linker to specify library paths.	D200068775	270

- 80386 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64420S004	01.00	Emulation hangs on shell commands that run quickly.	D200103374*	56

- 8048 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64846	01.00	Error message LR generated on valid JMP instruction	5000132662	271

- 8051 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64855	01.08	Assembler inconsistant in permitting forward referencing	5000169995	272
	64855	01.08	Defining a transfer address causes an ET error	5000171470	272
	64855	01.08	HIGH operator does not function correctly	D200068379	273
	64855	01.08	CONT in linker will overwrite addresses of variables in different module	D200091710	273
	64855	01.08	Cross reference goes into endless loop on macro reference.	D200092098	274
CODE GENERATOR	64855	01.20	Special operator "HIGH" does not work with DS pesudo opcode	5000240929	272
	64855	01.08	HIGH does not work	D200081570	273
PROBLEM ON 9000/S300	64855	01.20	Special operator "HIGH" does not work with DS pesudo opcode	5000240929	272

- 8051 ASSM - SSB ISSUE DATE: 02/03/89

CODE GENERATOR	64855-90902	01.05	In the manula pg 8-2 states the BIT instruc. shows operand is address.	5000206458	275
MANUAL	64855-90902	01.07	The assmblr manual needs to be updated w/ information in reference manul	D200086439	275

- 8051 EMUL - SSB ISSUE DATE: 02/03/89

********none********	64264-90901	01.01	Manual enhancement to reflect Port display info in more detail.	5000183475	276

- 8051 EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64264	01.04	MODIFY EXTERNAL MEMORY WITH ODD INITIAL ADDRESS DOES NOT WORK CORRECTLY	5000285536	277
	64264	00.00	Cannot load absolute file using remote file access.	1650042655	277

- 8051 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64264S004	01.00	Emulation core dumps when run in a small window.	D200095745	278
	64264S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200096032	278
	64264S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098384*	57
	64264S004	01.00	Emulation hangs on shell commands that run quickly.	D200103333*	57

- 8080/5 ASSEMB - SSB ISSUE DATE: 02/03/89

CODE GENERATOR	64840	01.00	xref incorrect with conditional assmbly IF when code generated for false	5000219220	279

- 8085 B PASCAL - SSB ISSUE DATE: 02/03/89

********none********	64825	01.90	Type casting the ADDR function to SET causes error #1006 on the VAX	D200093641	281
	64825	01.02	Incorrect data offsets in listing file.	D200060228	280
PASS 1	64825	01.01	$Range ON$ causes incorrect code to be generated for a test operation.	5000129023	280

- 8085 C - SSB ISSUE DATE: 02/03/89

********none********	64826	01.04	Real variable used as a test condition cause error.	D200081562	282
PASS 3	64826	01.03	Conditional compile fails if it suceeds a fixed parm function call.	D200069948	282

- 8085 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64203	01.07	64203A (8085) MEMORY MAPPING PROBLEMS	5000398396	284

- 8085 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64203S004	01.40	Emulation core dumps when run in a small window.	D200095521	285
	64203S004	01.40	"Copy to Read-Only files", fails to deliver an error message to screen	D200095810	285
	64203S004	01.40	Monitor commands may not complete execution correctly with target sys.	D200098160*	58
	64203S004	01.40	Emulation hangs on shell commands that run quickly.	D200103119*	58

- 8085 PASCAL - SSB ISSUE DATE: 02/03/89

CODE GENERATOR	64810	00.70	Compiler generates incorrect code for BOOLEAN assignment statement.	D200013334	286

- 8086 - SSB ISSUE DATE: 02/03/89

********none********	64762	00.02	Floating point disp from "nreg" usb't as accyrate as it should be.	D200098426*	59

- 8086 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64762	00.02	Analysis trace display does not distinguish bg from fg coproc. cycles.	D200098434*	59
	64762	00.02	IAL incorrectly disassembles and displays some instructions.	D200102780*	59

- 8086 -A SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64871S004	01.00	Should not be able to take offset of reg expression	D200102376*	60
	64871S004	01.00	Don't allow seg overrides on constants	D200102384*	60
	64871S004	01.00	EQ, And, etc. shouldn't allow mem and abs as operands	D200102392*	60
	64871S004	01.00	LOW, OFFSET, etc.shouldn't allow complex mem operand	D200102400*	61
	64871S004	01.00	Can't negate a relocatable item or memory location.	D200102418*	61
	64871S004	01.00	HIGH and LOW allow bad operands	D200102426*	61
	64871S004	01.00	Not Detecting variable not in group	D200102434*	62
	64871S004	01.00	LOW of a record field returns wrong value.	D200102442*	62
	64871S004	01.00	Expressions using constants > 32k shouldn't work.	D200102459*	63
	64871S004	01.00	%IN and %OUT should generate unknown macro errors.	D200102467*	63
	64871S004	01.00	8224 errors & warnings if pagewidth=(i) and (i)>231.	D200102475*	63
	64871S004	01.00	Ap86 changing case of text in macro arguments.	D200102483*	64
	64871S004	01.00	Ap86 using unsigned values when should be signed.	D200102491*	64
	64871S004	01.00	Using seg sym override on abs expression gives bad OMF.	D200102509*	65
	64871S004	01.00	%EVAL using unsigned values for negative values	D200102517*	65
	64871S004	01.00	Segment overflow should generate an ERROR since OMF is bad.	D200102525*	66
	64871S004	01.00	When called casesensative, won't recognize 06h	D200102533*	66
	64871S004	01.00	PUBDEF/DEBSYM recs wrong when groups involved.	D200102541*	66
	64871S004	01.00	NAME stuff anything gives CONTINUATION line error.	D200102558*	67
	64871S004	01.00	Missing THEN causes FORTRAN I?O error 922	D200102566*	68
	64871S004	01.00	%NASTY(%1%sinple) not expanded correctly.	D200102574*	68
	64871S004	01.00	JZ lab1 lab1 LABEL FAR (fwd) accepted, should error.	D200102582*	69
	64871S004	01.00	Still silence @ stdout for errors/no -1 opt	D200102590*	70
	64871S004	01.00	Don't allow group overrides on exprs in ORG statements.	D200102608*	70
	64871S004	01.00	Core dumps on db "853 char long string	D200102616*	71
	64871S004	01.00	EQU containing segment override fails.	D200102624*	71
	64871S004	01.00	sigint and sigquit wrong	D200102632*	72

- 8086 DQ EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64220S004	01.30	Emulation hangs on shell commands that run quickly.	D200103176*	73
	64220S004	01.30	Running w/o monitor will corrupt the offset value on INT 0 vector.	D200103523*	73

- 8086 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64222S004	01.00	Display memory line crossing segment boundary will be wrong	D200081240	287
	64222S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081414	287
	64222S004	01.00	Processes sometimes left running after parent has stopped.	D200082123	287
	64222S004	01.00	Loading a trace file from a different processor may cause core dump	D200083139	288
	64222S004	01.00	"modify memory" command results in an "end release".	D200084921	288
	64222S004	01.00	Tracelist symbols dissappear.	D200085944	288

- 8086 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64222S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090753	289
	64222S004	01.00	Emulation core dumps when run in a small window.	D200095604	289
	64222S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200095893	290
	64222S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098244*	74
	64222S004	01.00	Emulation hangs on shell commands that run quickly.	D200103192*	74

- 8086-89 ASSM - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
MANUAL	64853-90907	02.01	8086 Asm/linker manual doesn't doc. valid DQ and DT directives.	5000254730	291

- 8086/8 ASSEMB - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64853	02.80	EXT with the 70108 causes US error.	D200096594	293
CODE GENERATOR	64853	02.30	Assembler does not handle all string comparisons correctly.	5000247783	292
PROBLEM ON 9000/S300	64853	02.70	CMP statement is producing wrong label address.	5000283077	292
	64853	02.30	Assembler does not handle all string comparisons correctly.	5000247783	292

- 8086/8 C - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64818	03.80	Pointer deref. followed by type conv., generates bad code.	D200099440*	75
	64818	03.70	Constant divided by short in function call generates wrong code.	1650061572	294
	64818	03.70	Bad code generated on 64000 with "80286" directive.	5000278127	296
	64818	03.70	1006 Fatal error when use structure pointers inside IF statement.	5000406348	298
	64818	03.70	IF statement loads wrong segment for compare statement.	D200090332	308
	64818	03.02	Fields of a structure are dereferenced incorrectly (if fields are big).	D200076695	304
	64818	03.02	Real variable used as a test condition cause error.	D200081513	306
	64818	03.01	Conditional expressions with unsigned mixed operands may fail	D200063057	301
	64818	03.01	Illegal initialization causes error 1113.	D200068080	301
	64818	03.01	Libraries load constants into the data area	D200071787	303
	64818	03.00	1006 message generated when referenced to unspecified array element	5000135285	294
	64818	03.00	Illegal instruction generated by ASM_FILE	D200049908	300
	64818	02.00	Compiler using unacceptable amount of stack space for procedure returns.	D200038836	299
CODE GENERATOR	64818	03.70	PASS 2 error when pntr type used to invoke code stored in array.	D200085738	307
	64818	03.02	Bad code generated when casting a real constant into an integer	5000176891	295
	64818	03.02	When $POINTER_SIZE 32$ generates 32 bit arithmetic for 16 bit variables	5000191361	295
	64818	03.02	Compiler generates MOV SP, BP and LEAVE. This is redundant.	5000229245	296
	64818	03.02	Casting ptr. to int as short & incrementing it generates bad code	D200068700	301
	64818	03.02	~, &, | and ^ may not correctly expand shorts in condidionals	D200079343	305
	64818	03.01	Bad code generated when left shift short variable & AND w/ unsigned int	5000214858	296
	64818	00.56	Vars ORGed in seg. 0 in SHORT env. access current DS seg with no warning	D200008342	299
PASS 3	64818	03.01	Conditional compile fails if it suceeds a fixed parm function call.	D200069716	302
PROBLEM ON 9000/S300	64818	03.80	Incorrect code being generated for array assignment	5000440743*	75
	64818	03.70	& address operator generates PUSH DS1 when DS1 not defined.	5000294199	297
	64818	03.70	Wrong code generated for structure in while loop.	5000402214	297
	64818	03.70	Long arithmetic expression generates incorrect code.	D200092080	309
	64818	01.10	If condition is tested with a CMP D1,D1	D200079608	305

- 8086/8 C - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
PROBLEM ON 9000/S500	64818	03.70	Call to function using LONGS uses wrong segment.	D200099416*	75

- 8086/8 PASCAL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64814	03.60	Vector ref. followed by array ref. generates bad code.	D200099432*	78
	64814	03.50	Our products on the VAX do not implement directory default protection	5000422782*	76
	64814	03.50	Boolean Index into array generates bad code	D200093476	315
	64814	03.50	Test for set inclusion checks beyond the set boundary.	D200093484	316
	64814	03.02	CASE statement produces bad code for complicated expression	5000272021	311
	64814	03.02	Nested IFs inside a WITH may generate incorrect code.	D200027516	313
	64814	03.00	Out of expression storage error generated on code that ran on old ver.	5000138941	310
	64814	03.00	Libraries load constants into the data area	5000146829	310
	64814	02.01	80186Generates wrong offset within CONST_data area	D200047779	313
	64814	01.90	Error 1006 for complex statement using MOD operator	D200093518	317
CODE GENERATOR	64814	03.02	ERROR 117 generated, but does not indicate variable in error	5000244392	310
	64814	00.60	Byte values may be converted to 16-bit before comparison with byte var.	D200010280	312
	64814	00.46	Data structures larger than 64K are not flagged as an error.	D200006080	311
PASS 3	64814	03.00	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060061	313
PROBLEM ON 9000/S300	64814	03.60	Unable to assign a one dim array to a two dim array	D200099028*	78
	64814	03.50	Assignment of constant into array of 3 elements does not work.	D200090597	314
PROBLEM ON VAX	64814	03.60	Error messages are not specific.	5000430454*	76
	64814	03.60	DIV uses incorrect segment.	5000430611*	76
	64814	03.60	ES register destroyed when accessing external 32 bit pointer.	5000431965*	77
	64814	03.60	Procedure call after vector ref causes 1006 error on VAX	D200102822*	80
	64814	03.60	Variable function call inside IF statement generates bad code.	D200102830*	80
RUN-TIME LIBRARY	64814	01.10	Failed to detect out-of-bounds case.	D200014944	312

- 8086/88 C - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64818-90905	00.00	Enhance to include a disc. on the symbol limitations of the compiler.	5000283937*	82

- 8086/88 PASCAL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
MANUAL	64814-90903	03.00	Change manual to say that libraries need to be in same segment	5000238337	318

- 8086/88/186/188HLSAM - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64332-90902	02.00	Display variable may result in "ERROR:E64".	5000131029	319
	64332-90902	02.00	Data structures too large to display in "display variable" command.	5000141150	319

- 8088 DQ EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64221S004	01.20	Emulation hangs on shell commands that run quickly.	D200103184*	83
	64221S004	01.20	Running w/o monitor will corrupt the offset value on INT 0 vector.	D200103531*	83

- 8088 EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64226S004	01.00	Display memory line crossing segment boundary will be wrong	D200081273	320
	64226S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081422	320
	64226S004	01.00	Processes sometimes left running after parent has stopped.	D200082156	320
	64226S004	01.00	Loading a trace file from a different processor may cause core dump	D200083162	321
	64226S004	01.00	"modify memory" command results in an "end release".	D200084947	321
	64226S004	01.00	Tracelist symbols dissappear.	D200085969	321
	64226S004	01.00	Emulator does not work reliably with 64155B memory controller	D200089920	322
	64226S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090787	322
	64226S004	01.00	Emulation core dumps when run in a small window.	D200095638	323
	64226S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200095927	323
	64226S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098277*	84
	64226S004	01.00	Emulation hangs on shell commands that run quickly.	D200103226*	84

- 8096 ASSEMB - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64860	01.04	Pseudo instruction DCB treats absolute variable as relocatable.	5000275305	325
CODE GENERATOR	64860	01.03	Linker does not allocate the file at even addresses	5000191767	324
	64860	01.03	Using ORG statemnts can generate ERR_LR errors	5000225078	324

- EMUL TERMINAL IF - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64740-90901	01.00	"step" doesn't work when CMP is active; (need to change the manual)	D200091249*	85

- EMUL TERMINAL IF CMB - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64306-90901	01.00	"step" doesn't work when CMP is active; (need to change the manual)	D200096834*	86

- F9450 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64286S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090928	326
	64286S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098392*	87

- F9450 EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64286	01.04	RS232 Simulated IO will overrun the user's read buffer sometimes.	D200075150	327

- GENERIC ANALYSIS - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64740-90909	01.00	Errors in xtt help screen.	D200087395	328
	64740-90909	01.00	Measurements between the external/internal analyzers aren't synchronized	D200099424*	88

- GENERIC EMULATION FW - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64700	01.00	Combining CMB trigger driving and receiving can hang the system	D200098038*	89

- GENERIC EMULATION FW - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64700	01.00	Base 10 output of a 32-bit analyzer field contains the character ":".	D200098632*	89
	64700	00.05	Odd byte format records may cause an extra byte written to memory	D200091264	329
	64700	00.00	Ending value of data stream does not report proper error.	D200085530	329
	64700-90901	01.00	Manual needs to clarified concerning fg mon and user prog loading	5000417014*	90
	64700-90901	01.00	Improper coverage calculation of overlapping ranges	D200085647	330
	64700-90901	00.01	The "stty" command doesn't work correctly for baud rate <= 1200.	D200090134	330

- HOST PASCAL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64817	01.04	IOERROR not generated.	5000163303	331
	64817	01.04	Spurious run-time error doing WRITE(REAL_VAL) after previous I/O error	D200014357	331
	64817	01.04	STRWRITE function may produce run time error in specific case.	D200015305	331

- HOST SOFTWARE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64883	01.20	4K .A file high speed link failure	5000423681*	91
	64883	01.10	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093609	333
	64883	01.10	Break or ^C may not abort a foreground transfer with a file list	D200093633	334
	64883	01.10	Break or ^C may not abort a foreground transfer with a file list	D200093880	334
	64883	01.10	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093914	334
	64883	01.10	HPIB SELECT PORT PROBLEM	D200098020*	91
	64883	01.10	Output warning msg. if number of ndilbuffers is <13	D200098657*	91
	64883	01.10	A warning in a filelist transfer will abort the transfer.	D200102772*	91
	64883	01.00	Cluster - Cluster Transfer does not work with filelist	D200085076	333
TRANSFER	64883	01.00	Transfer does not handle extra line-feeds in file.	D200079483	333
	64883	01.00	Incorrect syntax/usage may not result in warning or error message.	D200079681	333

- HOST SOFTWARE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64880	01.90	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093583	337
	64880	01.90	Break or ^C may not abort a foreground transfer with a file list	D200093617	338
	64880	01.20	Transfer to blank userid does not translate file names correctly.	D200036608	337
	64880	01.20	xx.L TO xx:link_sym translation wrong for 0 length records (types 3 & 4)	D200037275	337
RCMAIN	64880	01.20	A session command is req'd before entering the menu in batch jobs.	D200043877	337
TRANSFER	64880	01.60	Transfer may not move library files.	5000191544	336
	64880	01.06	Transfer does not correctly parse "FILE:USERID:@HSL".	5000169698	336
TRANSLATE	64880	01.50	C.K.1 and C.K.2 both translate to C_K on the 64000.	D200062539	337

- HOST SOFTWARE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64882	02.40	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093591	342
	64882	02.40	Break or ^C may not abort a foreground transfer with a file list	D200093625	342
	64882	02.40	A warning in a filelist transfer will abort the transfer.	D200102798*	92
	64882	02.40	Recompile and relink on VMS 5.0 with 5.0 runtime libraries	D200102806*	92
	64882	01.60	Transfer fails when downloading relocatable libraries	1650016618	339

- HOST SOFTWARE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64882	01.60	RCMAIN corrupts RCDEVICE.dat file when aborted with Cntl C or Y	5000151290	339
	64882	01.20	Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.	D200045096	340
	64882	01.20	LONG COMMANDS GREATER THAN 1024 CHAR. MALFUNCTION WITH DMF-32 I/O CARD	D200047217	341
HIGH SPEED LINK	64882	01.60	HSL transfer from within RCMAIN does not return control to RCMAIN.	5000149724	339
RCMAIN	64882	01.70	RCDEVICE.DAT is not properly maintained.	5000180323	340
	64882	01.70	/DEVICES= does not work with a list of stations.	D200064055	341
	64882	01.60	HSL transfer from within RCMAIN does not return control to RCMAIN.	5000149724	339
	64882	01.60	Vax rcdevice file not updated correctly	D200059428	341
	64882	01.60	VAX remote control dumps when a very long command is entered	D200059444	341
TRANSFER	64882	02.00	Transfer of files over DECnet causes program to crash	5000239921	340

- HP TEAMWORK - SSB ISSUE DATE: 02/03/89

********none********	64711S004	03.00	Text is left on display after leaving a spawned shell	5000435248*	94
	64711S004	03.00	Index editor dies when doing a copy subtree	5000439620*	94
	64711S004	03.00	Capital file names not accepted by where referenced report.	D200102194*	95
	64711S004	03.00	SECURITY VIOLATION occurs when ID Module is not first in hil chain	D200102913*	95
	64711S004	02.30	Checker does not work properly for SEM cell with multiple actions	1650069583*	93
	64711S004	02.30	twk2hpgl can't convert the postscript output generated by twk_image	1650074849*	93
	64711S004	02.30	Formatted P-Spec prints FOOTER_TEXT on second page.	5000423236*	93
	64711S004	02.30	Simultaneous socket connections cause a hang.	D200089342	343
	64711S004	02.30	DOMAIN -SQRT ERROR generated when Data Flows become tangential to bubble	D200090118	343
	64711S004	02.30	Cannot run dc_server on machine which has file system nfs mounted	D200096859*	94
	64711S004	02.00	Removing models from the index does not delete all its files.	1650033720	343

- HP TEAMWORK SA - SSB ISSUE DATE: 02/03/89

********none********	64710-90903	01.00	Would like where_refer opt to determine which modules call a given mod.	5000241984	345
	64711-90903	01.00	Would like to see load M-SPECS using a command similar to load_dd cmd.	5000241976	346
	64711-90903	01.00	PRINT OBJECTS from the PI doesn't work correctly.	D200077636	346
	64711-90903	01.00	load_dd -c false_file_name causes an unknown error.	D200077883*	96
	64711-90903	01.00	Spline is too large for binder.	D200077891	346

- HP-UX 68000/8/10 A M - SSB ISSUE DATE: 02/03/89

********none********	64845-90905	01.30	Assembler flagging LR error for correct offset when using PC+IND+OFFSET.	1650004499	347
	64845-90905	01.30	Wrong offset calculated when using PC+index reg+ offset mode of addr.	D200045880	347
MANUAL	64845-90905	01.04	Cannot substitute Macro parameter at beginning of variable.	D200081836	347

- HP-UX 68000/8/10 C M - SSB ISSUE DATE: 02/03/89

********none********	64819-90903	01.40	Byte parameters are pushed onto the stack incorrectly.	D200064386	349

- HP-UX 8051 ASSM - SSB ISSUE DATE: 02/03/89

MANUAL	64855-90903	01.40	Change 8051 manual page 8-4	5000240937	350

Keyword	Product number	uu.ff	Description	KPR number	page
			- HP-UX 8085 C - SSB ISSUE DATE: 02/03/89		
********none********	64826-90902	01.50	New and dispose have inconsistient parameters	1650008128	351
			- HP-UX 8086/88 ASSM M - SSB ISSUE DATE: 02/03/89		
CODE GENERATOR	64853-90905	02.20	.	D200079574	352
MANUAL	64853-90905	02.20	.	D200079574	352
			- HP-UX 8086/88 C - SSB ISSUE DATE: 02/03/89		
********none********	64818-90903	03.02	Additional info about the $SEPARATE_CONST$ directive works, pg. 2-3.	5000211359	353
			- HP-UX 8086/88 PAS -M SSB ISSUE DATE: 02/03/89		
********none********	64814-90904	01.01	DOC. FOR THE PASCAL LIB. ERROR HANDLING ROUTINES NEEDS IMPROVEMENT.	5000188813	354
			- HP-UX OP SYS - SSB ISSUE DATE: 02/03/89		
********none********	64801-90903	01.00	Add documentation for configuring swap space.	5000235150*	97
	64801-90903	01.00	Meas system unuseable if WINDEX exited without ending measurement.	D200079517	355
	64801-90903	01.00	ftio command for hp-ux 6.01 does not function as documented.	D200090431	355
			- HP-UX SYSTEM INST -M SSB ISSUE DATE: 02/03/89		
********none********	64880-90901	01.02	DOC SHOULD INCLUDE LIST OF SUPPORTED CARDS FOR RS232 XFER.	5000182824	356
	64880-90901	01.00	Manual needs to be more explicit about /dev/ttyXX where XX is numeric	5000269381	356
HIGH SPEED LINK	64880-90901	01.02	Fails to transfer first passworded file, but doesn't notify the user.	D200068429	356
			- HS ST/TIM ANAL - SSB ISSUE DATE: 02/03/89		
********none********	64610-90901	01.00	Remove HP2225 and HP2631G printers for support from the manual.	5000293530*	98
			- INVERSE ASSEMB - SSB ISSUE DATE: 02/03/89		
********none********	64856	01.01	Can loop forever when a source file contains macros.	D200077933	357
			- NETWORK TRANSFER 300 - SSB ISSUE DATE: 02/03/89		
********none********	64887S004	01.00	The transferII utility does not work using nft as the transport	D200093088	358
			- NSC800 EMULATION - SSB ISSUE DATE: 02/03/89		
********none********	64292	01.03	Incorrect memory display between two odd addresses	D200094490	359

- OPERATING SYSTEM - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64100	02.10	Problem with Macro code generation.	1650058925	360
	64100	02.10	Macro use of a label is missing from xref.	D200086694	362
	64100	02.07	Recover cmd on 64000(PISCES I) will recover all types on disc's > 150Mb	D200084897	361
	64100	02.02	Nested macro calls cause incorrect macro expansion.	D200041178	360
	64100	02.00	CDC FLOPPY DRIVE DOESN'T FORMAT CORRECTLY IN A COMMAND FILE.	D200015297	360
DC600	64100	02.06	store to DC600 causes 64000 to reboot.	D200069989	361
	64100	01.39	DC600 backup hangs up when it encounters a defective tape.	2700005769	360

- P1750 -E SSB ISSUE DATE: 02/03/89

********none********	64288S004	01.10	Incorrect disassembly of traces	5000441485*	99
	64288S004	01.10	Monitor commands may not complete execution correctly with target sys.	D200098400*	99
	64288S004	01.10	The emulator does not configure the sys. config. register properly.	D200103077*	99

- PROM PROGRAMMER - SSB ISSUE DATE: 02/03/89

********none********	64501	01.10	Intel D2764A PROMs blowing up	1650074567*	101

- PROM PROGRAMMER -3 SSB ISSUE DATE: 02/03/89

********none********	64501S004	01.50	Customer unable to program NMC27C32 EPROMs	5000430421*	102
	64501S004	01.30	PROM programmer has problems in UX envr programming 32 bit system.	5000240952	363
	64501S004	01.30	Adden an enhancement to the checksum stuff. checksum file.	D200096214*	102

- ROM EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64272	01.04	store command generates 16-bit width absolute file only	5000231571	364
	64272	01.04	Some 64502A revisions cause the 64272 to fail its option test	5000429795*	103

- RS-232 TRANSFER -3 SSB ISSUE DATE: 02/03/89

********none********	64885	01.30	Break or ^C may not abort a foreground transfer with a file list	D200093906	365
	64885	01.30	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093930	365
	64885	01.30	Output warning msg. if number of ndilbuffers is <13	D200098665*	104

- RS-232 TRANSFER -5 SSB ISSUE DATE: 02/03/89

********none********	64884	01.40	Break or ^C may not abort a foreground transfer with a file list	D200093898	366
	64884	01.40	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093922	366
TRANSFER	64884	01.10	Transfer hangs after bad options message is displayed.	D200065219	366

- RS-232 TRANSFER -V SSB ISSUE DATE: 02/03/89

********none********	64886	01.60	Recompile and relink on VMS 5.0 with 5.0 runtime libraries	D200102814*	105

- RS-232 TRANSFER -V SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
TRANSFER	64886	01.10	Inaccurate specification in HELP for TRANSFER command	5000194951	367

- SOFTKEY EDITOR - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64790-90901	01.00	The find command does not work correctly, cannot find string includ '$'.	5000211375	368
	64790-90901	01.00	AND '$' NEEDS TO BE ESCAPED ON COMMAND LINE TO PREVENT SHELL EXPANSION	D200089896	368

- SOFTKEY EDITOR - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64790S004	02.10	Status line does not change after file is written for the save command.	5000401349	369
	64790S004	02.10	sk editor replace command does not work properly with anystring (*).	5000401372	369
	64790S004	02.10	When retrieving enough lines to get file exactly 1024 in size; core dump	D200090241	370
	64790S004	02.00	When 4 retrieves are done, the sk editor jumps to shell.	1650061580	369

- STATE 80386 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64659	01.00	SPECIFYING DISASSEMBLER IS CUMBERSOME	5000417063*	106

- STATE ANALYZER - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64620	00.71	Source referencing will not work with non-zero segments (8086, etc)	D200075028	371

- STATE ANALYZER - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64620S004	01.10	File names <8 chars in link_sym will cause translate problems	D200091538	372

- SW PERF ANALYZER - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64310	01.11	"show curr_meas" after measurement change crashes station.	5000122374	373

- SW PERF ANALYZER 300 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64310S004	01.20	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080176	374
	64310S004	01.20	Using Emulation across RFA can give incomplete symbol information	D200081026	374
	64310S004	01.20	Processes sometimes left running after parent has stopped.	D200082347	374

- TIMING ANALYZER -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64610S004	01.60	Processes sometimes left running after parent has stopped.	D200082370	376

- TIMING/STATE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64610	01.00	label cannot be deleted in trigger specification	5000089359	377

- TMS32020 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64786	01.02	A ROVM instruction following a DINT is skipped when single stepping	D200097626*	107

- TMS320C25 - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64787	00.01	Data words at address 6 & 7 can apparantly be displayed and modified	D200090522	378

- UPROG - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64276	01.00	IN UP_CNTL,"LIST TRACEDATA" SHOWS "AND" EVEN IF NO "ABSOLUTE IS"	D200035261	379
	64276	01.00	IN UP_CNTL, NO ERRMSG ISSUED IF "RUN UNTIL W/JAM ATTEMPTED W/O JAM LABEL	D200035287	379

- USER DEF ASSEMB - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851	00.70	Expand Directive not working on 64000.	5000251322	380
	64851	00.70	Duplicate Symbols in Symbols Declarations not flagged as an error.	D200068924	380
	64851	00.70	Duplicate SYMBOLS Definitions are not flagged as an error	D200068932	380
	64851	00.70	Bad table code generated when more than 25 SYMBOLS definitions	D200068940	381
	64851	00.70	REPT will only take arguments range 1 thru 32767	D200089409	382
	64851	00.70	Page size is different between PI and Hosted assemblers	D200089433	382
	64851	00.70	line number only 16-bits in size... This is too small for long files.	D200089458	382
	64851	00.70	COPY :asmb_sym to display behaves like disc_image on.	D200092619	382
CODE GENERATOR	64851	00.70	High order bits stripped from source characters in Pisces I	D200079376	381
	64851	00.70	Error Message ", errors= " will appear on listing.	D200094078	383

- USER DEF ASSEMB -3 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851S004	02.20	UDA produces "core dump" when external is used.	5000409102	384
	64851S004	02.20	Tabs converted to spaces in assembly code.	D200094565	385
	64851S004	02.10	Undefined Error placed on all macro usage, if just one label undefined	5000294181	384
	64851S004	02.10	ORG 10000H will change address of next line, but not those following.	5000417790	384
	64851S004	02.10	Linker ERROR messages go only to standard error.	5000419440	384
	64851S004	02.10	DE errors anr not declared in all cases for forward references.	D200087569	385

- USER DEF ASSEMB -5 SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851S001	02.20	Not all Parameters in linker config. table read.	D200095075	388
	64851S001	02.10	DE errors anr not declared in all cases for forward references.	D200087544	387
	64851S001	01.50	Conditional assembly for INCLUDE files causes error.	D200065391	387
CODE GENERATOR	64851S001	02.20	Problem with parameter passing in macros	D200093781	387

- USER DEF ASSEMB -D SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851S006	02.11	Can not assemble a file on a different disk...(ie: 'A:' from C:)	D200091272	390

- USER DEF ASSEMB -D SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851S006	02.11	Assembler crashes when directory path name is too long	D200091314	390

- USER DEF ASSEMB -V SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851S003	02.10	DE errors anr not declared in all cases for forward references.	D200087551	391
	64851S003	01.50	Conditional assembly for INCLUDE files causes error.	D200065409	391

- USER DEF EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64274S004	01.20	64000-UX "configude" executes utils using $PATH, not $HP64000	D200094656	392
	64274S004	01.20	Emulation core dumps when run in a small window.	D200095760	392
	64274S004	01.20	"Copy to Read-Only files", fails to deliver an error message to screen	D200096057	392
	64274S004	01.20	DEFAULT_STATUS can not be set to a combination of 0's and X's.	D200096206	392
	64274S004	01.20	Emulation hangs on shell commands that run quickly.	D200103358*	108

- USER DEF EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64274	01.06	UDE displays incorrect data in emulation memory when read fails	D200097071*	109

- USER DEFIN ASM - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851-90904	01.00	DE must be defined before being referenced.	5000153981	393
	64851-90904	01.00	Assembler reference manual should explain EQU and MACROs better.	D200094854*	110
	64851-90904	00.70	64000 station resets when linking if SKELETON command used improperly	D200079558	393

- USER INTERFACE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64808-90901	01.00	Need to add Note saying that 64100 Terminal Mode is not supported.	5000267468	394
	64808-90901	01.00	PMON doesn't allow a file to begin with a numeric value.	5000291427	394
	64808-90901	01.00	Two points that need to be indicated on pg 3-5 of 64808-90901 manual.	5000413161	394

- USER INTERFACE - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64808S004	02.10	A command file containing these three characters in that order #'! fails	D200090613	395
	64808S004	02.10	Pmon flags legitimate option for lnk (for 64859) as syntax error	D200099390*	111

- UTILITIES PKG - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64888S001	01.20	A HP-UX directory can be destroyed by transfer (1)	D200092502	396

- VMS 68000/8/10 ASM M - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64845-90906	01.30	LR error flagged for correct offset using PC+INDEX+OFFSET mode of addr.	D200046268	397

- VMS 68000/8/10 ASM M - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
MANUAL	64845-90906	01.60	Manual explains linker options incorrectly.	5000220764	397
	64845-90906	01.60	Manual states incorrectly that EXT is a pseudo op.	5000220772	397

- VMS FILE FORMATS - SSB ISSUE DATE: 02/03/89

********none********	64882-90903	01.02	VAX file format manual doesn't give clear explantion of VAX file types.	5000238543	399

- VMS SYSTEM INSTAL -M SSB ISSUE DATE: 02/03/89

********none********	64882-90904	01.03	Need setting for rear panel of old HP 64000 and 64110(with jumper jacks)	D200055202	400

- VMS USERS GUIDE - SSB ISSUE DATE: 02/03/89

********none********	64882-90902	01.60	Page 3-9 states vt52 emulation using 64100 but does not perform functs.	5000222489	401
	64882-90902	01.01	Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.	D200045492	401

- Z8 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64850	00.01	Assembler generates Phase Error of forward referenced EQU	D200091645	402
	64850	00.00	Assembler not generating error message when attempt to load label.	2700005918	402

- Z80 - SSB ISSUE DATE: 02/03/89

********none********	64753S006	01.01	The terminal window escape sequence is not available on German keyboards	1650069773	403

- Z80 ASSEMB - SSB ISSUE DATE: 02/03/89

********none********	64842	01.90	"/" character does not work as delimiter for lnk options	1650069765	404
	64842	01.12	Using HEX psuedo is causing bad address calculations.	5000139535	404
	64842	01.12	Xref lists symbols which are under False conditional assembly blocks.	5000239939	404
	64842	01.12	Difference between 64000 and host in XREF when no symbols.	D200086686	405
	64842	01.11	Revision number on output listing is incorrect.	5000152819	404
	64842	01.10	Complex macro interaction causing invalid errors.	5000264986	405

- Z80 EMULATION - SSB ISSUE DATE: 02/03/89

********none********	64252S004	01.00	HPIB 64120 I/O AND POWER FAILED WHEN MODIFYING TARGET MEMORY	1650047167	406
	64252S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069542	406
	64252S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080655	406
	64252S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080952	407
	64252S004	01.00	PC contents lost over continuation if in break state	D200081489	407
	64252S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081901	407

- Z80 EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64252S004	01.00	Processes sometimes left running after parent has stopped.	D200082230	408
	64252S004	01.00	Loading a trace file from a different processor may cause core dump	D200083246	408
	64252S004	01.00	IMPROPER IDENTIFICATION OF THE SECOND Z80 CONTRL CARD IF TWO Z80 PRESENT	D200085332	408
	64252S004	01.00	Tracelist symbols dissappear.	D200086033	408
	64252S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090860	409
	64252S004	01.00	Emulation core dumps when run in a small window.	D200095711	409
	64252S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200096008	410
	64252S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098350*	112
	64252S004	01.00	Emulation hangs on shell commands that run quickly.	D200103309*	112
	64252S004	00.00	EMULATION SOFTWARE STATUS DOES NOT RECOGNIZE THE "HALT" INSTRUCTION	1650047340	406

- Z80 PASCAL - SSB ISSUE DATE: 02/03/89

CODE GENERATOR	64812	00.00	$ORG directive can cause incorrect code to be generated.	1650041624	411

- Z80/NSC800 C - SSB ISSUE DATE: 02/03/89

********none********	64824	02.10	Indirect comparison of parameter bytes may fail	D200090175	413
	64824	02.10	Certain set operations with explicit type changes may fail.	D200090217	414
	64824	01.04	Real variable used as a test condition cause error.	D200081554	412
PASS 3	64824	01.03	Conditional compile fails if it suceeds a fixed parm function call.	D200069906	412

- Z80/NSC800 P - SSB ISSUE DATE: 02/03/89

********none********	64823-90901	01.00	Documentation and examples for Z80 I/O port	5000170191	415

- Z80/NSC800PASCAL - SSB ISSUE DATE: 02/03/89

********none********	64823	02.00	Type casting the ADDR function to SET causes 1006 error	D200093823	419
	64823	01.90	Certain set operations with explicit type changes may fail.	D200090209	417
	64823	01.90	Incorrect code in complex parameter assignments	D200092593	418
PASS 1	64823	01.02	$Range ON$ causes incorrect code to be generated for a test operation.	D200059600	416
PASS 3	64823	01.02	Incorrect data offsets in listing file.	D200060186	416

- Z8000 C - SSB ISSUE DATE: 02/03/89

********none********	64820	01.06	Logical AND produces a multiply operation.	5000160671	420
	64820	01.06	Local parms not accessed properly when func called via pointer.	5000246983	422
	64820	01.06	Oversized data segment not being flagged as an error.	D200078873	426
	64820	01.06	Real variable used as a test condition cause error.	D200081521	428
	64820	01.06	Function calls via pointers with parameters mess up subsequent calls.	D200085381	429
	64820	01.05	Superfluous register load in switch statement on the 64000	D200064808	424
	64820	01.05	Illegal initialization causes error 1113.	D200068155	425
	64820	01.04	$OPTIMIZE$ compiler directive works differently for signed and unsigned.	5000181545	421

- Z8000 C - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64820	01.04	Inconsistient error message when linking ASM.R files versus COMP.R files	D200061762	423
	64820	00.01	Code generated for unsigned multiply is the same as for signed multiply.	1650006544	420
PASS 3	64820	01.05	Conditional compile fails if it suceeds a fixed parm function call.	D200069781	426
PROBLEM ON 9000/S300	64820	02.10	Compiler does not create an 'array too large' error when size > 32k.	5000280958	423
	64820	01.06	If condition is tested with a CMP D1,D1	D200079616	427

- Z8000 PASCAL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64816	01.90	Large sets may produce invalid results for elements outside set range	D200093658	434
	64816	01.90	Type casting the ADDR function for masking may cause error #1006	D200093666	436
	64816	01.90	Boolean Index into array generates bad code	D200093674	437
	64816	01.12	$RANGE$ & type conversion of UNSIGNED_32 var may cause error 1006.	D200085282	434
	64816	01.11	BA address mode may attempt to use RR0 Illegally as source	D200073015	434
	64816	01.10	Inconsistient error message when linking ASM.R files versus COMP.R files	D200061721	433
	64816	01.09	Jump table may generate code which accesses wrong data space.	5000123497	430
	64816	01.04	"Downto" used in a for statement generates incorrect code.	5000150151	431
PASS 3	64816	01.10	Calling func. twice in statement causes return value to be overwritten	5000134916	430
	64816	01.10	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060145	432

- Z8001 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64232S004	01.00	Emulation hangs on shell commands that run quickly.	D200103234*	113

- Z8001/2 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64980-90923	01.00	Need more info on sharing user system calls & monitor interaction.	5000131573	438

- Z8002 EMUL - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64233S004	01.00	Emulation hangs on shell commands that run quickly.	D200103242*	114

- Z80H EMULATION - SSB ISSUE DATE: 02/03/89

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64253S004	01.00	CANNOT ACCESS COMPILER GENERATED SYMBOLS IN HP64000-UX EMUL ENVIRONMENT	5000258616	439
	64253S004	01.00	Emulation core dumps when run in a small window.	D200095729	439
	64253S004	01.00	"Copy to Read-Only files", fails to deliver an error message to screen	D200096016	439
	64253S004	01.00	Monitor commands may not complete execution correctly with target sys.	D200098368*	115
	64253S004	01.00	Emulation hangs on shell commands that run quickly.	D200103317*	115

KPR #: D200098178 Product: 6301V EMULATION 300 64206S004 01.20

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103127 Product: 6301V EMULATION 300 64206S004 01.20

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098186 Product: 6301X EMULATION 300 64207S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103135 Product: 6301X EMULATION 300 64207S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098194 Product: 6301Y EMULATION 300 64208S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103143 Product: 6301Y EMULATION 300 64208S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200099523 Product: 64000-UX OP-ENV 300 64801S004 02.40

One-line description:
'edbuild' involked from within a product may execute private command.

Problem:
Detailed Listing for Defect Number LSDqf05425
Text:
'edbuild' invoked from w/in product may execute private command
.....
When a product does an 'edbuild' (such as in emulation, when the user does a 'load file', and he has not done an 'edbuild file'), the edbcreate() routine uses the users PATH variable to determine the absolute path of 'edbuild'.

If the user has a command 'edbuild' in his PATH prior to /usr/hp64000/bin, the user's 'edbuild' will be executed instead of the /usr/hp64000/bin/edbuild

WORKAROUND:
1) make certain that /usr/hp64000/bin comes before any personal versions of edbuild.

2) Do not create personal commands called 'edbuild'

Bruce Erickson
.labnotes

This problem is in the file 'skedbcreat.c'
It should create the result of
${HP64000:-/usr/hp64000/bin}/edbcreate
and send that result to shellexec(), instead of just sending 'edbcreate' to shellexec().

- Bruce E
.workaround

The command 'edbuild' is peculiar to the HP64000 system; therefore, if an incorrect 'edbuild' is being executed, the end-user has created it.
Solution: don't create personal commands with the name 'edbuild'...

KPR #: 5000430900 Product: 6800/2 ASSEMB 64841 01.80

Keywords: MACROS PROBLEM ON 9000/S300

One-line description:
MACRO expansion of constants fails.

Problem:

```
          1 "6303"
          2 LDCMPA     MACRO  &A0,&B1
          3            LDAA   &A0
          4            CMPA   &B1
          5            MEND
          6            LDCMPA  #50H,#80H
    8650  +            LDAA    #50H
    9180  +            CMPA    80H
    ^                        ^
    |                        |----- this sould be #80H
    |---- this sould be 8180 .
```

KPR #: 5000436360 Product: 6800/2 ASSEMB 64841 01.80

One-line description:
JMP to EQU'd label fails.

Problem:
JMP instruction generates incorrect codes.

EXAMPLE

```
               1 "6801"
               2
    <0012>     3 LABEL         EQU          12H
0000 7E12      4               JMP          LABEL
```

OPcode '7E' requires 2 bytes codes as jump address. So incorrect address is read.

KPR #: D200102368 Product: 68000 12MHZ EMUL FW 64742 00.05

One-line description:
Entering the command "cf mon=bg" many times produces an error message.

Problem:
Detailed Listing for Defect Number LSDqf05470
Text:
  entering command "cf mon=bg" many times produces error msg.
.....
If the command "cf mon=bg" is entered several times in succesion (greater than 7) an error message stating that the error stack is full is produced.

.labnotes

The function cfmon in file funcs/config.c has been fixed. In cases where the value of the monitor type configuration variable was already equal to the requested type the error stack was not correctly popped.
This fix will be included in firmware that supports both 128 kbytes and 512 kbytes of emulation memory. This firmware will require the use of CPU/control board 64762-66507 or later.

KPR #: D200097212 Product: 68000 12MHZ FUI DOS 64742S006 01.02

One-line description:
Bringing up multiple windows can cause demon timeout.

Problem:
bringing up multiple windows can cause demon timeout
.....
Starting four windows two seconds or so apart can cause one or more of the interfaces to exit with the following messages:

ERROR: Timeout in emul700dmn communication
fast: Status unknown, run "emul700 -l fast"

["fast" is the logicalname of the emulator]

Temporary solution:
Workaround is to restart the window that failed due to the timeout.

KPR #: D200098855 Product: 68000 12MHZ PTUI 300 64742S004 01.00

One-line description:
Deamon hangs when unable to communicate to 64700 pod

Problem:
Detailed Listing for Defect Number LSDqf05375
Text:
Daemon hangs when unable to communicate to 64700 pod
.....
When the user disconnects the 64700 pod while ptui is running, ptui forces the user to end_release. After ptui has ended, the daemon is still running.

This problem exists for all currently released PTUI's.

- Cheryl Brown

.submitter

If the communications cable is disconnected from a 64700 series emulator while the user interface is running, the interface forces an "end released" command which terminates the user interface. However, the communications daemon that handles communication to the emulator does not die. This process must be killed manually.

This effects the following released products:

64742S004
64745S004
64764S004
64765S004

KPR #: 5000422394 Product: 68000 ASSEMB 64845 02.10

One-line description:
Absolute Long code is not generated correctly.

Problem:
Absolute Long code is not generated correctly.
"68000"

```
      ORG     0130100H
      ABS_LONG
      MOVE.B  000FFFE00H,D0   --> 1038FE00 (Abs.W) code is generated.
      MOVE.B  000FF0000H,D0   --> 103900FF0000 (Abs.L) code is generated
      MOVE.B  VTEST,D0        --> 103900FFFE00 (Abs.L) code is generated
      ORG     0FFFE00H
VTEST EQU     $
      MOVE.L  #288H,D1
```

Temporary solution:
Use a label as an operand.

KPR #: D200098897 Product: 68000 BBA 300 64380S004 01.10

One-line description:
Misc. severe syntax errors cause "Fatal error in ...bbacpp"

Temporary solution:
WORKAROUND:
Fix the syntax error and recompile. Note that the compiler will also show a syntax error but will not core dump.

KPR #: D200098913 Product: 68000 BBA 300 64380S004 01.10

One-line description:
Comments do not force seperator

Temporary solution:
WORKAROUND:
place a space before of after the comment:

```
a /* comment */b
     - or -
a/* comment */ b
```

KPR #: D200098939 Product: 68000 BBA 300 64380S004 01.10

Keywords: CODE GENERATOR

One-line description:
Struct field and typedef cannot have the same name

Temporary solution:
WORKAROUND:
rename either the typedef or the structure name. Note that the AxLS compilers have the same problem.

KPR #: 5000429126 Product: 68000 C 64819 02.10

One-line description:
unsigned long i; gets EXT.L when i = 0x8000;

Problem:
Compiler produces incorrect code.

```
"C"
"68000"
unsigned long i;
main()
{
  i = 0x8000;
}
```

Using the "options expand" one can see that the compiler generates incorrect code if "i" is assigned a value from 8000H to 0FFFFH (inclusive). To do the assignment the compiler moves the word value to a data register, sign extends to a long word (this is the problem - it should NOT do this sign extend for an unsigned integer!) and then moves this long word to the destination location.

Temporary solution:
For values of "i" less than 8000H this code is alright. For values of "i" greater than 0FFFFH the compiler will generate a MOVE.L immediate instruction which is also alright.

KPR #: D200098319 Product: 68000 DQ EMUL 300 64243S004 01.30

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103267 Product: 68000 DQ EMUL 300 64243S004 01.30

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: 1650049650 Product: 68000 EMUL 12.5 MHZ 64243 01.01

One-line description:
68000 inverse assembler doesn't recognize the opcode "ILLEGAL"

Problem:
IF YOU WRITE A PROGRAM INCLUDING THE NEMONIC INSTRUCCION "ILLEGAL" AND YOU TRACE THAT PROGRAM YOU WILL SEE THAT THE INVERSE ASSEMBLER OF THE 68000dq EMULATOR DOESN(T RECOGNIZE THAT OPCODE AND SHOWS A DIFFERENT PROGRAM WITH OTHER NEMONICS INSTEAD OF "ILLEGAL".

KPR #: 5000281261 Product: 68000 EMUL 12.5 MHZ 64243 01.01

One-line description:
Modify memory does not generate correct IEEE format value

Problem:
Modify memory commnad does not generate correct IEEE format value.
The following command shows the sample;
modify memory real short 1000H to 12.54
The real number 12.54 should be 41473300H in IEEE 32bit format. But emulator assigns 41473333H to 12.54.

KPR #: 5000264523 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
The emulation monitor program may hide target system bus errors.

Problem:
The emulation monitor program may hide target system BUS ERRORs under certain situations. The following test was run on a 64242S004 emulator using the CSA demo box. Consider the following scenario:
- The system is running user code, and a modify memory 20000H to 55H is issued. (20000H returns a BE signal when accessed)
- The emulator breaks the processor into monitor, then issues the MOVE_MEM command which does a MOVE.B [A0]+,[A1]+.
- The bus error occurs, and the PCL stacked is not the address of the SUBI.W #001,D3, but is the address of the 0001H operand of that same SUBI instruction.
- The monitor BE_ENTRY routine is executed, and the check for a re-entrant condition finds that this a re-entrant situation. It therefore does an RTE, returning to execute the 0001H as an opcode.
- Coincidentally, the 0001H and the following byte execute as a ORI.B #F8,D1, using the BNE that follows the SUBI.W as an operand.
- The next instruction is the JMP LOOP_REENTRY, which signals to the emulation software that the operation was a success!!!!!! The operation was in fact not successful, and if this was a modify 20000H thru 200ffh to 55H, the operation would stop after the access to 20000H.
- In this scenario no bus error message has been displayed. (although if the code was running in the monitor initially, the "--- bus error---" message would be displayed)
- In a display memory, a similar situation occurs, but instead of the address of the operand of the SUBI.W #0001,D3, the opcode address is stacked. This means that the SUBI.W will be executed, and all the addresses may be accessed. This leads to the possibility of a user issuing:"modify memory 20000H thru 2003FH to 55H", followed by: "display memory 20000H" and seeing 55H in the locations he/she expected, even though each access resulted in a bus error.

NOTES:
There are several things that make this scenario possible. One important point in this test is that the bus error timeout for the CSA is relatively short (approx 8us). This allows many bus errors to occur in a short time period. Another target system with a longer bus error timeout will see different results, because after some time period of not seeing MONITOR_CONTROL cleared the emulation software will inidicate a failure occured during the target memory access.

KPR #: D200098301 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

KPR #: D200098301 **CONTINUED**

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103259 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: 5000421230 Product: 68000C AXLS COMP M 64902-90901 01.00

One-line description:
HP64000 variable cannot be set to a \net directory. Manual fix requested

---

KPR #: D200098574 Product: 68000C AXLS COMP M 64902-90901 01.00

One-line description:
Manual needs to have more info on libraries.

Problem:
The manuals need to contain more information concerning the libraries provided. Many customers do not wish to use the default linker command file, and therefore attempt to write their own. The libraries are not understood well enough and many customers are frustrated when they find the only linker command file that works with their application is the default.
The following is a list of concerns that need to be addressed:

1. Most of the libraries are required - but not desired. It should be stressed that the customer can write his own functions for the "unresolved externals" and then not load the libarary that declares it.

2. The order in which the libraries are loaded is critical and must be discussed.

3. A better picture of just what is contained in which library would be helpful.

4. What other information in the default linker command file is critical.

KPR #: 5000422485 Product: 68000C AXLS COMP 300 64902S004 02.00

One-line description:
-u causes erroneous warning to be generated.

Problem:
Customer would like to change the way the "-u" option to the cc68000 compiler generates errors. Consider the following example:

```
struct a {
         int b;
         int c;
        };
const struct a str_a[] = {{ 1,2 }, { 3,4 }};
```

When compiled without the -u option, no errors or warnings are generated, but when the -u option is used, the following warning:
"file.c" 5:warning- Static initializer will not be loaded
appears for each of the 4 elements that are initialized. Because this is declared with the "const" type modifier, no warnings should be generated.

KPR #: 5000435669 Product: 68000C AXLS COMP 300 64902S004 02.10

One-line description:
Division with mixed types can generate incorrect code with -O.

Problem:
The 68000 C compiler generates incorrect code for the following if the "-O" optimize option is used.

```
short res,den;
long num;

main()
{
res = num/den;
    MOVE.L  (_num+0).L,D0
    DIVS.W  (_den+0).L,D0
    MOVE.W  D0,(_res+0).L
}
```

If the result of the division is greater than 16-bits an overflow is generated and the division does not occur. Without the -O option the compiler correctly calls a library routine to do a 32-bit division.

Temporary solution:
1. Don't use the -O option.
2. Change the variable "res" to a long and then perform an cast to generate the short.
3. Cast the "short" to a "long" in the division expression such as:

```
res = num/(long)den;
```

KPR #: 5000439216 Product: 68000C AXLS COMP 300 64902S004 02.00

One-line description:
Division of unsigned variable by a constant.

Problem:
something - (unsigned_variable / constant)

fails.

Temporary solution:
Insert a "+" sign after the "-" such as:

```
something - +(unsigned_variable / constant)
```

or perform the evaluation in two steps:

```
temp = unsigned_variable / constant;
something - temp;
```

KPR #: D200098806 Product: 68000C AXLS COMP 300 64902S004 02.00

One-line description:
"LOCAL" pseudo not useable inside #pragma ASM.

Problem:
Compiler generates comment lines between MACRO directives and LOCAL directives.

```
func(){
#pragma ASM
jump    MACRO   _______________     Comment lines are inserted here by
        LOCAL     LAB1              the compiler.
LAB1     nop
         bra.\0  LAB1
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

As the result, as68k generates "**ERROR:(553) Duplicate label...".

Temporary solution:
Use uniquely generated labels from the "\@" construction:

```
func(){
#pragma ASM
jump    MACRO
LAB1\@   nop
         bra.\0  LAB1\@
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

KPR #: D200098764 Product: 68000C AXLS COMP 800 64902S005 02.00

One-line description:
-u causes erroneous warning to be generated.

Problem:
Customer would like to change the way the "-u" option to the cc68000 compiler generates errors. Consider the following example:

```
struct a {
         int b;
         int c;
        };
const struct a str_a[] = {{ 1,2 }, { 3,4 }};
```

When compiled without the -u option, no errors or warnings are generated, but when the -u option is used, the following warning:
"file.c" 5:warning- Static initializer will not be loaded
appears for each of the 4 elements that are initialized. Because this is declared with the "const" type modifier, no warnings should be generated.

KPR #: D200098814 Product: 68000C AXLS COMP 800 64902S005 02.00

One-line description:
"LOCAL" pseudo not useable inside #pragma ASM.

Problem:
Compiler generates comment lines between MACRO directives and LOCAL directives.

```
func(){
#pragma ASM
jump    MACRO  _______________   Comment lines are inserted here by
        LOCAL    LAB1            the compiler.
LAB1     nop
         bra.\0  LAB1
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

As the result, as68k generates "**ERROR:(553) Duplicate label...".

Temporary solution:
Use uniquely generated labels from the "\@" construction:

```
func(){
#pragma ASM
jump    MACRO
LAB1\@   nop
         bra.\0  LAB1\@
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

KPR #: 5000274506 Product: 68008 EMULATION 64244 01.01

One-line description:
Unknown software bp error message after software bp is set.

Problem:
Customer is using the 64244 MC68008 Emulator in the 64100 environment and having problems with software breakpoints. He is running rev 2.09 of the 64100 op sys and rev 1.01 of the emul sw. The following can be used to reproduce the problem:

```
"68008"
             PROG      ;  link at 1000H
             JMP.W  0FFFF8000H
             DATA      ;  link at 0F8000H
             NOP
LOOP         NOP
             BRA  LOOP
             END
```

Assemble and link (with the monitor configured for software breakpoints) at locations specified.

Program is loaded, breakpoint is set at address 0F8000H and program is run. Get an error message - unknown software breakpoint is encountered at address 0F8000H.

KPR #: D200098327 Product: 68008 EMULATION 300 64244S004 01.30

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103275 Product: 68008 EMULATION 300 64244S004 01.30

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098376 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103325 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200097204 Product: 68010 16MHZ FUI DOS 64745S006 01.02

One-line description:
Can not run with foreground monitor and standard SSP

Problem:
Detailed Listing for Defect Number LSDqf05030
Text:
cannot run with foreground monitor and standard SSP
.....
Using the background monitor, standard memory map (0 thru 1FFFH emulation ram), and standard SSP for reset (1FFEH), a test file runs fine. Switching to foreground monitor (and obviously resetting the memory map and reloading the program) causes the run to fail - it claims the stack is in guarded memory. Display registers still shows SSP to be 1FFEH which is certainly NOT guarded memory. The stack pointer should decrement as things are pushed, so why doesn't it work?

KPR #: D200098863 Product: 68010 16MHZ PTUI 300 64745S004 01.00

One-line description:
Deamon hangs when unable to communicate to 64700 pod

Problem:
Detailed Listing for Defect Number LSDqf05375
Text:
Daemon hangs when unable to communicate to 64700 pod
.....
When the user disconnects the 64700 pod while ptui is running, ptui forces the user to end_release. After ptui has ended, the daemon is still running.

This problem exists for all currently released PTUI's.

- Cheryl Brown

.submitter

If the communications cable is disconnected from a 64700 series emulator while the user interface is running, the interface forces an "end released" command which terminates the user interface. However, the communications daemon that handles communication to the emulator does not die. This process must be killed manually.

This effects the following released products:

64742S004
64745S004
64764S004
64765S004

KPR #: D200098335 Product: 68010 EMUL 12.5M 300 64245S004 01.30

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103283 Product: 68010 EMUL 12.5M 300 64245S004 01.30

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098343 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103291 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098566 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Register indirect with 8 bit displacement uses 3 words.

Problem:
The following code uses the Full Format (3 words) instead of the Brief Format (2 words):

```
(object code)          (source)
-------------------------------------------------------------
                       CHIP  68020
3A33 2520 0004         MOVE.W  (4,A3,D2.W*4),D5
                       END
```

The lab is aware of the problem and has a related DTS report.

Temporary solution:
The code generated is correct, however it could be written in a word less. No known workaround at this time.

KPR #: D200098905 Product: 68020 BBA 300 64381S004 01.20

One-line description:
Key=Z470 Misc. severe syntax errors cause "Fatal error in...bbacpp"

Temporary solution:
WORKAROUND:
Fix the syntax error and recompile. Note that the compiler will also show a syntax error but will not core dump.

KPR #: D200098921 Product: 68020 BBA 300 64381S004 01.20

One-line description:
Comments do not force seperator for variable

Temporary solution:
WORKAROUND:
place a space before of after the comment:

```
a /* comment */b
      - or -
a/* comment */ b
```

KPR #: D200098947 Product: 68020 BBA 300 64381S004 01.20

One-line description:
struct field and typedef can not have the same name

Temporary solution:
WORKAROUND:
rename either the typedef or the structure name. Note that the AxLS compilers have the same problem.

KPR #: 1650048652 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
If memory mapping is not contiguous, program doesn't load properly.

Problem:
IF THE MEMEORY ENTRIES ARE NOT CONTINGOUS THE PROGRAM IS NOT LOADED PROPERLY.
THE SAMPLE PROGRAM WAS USED IN CHAPTER 1 (64411 APR87) ALONG WITH THE MONITOR PROGRAM OF REV 2.0 SOFTWARE AND THE MEMORY MAP WAS SET UP AS FOLLOWS:
0 THRU 0FFH ERAM WIDTH 32 BIT
1000H THRU 18FFH ERAM WIDTH 32 BIT
2000H THRU 20FFH ERAM WIDTH 32 BIT
2500H THRU 25FFH ERAM WIDTH 32 BIT
3000H THRU 3FFFH ERAM WIDTH 32 BIT
4000H THRU 3FFFFH ERAM WIDTH 32 BIT

THE COMMAND "DISPLAY MEMORY 2500h" (ADDRESS OF "INCREMENT SUBROUTINE) SHOWS WRONG DATA AND THE PROGRAM DOES NOT EXECUTE CORRECTLY.
HOWEVER WHEN THE ENTRY 2000H THRU 25FFH IS MADE TO THE MEMORY MAP THE PROGRAM IS LOADED PROPERLY.

KPR #: 5000275693 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
Improper overlay of emulation memory.

Problem:
The 68020 emulator will improperly overlay emulation memory under certain mapping conditions.
With a memory map that involves overlays, writing to one address, may result in modification of another address that is not overlayed on top of the modified address.

KPR #: 5000291765 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
slow target ram causes "partial load" error

Problem:
Revision 2.00 of the 68020 emulation software has introduced a problem when downloading a file to slow target memory. The customer has off-board memory running with 18 wait states. When he loads a file into this memory the emulator gives a "Partial Load" error message, but the file is indeed loaded fine. The error only occurs when he is loading the slow memory, if emulation or faster target memory is loaded then the error message does not occur.

KPR #: 5000296541 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
68020 monitor functions do not work properly when MSP is the active SP

Problem:
The 68020 emulator monitor functions do not work properly when the Master Stack Pointer is the active SP, and the processor is in the

KPR #: 5000296541 **CONTINUED**

USER state (S bit = 0).
If the User state is active, and the Master Stack Pointer is active, and a display registers is done, the MSP will be indicated as the active SP, and the status register will show that the processor is in the SUPERVISOR state, even though the processor is in the USER state.
Refer to page 2-20 of the MC68020 User's Manual for more info on use of the MSP.

KPR #: 5000403584 Product: 68020 EMUL 300 64410S004 02.10

One-line description:
Memory display does not work properly if function codes are enabled.

Problem:
THE MEMORY DISPLAY DOES NOT WORK PROPERLY IF FUNCTION CODES ARE ENABLED. IF A DISPLAY MEMORY COMMAND IS EXECUTED, THE FIRST PAGE IS SHOWN CORRECTLY. HOWEVER, IF THE DISPLAY IS PAGED THE FIRST LINE SHOWN IN THE NEXT PAGE IS INCORRECT. THE FIRST LINE SHOWN SHOULD ACUTALLY BE THE SECOND LINE.

KPR #: 5000403915 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
Malloc error: called from read memory.

Problem:
The error:
Malloc error: called from readmemory
may occur if the monitor is not completely mapped to emulation ram. The customer had the following config:
- in circuit
- 7D000H thru 7DFFFH emul ram ; 7E000H thru 7FFFFH target ram
- the mon_prog section was from 7D000H to 7D87BH
- the mon_data section was from 7D87CH to 7E06DH (6DH bytes in target)

A load of the program did report any errors, and the user was able to run the monitor program. However, when a display of target memory was attempted, the error occured. The message repeated several times and the keyboard would not respond, forcing the user to kill the emulation process.
The emulator should check for this improper mapping condition and indicate an appropriate error.

KPR #: D200092601 Product: 68020 EMUL 300 64410S004 02.10

One-line description:
Power to ICC must be cycled after updating from Version 2.00 to 2.10

Problem:
Detailed Listing for Defect Number LSDqf04288
Text:
Power to ICC must be cycled after updating from Version 2.00 to 2.10.

.submitter

KPR #: D200092601 **CONTINUED**

I have discovered a minor problem that may be encountered when updating the HP64410 68020 SW from Version 2.00 to 2.10. The problem WILL occur if the customer does NOT power down the cardcage during the update, and HAS 68020 measurement systems configured.

If the cardcage has not been powered down, and error message will be displayed ("Corupt module file ... ") the first time msinit is run. The message will also be displayed when the customer tries to enter the emulator, and the emulator will subsequently "end release".

The fix for this problem is quite simple: cycle the power to the card cage and then run msinit again. This will cause the module file to be rebuilt.

Temporary solution:
The fix for this problem is quite simple: cycle the power to the card cage and then run msinit again. This will cause the module file to be rebuilt.

KPR #: D200098418 Product: 68020 EMUL 300 64410S004 02.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103366 Product: 68020 EMUL 300 64416S004 02.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098582 Product: 68020C AXLS COMP M 64903-90901 01.00

One-line description:
Manual needs more info on libraries.

Problem:
The manuals need to contain more information concerning the libraries provided. Many customers do not wish to use the default linker command file, and therefore attempt to write their own. The libraries are not understood well enough and many customers are frustrated when they find the only linker command file that works with their application is the default.
The following is a list of concerns that need to be addressed:

1. Most of the libraries are required - but not desired. It should be stressed that the customer can write his own functions for the "unresolved externals" and then not load the libarary that declares it.

2. The order in which the libraries are loaded is critical and must be discussed.

3. A better picture of just what is contained in which library would be helpful.

4. What other information in the default linker command file is critical.

KPR #: 5000423129 Product: 68020C AXLS COMP 300 64903S004 02.00

One-line description:
"LOCAL" pseudo not useable inside #pragma ASM.

Problem:
Compiler generates comment lines between MACRO directives
and LOCAL directives.

```
func(){
#pragma ASM
jump    MACRO  _______________  Comment lines are inserted here by
        LOCAL    LAB1           the compiler.
LAB1     nop
         bra.\0  LAB1
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

As the result, as68k generates "**ERROR:(553) Duplicate label...".

Temporary solution:
Use uniquely generated labels from the "\@" construction:

```
func(){
#pragma ASM
jump    MACRO
LAB1\@   nop
         bra.\0  LAB1\@
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

KPR #: D200098772 Product: 68020C AXLS COMP 300 64903S004 02.11

One-line description:
-u causes erroneous warning to be generated.

Problem:
Customer would like to change the way the "-u" option to the
cc68000 compiler generates errors. Consider the following
example:

```
struct a {
         int b;
         int c;
        };
const struct a str_a[] = {{ 1,2 }, { 3,4 }};
```

When compiled without the -u option, no errors or warnings are
generated, but when the -u option is used, the following warning:
"file.c" 5:warning- Static initializer will not be loaded
appears for each of the 4 elements that are initialized. Because
this is declared with the "const" type modifier, no warnings should
be generated.

KPR #: D200098780 Product: 68020C AXLS COMP 800 64903S005 02.00

One-line description:
-u causes erroneous warning to be generated.

Problem:
Customer would like to change the way the "-u" option to the cc68000 compiler generates errors. Consider the following example:

```
struct a {
         int b;
         int c;
        };
const struct a str_a[] = {{ 1,2 }, { 3,4 }};
```

When compiled without the -u option, no errors or warnings are generated, but when the -u option is used, the following warning:
"file.c" 5:warning- Static initializer will not be loaded
appears for each of the 4 elements that are initialized. Because this is declared with the "const" type modifier, no warnings should be generated.

KPR #: D200098798 Product: 68020C AXLS COMP 800 64903S005 02.00

One-line description:
"LOCAL" pseudo not useable inside #pragma ASM.

Problem:
Compiler generates comment lines between MACRO directives and LOCAL directives.

```
func(){
#pragma ASM
jump    MACRO  _______________   Comment lines are inserted here by
        LOCAL    LAB1           the compiler.
LAB1     nop
         bra.\0  LAB1
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

As the result, as68k generates "**ERROR:(553) Duplicate label...".

Temporary solution:
Use uniquely generated labels from the "\@" construction:

```
func(){
#pragma ASM
jump    MACRO
LAB1\@   nop
         bra.\0  LAB1\@
        ENDM
        jump.B
        jump.W
#pragam END_ASM
}
```

KPR #: D200098152 Product: 6805 E EMUL 300 64195S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103101 Product: 6805 E EMUL 300 64195S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098145 Product: 6805 G EMUL 300 64194S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103093 Product: 6805 G EMUL 300 64194S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098137 Product: 6805 P EMUL 300 64193S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103085 Product: 6805 P EMUL 300 64193S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200097659 Product: 6805 U&R EMUL 300 64192S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200098640 Product: 6805 U&R EMUL 300 64192S004 01.10

One-line description:
Registers will not display on term with more than 58 lines

Problem:
Detailed Listing for Defect Number LSDqf05364
Text:
Registers will not display on term with more than 58 lines.
.....
If a display has more than 58 lines, "display registers" will not work. The header will be displayed, but no register data will be shown.

.labnotes
The problem is that the work area is declared to be 50 lines long. A wviewset() is then done for the size of the screen available for the work area. If the number of lines available is more than 50, the wviewset() fails since the work area is not large enough.

The solution is to declare the register work area to be the minimum of 50 lines long or the the number of lines available on the display to show the registers.

File: generic/registers/reg_disp.c
Function: buildregdisplayheader()

KPR #: D200098707 Product: 6805 U&R EMUL 300 64192S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to

KPR #: D200098707 **CONTINUED**

try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098202 Product: 6809 EMULATION 300 64215S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103150 Product: 6809 EMULATION 300 64215S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098210 Product: 6809E EMULATION 300 64216S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103168 Product: 6809E EMULATION 300 64216S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103341 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200095240 Product: 70108 EMUL 300 64295S004 01.10

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

KPR #: D200095216 Product: 70116 EMUL 300 64294S004 01.10

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

KPR #: D200095281 Product: 70208 EMUL 64297S004 01.00

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

KPR #: D200095265 Product: 70216 EMUL 64296S004 01.00

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

KPR #: D200097238 Product: 80186 FUI DOS 64764S006 01.02

One-line description:
Bringing up multiple windows can cause demon timeout.

Problem:

Text:
bringing up multiple windows can cause demon timeout

Starting four windows two seconds or so apart can cause one or more of the interfaces to exit with the following messages:

ERROR: Timeout in emul700dmn communication
fast: Status unknown, run "emul700 -l fast"

["fast" is the logicalname of the emulator]

Temporary solution:
Restart the window that failed due to the timeout.

KPR #: D200098871 Product: 80186 PTUI 300 64764S004 01.00

One-line description:
Deamon hangs when unable to communicate to 64700 pod

Problem:

Text:
Daemon hangs when unable to communicate to 64700 pod

When the user disconnects the 64700 pod while ptui is running, ptui forces the user to end_release. After ptui has ended, the daemon is still running.

This problem exists for all currently released PTUI's.

If the communications cable is disconnected from a 64700 series emulator while the user interface is running, the interface forces an "end released" command which terminates the user interface. However, the communications daemon that handles communication to the emulator does not die. This process must be killed manually.

This effects the following released products:

64742S004
64745S004
64764S004
64765S004

Duplicate Service Requests: D200098889

KPR #: D200102640 Product: 80186 PTUI 300 64764S004 01.00

One-line description:
Performance measurements do not work with non zero segments.

KPR #: D200103200 Product: 80186 EMULATION 300 64224S004 01.20

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103549 Product: 80186 EMULATION 300 64224S004 01.20

One-line description:
Running w/o monitor will corrupt the offset value on INT 0 vector.

Problem:
Running the emulator from reset without a monitor loaded will cause the value 0080h to be written to address 0h. This location is the offset value of the vector for interrupt 0 (divide by zero).

The problem is that when the state of the emulator is changed from RESET to RUNNING the routine check_state() is called. This calls i80xx_monitor_status() which does an ARE_YOU_THERE. If the monitor control word is not defined, this i80xx_monitor_command() should return FAILED before doing the ARE_YOU_THERE (since the monitor control word is undefined, 0080h is being written to address 0h).

Temporary solution:
There are two work arounds. The first is to make sure a monitor is loaded. This is the easiest. If a monitor cannot be loaded, then the global symbol "MONITOR_CONTROL" must exsist and be a word location in emulation memory that can be written to without messing anyting up.

KPR #: 1650072553 Product: 80188 EMULATION 300 64225S004 01.02

One-line description:
Incorrect reads and writes to odd target system memory locations

Duplicate Service Requests: 1650075119

KPR #: D200099036 Product: 80188 EMULATION 300 64225S004 01.20

One-line description:
Unable to download large amounts of code to target memory

KPR #: D200103218 Product: 80188 EMULATION 300 64225S004 01.20

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103556 Product: 80188 EMULATION 300 64225S004 01.20

One-line description:
Running w/o monitor will corrupt the offset value on INT 0 vector.

Problem:
Running the emulator from reset without a monitor loaded will cause the value 0080h to be written to address 0h. This location is the offset value of the vector for interrupt 0 (divide by zero).

The problem is that when the state of the emulator is changed from RESET to RUNNING the routine check_state() is called. This calls i80xx_monitor_status() which does an ARE_YOU_THERE. If the monitor control word is not defined, this i80xx_monitor_command() should return FAILED before doing the ARE_YOU_THERE (since the monitor control word is undefined, 0080h is being written to address 0h).

Temporary solution:
There are two work arounds. The first is to make sure a monitor

KPR #: D200103556 **CONTINUED**

is loaded. This is the easiest. If a monitor cannot be loaded, then the global symbol "MONITOR_CONTROL" must exsist and be a word location in emulation memory that can be written to without messing anyting up.

KPR #: D200103374 Product: 80386 EMUL 64420S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098384 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103333 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098160 Product: 8085 EMULATION 300 64203S004 01.40

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103119 Product: 8085 EMULATION 300 64203S004 01.40

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098426 Product: 8086 EMUL FW 64762 00.02

One-line description:
Floating point disp from "nreg" usb't as accyrate as it should be.

Problem:
The output of the command "nreg" gives values for the floating point registers which are not completely accurate. For example, the number "2" is shown as "1.99999999999999999983". Numbers which can be represented exactly in IEEE-754 80 bit format should be displayed with full precision.

Fix information:
Two errors in the function float_to_ascii() have been fixed to solve this problem. This function is found in the funcs/fpdisplay.c file. The regression tests will be updated accordingly.

This fix is implemented in the firmware that will be shipped with all 808X emulators after the introduction of the 512kbyte version of the 8086 (64762B). This firmware will require the use of the new control/CPU board 64762-66507.

KPR #: D200098434 Product: 8086 EMUL FW 64762 00.02

One-line description:
Analysis trace display does not distinguish bg from fg coproc. cycles.

Problem:
The traces display of the 8086/8088 emulators does not distinguish foreground 8087 memory cycles which occur after the emulator has gone to background from 8087 memory cycles which are the result of monitor operations and should show as background cycles.

Fix information:
The inverse assembler has been modified to differentiate between foreground and background coprocessor cycles. A hardware change was also necessary to implement this fix.

This fix will be included in the firmware that will be used after the release of the 512kbyte version of the 8086 (64762B). This firmware requires the use of the new control/CPU board 64762-66507.

KPR #: D200102780 Product: 8086 EMUL FW 64762 00.02

One-line description:
IAL incorrectly disassembles and displays some instructions.

KPR #: D200102376 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Should not be able to take offset of reg expression

Problem:
As86 does not error when the offset of a variable which is part of a register expression is taken.
It should not be possible to take the offset of a variable if it is part of a register expression.
Some examples are as follows:

```
8A 1F                   mov bl, offset byte ptr su1[bx]
8A 1F                   mov bl, offset cs: byte ptr su1[bx]
8A 18                   mov bl, offset byte ptr su1[bx][si]
8A 18                   mov bl, offset byte ptr su1[bx+si]
8A 5F 05                mov bl, offset byte ptr su1[bx+5]
8A 58 05                mov bl, offset byte ptr su1[bx+si+5]
8A 58 05                mov bl, offset byte ptr su1[bx][si][5]
B3 00                   mov bl, offset cs: su1
```

KPR #: D200102384 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Don't allow seg overrides on constants

Problem:
As86 allows a segment override to be applied to a constant.
The override is ignored in such an instance.
An error really should have been generated in these cases.
Examples of the instructions are as follows:

```
        mov bl, cs: 'a'
        mov bl, cs: 3
```

KPR #: D200102392 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
EQ, And, etc. shouldn't allow mem and abs as operands

Problem:
If ds: is applied to a constant and the result is a memory location, then it should not be possible to OR, XOR, or AND that value with a constant. Likewise, in the EQUATES, it should not be possible to use the memory location in a test against a constant.
As86 allows this.
Examples of this are as follows:

```
B8 07 00                mov ax, (ds: 5) or 3
B8 07 00                mov ax, ds: 5 or 3
B8 07 00                mov ax, 5 or ds: 3
B8 06 00                mov ax, (ds: 5) xor 3
B8 06 00                mov ax, ds: 5 xor 3
B8 06 00                mov ax, 5 xor ds: 3
B8 01 00                mov ax, (ds: 5) and 3
B8 01 00                mov ax, ds: 5 and 3
B8 01 00                mov ax, 5 and ds: 3
```

KPR #: D200102392 **CONTINUED**

```
B8 FA FF                    mov ax, not ds: 5
B8 FF FF                    mov ax, ds: 5 ne 3
B8 FF FF                    mov ax, (ds: 5) ne 3
B8 FF FF                    mov ax, 5 ne ds: 3
B8 FF FF                    mov ax, 5 ne (ds: 3)
```

KPR #: D200102400 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
LOW, OFFSET, etc.shouldn't allow complex mem operand

Problem:
As86 allows the user to take HIGH or LOW or offset of a register expression when it should generate an error.
Examples of this are as follows:

```
B8 03 00                    mov ax, high mem2[bx]
B8 00 03          R         mov ax, offset (mem1[bx])
```

KPR #: D200102418 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Can't negate a relocatable item or memory location.

Problem:
As86 allows the user to take the negative of a memory location or of a relocatable item while it should give an error indicating that memory locations cannot be negated.
Some examples of this are as follows:

```
151   0363   2E A1 00 03                mov ax, word ptr (- mem1)
213   03A6   B8 01 03                  mov ax, offset (- mem2)
299   0442   B8 00 00          R       mov ax, seg (- mem1)
333   047B   B8 FF FF                  mov ax, type (- mem1)
373   0484   B8 89 04                  mov ax, 5 - (this word)
717   04CF   B8 05 03                  mov ax, 5 - (mem1.sf1)
```

KPR #: D200102426 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
HIGH and LOW allow bad operands

Problem:
In as86, LOW and HIGH are allowed too many different types of operands.
For example, as86 allows the high or low byte of an address to be taken when the address should not be known until run time.
Errors should have been issued for such instructions.
Also, LOW and HIGH shdn't be able to take segments as arguments since they should only operate on numbers or offsets.
So, a BASE-relocatable value should not be allowed.
Some examples of this are as follows:

```
B3 00                       mov bl, low word ptr [bx]
B3 00                       mov bl, low byte ptr [bx]
B3 00                       mov bl, low byte ptr su1[bx]
```


KPR #: D200102426 **CONTINUED**

```
B3 00                       mov bl, low cs: byte ptr su1[bx]
B3 00                       mov bl, low byte ptr su1[bx][si]
B3 00                       mov bl, low byte ptr su1[bx+si]
B3 05                       mov bl, low byte ptr su1[bx+5]
B3 05                       mov bl, low byte ptr su1[bx+si+5]
B3 05                       mov bl, low byte ptr su1[bx][si][5]
B3 00             R         mov bl, low ??SEG
```

KPR #: D200102434 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Not Detecting variable not in group

Problem:
As86 allows the user to reference a variable using a groupname that the variable doesn't belong to which is misleading to the user. For example,

```
name grouptest
datagrp group data2
data1 segment
d2 dw 1234h
data1 ends
data2 segment
d3 dw 5678h
data2 ends
code segment
assume cs:code,ds:datagrp
;The following instructions should cause errors
;because d2 isn't in the datagrp
mov bx, offset datagrp:d2
mov bx, datagrp:d2
code ends
end
```

KPR #: D200102442 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
LOW of a record field returns wrong value.

Problem:
As86 may return the wrong value when using LOW on a record field.
This is true whether the field name is used or whether an EQU is made to the field name and then the equ symbol is used.
In this example, both are coming out as '0A' when they should be 5.

```
5     0000                                r1 record f1:3, f2:5
33    0037
34    0037   B0 0A                        mov al, low f1
54    0053                                e7 equ f1
71    0063   B0 0A                        mov al, low e7
e7          EQU                             -54 71
f1          REC_FLD   0005 WIDTH=0003       -5 34 35 44 45 54 56
```

KPR #: D200102459 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Expressions using constants > 32k shouldn't work.

Problem:
Expressions using the large EQU symbol should not work.
Instead, overflow errors should be generated since the
value can't be represented by a 17 bit constant.
Only the shift operators appear to demonstrate this problem.
The multiply, divide, and additive operators give the correct errors.
Some example which should have given errors are as follows:

```
5     0000                                L1      EQU     50000000H
6     0000    00                                  DB      L1 SHR 4
7     0001    00 00                               DW      L1 SHR 4
8     0003    00 00 00 00 00 00                   DQ      L1 SHR 4
8             00 00
9     000B    00 00 00 00                         DD      L1 SHR 4
10    000F    00 00 00 00 00 00                   DT      L1 SHR 4
10            00 00 00 00
```

KPR #: D200102467 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
%IN and %OUT should generate unknown macro errors.

Problem:
The %IN and %OUT macros in AP86 do not make sense since they both
put output on STDOUT. As a result, any preprocessed file will not be
useful since AS86 will error on the output of the %IN and %OUT
macros. So, the %IN and %OUT macros should have been removed from AP86
and should have generated "unknown macro name" (301) errors if used.
An example using %IN and %OUT follows:

```
name inerror
a segment
; none of these macros should work (meaning %4%OUT and %3%IN)
%OUT(hello)
%SET(INVALUE,%IN)
db %INVALUE
a ends
end
```

KPR #: D200102475 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
8224 errors & warnings if pagewidth=(i) and (i)>231.

Problem:
The as86 error summary may indicate very many (usually 8224)
errors and warnings when nothing is wrong.
This only appears to occur if
((there is a -f option on the command line) OR
(a $symbols or $xref is included in the .s file))
AND
(the -L option is used)

KPR #: D200102475 **CONTINUED**

AND
($pagewidth(n) is used where n > 231)
I believe that some internal location is being overwritten by a long
pagewidth and that that internal location is then being interpreted as
the integer value for the number of errors and warnings.
There is never a problem if $pagewidth is less than 231.

KPR #: D200102483 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Ap86 changing case of text in macro arguments.

Problem:
The macro preprocessor, ap86, changes some lowercase text to uppercase
when the text is passed as an argument to a macro. This is wrong.
This could potentially cause problems where a
a compiler symbol is changed from one case to another.
An example of this is as follows: -first the ap86 input...

```
%*define(maccall(arg1,arg2,arg3))
(
        ; %arg1
        ; %arg2
        ; %arg3
)
%*maccall(1%9%%%%%%%%lots of metachars, 2%1% one metachar,
3%2)) more stuff)
```

The output from the macro preprocessor, ap86, is as follows:

```
        ; 1%%%%%%%%%LOTS of metachars
        ;  2% one metachar
        ;  3)) more stuff
```

KPR #: D200102491 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Ap86 using unsigned values when should be signed.

Problem:
The numeric compare functions in the macro preprocessor do not always
return the desired value. In the test case below I tried comparing two
values such that it returned a TRUE value. From the second test, it was
apparent that this value was 0FFFFH, not -1. As a result, an error was
returned when a -1 should have been returned.
Also note that the resulting number is stored as an unsigned 16-bit
value when it should be stored as a signed value, so that these results
can be used in further operations inside the assembler.
The result of these two problems can mean some
code might fail that otherwise shouldn't.
Example source is as follows:

```
; showing how EQ doesn't return -1, but returns 0FFFFH
dw %eval((5 eq 5))
dw %eval((5 eq 5) * 4)
dw %eqs(hello,hello)
dw %eval(%eqs(hello,hello) * 4)
dw %eval(-1 * 4)
```

KPR #: D200102491 **CONTINUED**

```
Output of macro preprocessor,ap86
name atest
; showing how EQ doesn't return -1, but returns 0FFFFH
dw 0FFFFH
; ** ERROR   304 Logical expression error.
dw 00H
dw -1H
dw 0FFFCH
dw 0FFFCH
end
;   MACRO ERRORS =     1
```

KPR #: D200102509 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Using seg sym override on abs expression gives bad OMF.

Problem:
Using a segment symbol as an override for a register expression results in incorrect OMF. The segment belongs to the group that is pointed to by DS, so the resulting FIXUP should have a group as a frame and the segment as the target. Instead, the resulting fixup has bad group for the target. A sample code segment follows:

```
0000                            g group a,b
0000                            c segment public
0000                            assume cs:c, ds:g, ss:g
0000   8B 07                    mov ax, [bx]
0002   8B 87 00 00        R     mov ax, a:[bx]
0006   8B 87 00 00        R     mov ax, g:[bx]
The HP OMF-86 dump for the above code follows:
(0043) SEGDEF: SI(01) - 'c'. Class ''. Public Reloc para align Len=0036
(004D) SEGDEF: SI(02) - 'a'. Class ''. Public Reloc word align Len=0001
(0057) SEGDEF: SI(03) - 'b'. Class ''. Public Reloc word align Len=0001
(0061) SEGDEF: SI(04) - '??SEG'. Class ''. Public Rel para align Len=000
(00E4) GRPDEF: GI(01) - 'g' Segment SI(02) - 'a'.  Segment SI(03) - 'b'.
(0149) FIXUPP: Offset   at 004. Seg. relative. Frame: GI(01). Target:
ERROR: (0149) Invalid group index 01 in fixup.
GI(00) + 0000.
Offset   at 008. Seg. relative. Frame: GI(01). Target: GI(01) + 0000.
```

KPR #: D200102517 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
%EVAL using unsigned values for negative values

Problem:
The macro preprocessor, ap86, tends to output a 16-bit unsigned constant for the result of the EVAL macro. While this is acceptable in many cases it would result in a wrong answer or possibly an error if this value was used in an expression that is processed by the assembler. For example, if the evaluated result was to be used to multiply another constant at assembly time, the result would be totally wrong.
What would be better is for the macro preprocessor to output constants by preserving the sign that the constant had after it was calculated

KPR #: D200102517 **CONTINUED**

inside the macro preprocessor so that negative numbers stay negative and positive numbers stay positive. A sample ap86 input follows:

```
dw %eval(- 2)    ; should be -2H, not 0FFFEH
dw %eval(not 2) ; should be -3H, not 0FFFDH
dw %eval(5 - 7) ; should be -2H, not 0FFFEH
The ap86 output follows:
name atest
dw 0FFFEH        ; should be -2H, not 0FFFEH
dw 0FFFDH        ; should be -3H, not 0FFFDH
dw 0FFFEH        ; should be -2H, not 0FFFEH
end
```

KPR #: D200102525 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Segment overflow should generate an ERROR since OMF is bad.

Problem:
If a file overflows the 64k boundary, as86 is currently generating a warning and restarting the location counter at 0.
This is not compatible, however, with the code that is being generated. The problem is that the code extends beyond the 64k segment limit. If this code were to be loaded in ld86 an ERROR would be generated for exceeding the 64k segment limit. This is true regardless of whether the user tries to change the size of the segment with the SEGSIZE directive in the loader. So, the resulting code is not usable. Also, if the user tried to use the code in the loader, the results would not be predictable, due to overlapping of code from different segments. So, the WARNING message should be changed to an ERROR message to indicate that this code is not usable.

KPR #: D200102533 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
When called casesensative, won't recognize 06h

Problem:
Ap86, the macro preprocessor, does not recognize either the characters 'a-f' when used in a hex number or the trailing character for a number (i.e., h, b, o, q, or d) if the preprocessor is invoked with case-sensitivity turned on. Since the numbers are not converted to upper case, the preprocessor is not able to recognize the characters. This should work, however, and should act no differently than in case-insensitive mode. An example follows:

```
db %EVAL(0110B) ; shd be 6H
db %EVAL(0110b) ; shd be 6H
```

KPR #: D200102541 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
PUBDEF/DEBSYM recs wrong when groups involved.

Problem:
As86 is not currently generating PUBDEF and DEBSYM entries correctly

KPR #: D200102541 **CONTINUED**

when groups are involved. If a group is defined that contains some segments and those segments have symbols, as86 is generating PUBDEF and DEBSYM records that have both segment and group indexes, even though the user may not have specified that the symbols within the segment were to be accessed from the group frame. This impacts code generation since all fixups matching these symbols will always be group relative, regardless of how the user meant to access the symbols. Instead, different types of PUBDEF and DEBSYM records should be generated depending upon the type of symbol and how the CS assume register is set.
The following behavior is how the behavior should be defined:
near and far labels:
  If a CS assume has been set to a group at the time of the label's definition, the label will be placed in a PUBDEF and DEBSYM that has a segment and group index. If the CS assume is set to the current segment, then only a segment index will be used. The reasoning behind the above definitions is that fixups to externals will be modified depending upon the PUBDEF definition. If the PUBDEF has a group index, then the fixup will always be group relative, whether the user intended it or not. Also, the symbol should be stored depending upon how the user intended to use the label, which is shown by the CS assume value.
variables:
  Regardless of the CS or any other assume value, the PUBDEF or DEBSYM that contains this symbol will always have only a segment index. The reasoning behind this is that the symbol's address will depend upon how the user accesses the symbol, not how the symbol is defined. This is due to whether or not a segment register is set to a group that the symbol belongs to.
The above paragraphs translate to having only a segment index for a PUBDEF or DEBSYM record as long as CS is assumed to the current segment or the symbols in the PUBDEF or DEBSYM record are variables. If CS is assumed to be a group and a label is defined, then the PUBDEF or DEBSYM record should have both a group and segment index. Note that this indicates that such a segment would end up with labels in a different PUBDEF or DEBSYM than any variables defined within that segment. This would be since the frame of reference is different depending upon the type of symbol.

KPR #: D200102558 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
NAME stuff anything gives CONTINUATION line error.

Problem:
If a NAME directive is used to set a module name and there is anything (except a comment) following the name field, as86 gives an error message about a continuation line found where an initial line was expected. This particular error message has nothing to do with the problem and so it should be replaced with a more appropriate message. The caret correctly points to the error, it is just that the ERROR:(551) is an inappropriate message for the error.

KPR #: D200102566 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Missing THEN causes FORTRAN I?O error 922

Problem:
If an ap86 source file has a missing THEN for a macro preprocessor , ap86 correctly discovers this and points it out with the following message(s):
; ** ERROR 306 Missing "THEN".
; ** ERROR 319 Unbalanced right parenthesis.
But ap86 continues to output dozens of blank lines and finally halts with an internal error message of the following:
*** FORTRAN I/O Error 922: I/O on illegal record attempted
The processing of the missing THEN is fine, but the unplanned halt on Fortran I/O Error indicates that ap86 failed to recover from the error properly.

KPR #: D200102574 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
%NASTY(%1%sinple) not expanded correctly.

Problem:
The macro using %1%simple defined below is expanded incorrectly:

```
        name stuff
someseg segment
        assume cs:someseg
%*DEFINE(simple) LOCAL send_a (
        OR      AX,1
%send_a LABEL   NEAR
)
%*DEFINE(inter_call(routine)) LOCAL LD0_ (
%LD0_:  %routine
)
%*DEFINE(NASTY(routine)) LOCAL LAB1 (
        %inter_call(%routine)
%LAB1:  %routine
)
        %NASTY(%1%simple)
someseg ends
        end
```

This should expand as follows:

```
                                1             name stuff
----                            2     someseg segment
                                3             assume cs:someseg
                                9 +1
                               10 +2
0000                           11 +3  LD0_02:
0000 0D0100                    12 +3          OR      AX,1
0003                           13 +3  SEND_A01        LABEL   NEAR
                               14 +3
                               15 +2
0003                           16 +3  LAB100:
0003 0D0100                    17 +3          OR      AX,1
0006                           18 +4  SEND_A03        LABEL   NEAR
                               19 +3
```

KPR #: D200102574 **CONTINUED**

```
                          20 +1
----                      21      someseg ends
                          22              end
```

However, ap86 expands this to create the following listing:

```
        name stuff
someseg segment
        assume cs:someseg
02:     LAB100
LAB100: LAB100
someseg ends
        end
;   MACRO ERRORS =      0
```

KPR #: D200102582 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
JZ lab1 lab1 LABEL FAR (fwd) accepted, should error.

Problem:
A JZ label1 where label1 has been defined as LABEL FAR (and happens to be a forward reference) is accepted by as86 but should not be.
We might argue that as86 merely 'assumed' a NEAR before the label on the label1 in the JZ, but the code that is generated does not reflect that correctly (although the listing indicates that this is what as86 did).
The .o file contains an A8 00 00 02 whereas the listing indicates that it generated an A8 80 74 02 which is what it would have correctly been if as86 had assumed a NEAR qualifier on the label in the JZ.
So this is silently generating bad code on an erroneous input while not flagging the error.
Note that somehow the use of the JZ has destroyed the code for the TEST instruction preceding it. An example of this follows:

```
0000        $mod186
0000        name test
0000              sysprog   segment byte public
0000              assume    cs:sysprog
0000              PUBLIC    Continue_power_up
0000              Dil_test  LABEL       FAR
0000    A8 80     TEST      AL,80H
0002    74 02     JZ        Continue_power_up
0004    F6 D0     NOT    AL
0006              Continue_power_up      LABEL   FAR
0006    C3        RET
0007              sysprog                        ENDS
0000              END
NO ASSEMBLY ERRORS
NO ASSEMBLY WARNINGS
```

Note that the code generated for the TEST/JZ indicates that as86 has assumed a NEAR qualifier for the label, Continue_power_up at the JZ. But the code generated reflects different code which is not correct.

KPR #: D200102590 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Still silence @ stdout for errors/no -l opt

Problem:
Under some circumstances, as86 does not give an error summary at stdout when it should and under other circumstances gives an error summary at stdout when it should not. The following is what it should do:
If there are any errors or warnings,
then they will always go to stderr.

if there is a -L option
OR there are any errors or warnings
then the error summary will go to stdout.
(This will follow a listing if there is one at either the terminal or the listing file.)

if there is NOT a -L option
AND there are no errors or warnings
then the error summary will NOT be printed to stdout.

$nolist and $list merely turn listings on and off AFTER a listing is already being created because of the -L option.
A $nolist does NOT cause an entire listing file to not even be created, it merely alters the contents.

The following is how as86 behaves:

-L	$noprint	redir?	w err @stderr	wo err @stderr	w err @stdout	wo err @stdout
Yes	No	No	noisy	silent	noisy	noisy
Yes	Yes	No	noisy	silent	noisy	noisy
Yes	No	Yes	noisy	silent	noisy	noisy
Yes	Yes	Yes	noisy	silent	noisy	noisy
No	No	No	noisy	silent	noisy	noisy [WRONG]
No	Yes	No	noisy	silent	noisy	noisy [WRONG]
No	No	Yes	noisy	silent	silent [WRONG]	silent
No	Yes	Yes	noisy	silent	silent [WRONG]	silent
			ALL OK	ALL OK		

KPR #: D200102608 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Don't allow group overrides on exprs in ORG statements.

Problem:
As86 allows group or segment overrides on ORG expressions when the segment is other than the current segment.
Org expressions can be relocatable when the expression involves symbols within the current segment. It does not make sense, however, to allow group overrides or segment overrides when the segment is other than the current segment. In these cases, the expression is no longer relocatable to the current segment and as such cannot be evaluated within the assembler. So, errors should be generated if such an expression is seen. An example follows:

```
3     0000                              g group a,b
```

KPR #: D200102608 **CONTINUED**

```
5     0000                              a segment at 900h
6     0000                              assume cs:a
7     0000   90                         nop
8     0001   90                         nop
9     0002                              a ends
11    0000                              b segment public
12    0000                              assume cs:b
14    0000   90                         nop
15    0001
16    0065                              org offset g:$+100
17    0065   90                         nop
18    0066
19    00CA                              org offset(g:$+100)
```

KPR #: D200102616 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
Core dumps on db "853 char long string

Problem:
As86 core dumps on a file which has a line which contains
db "a very long string" in it where 'a very long string'
is 853 characters long or longer.
If the string is shortened to 852 characters or shorter, as86 doe
not core dump
As86 should be able to accept a line of 1024 character, not 858-1 .

KPR #: D200102624 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
EQU containing segment override fails.

Problem:
If one has has an EQU which contains an operand with a segment override
such as LAB EQU ES:symbol
When one references LAB, the assembler fails to generate the segment
override code. This happens regardless of whether "symbol" is external
or local and regardless of the type of "symbol". For example,

```
        MOV     AX,LAB  ; Fails to generate ES: override
0000                              $mod086
0000                              PROG SEGMENT BYTE PUBLIC     ;+line+
0000                                   ASSUME  CS:PROG ;+line+
0000                                   NAME    equ     ;+line+
0000                                   EXTRN   X1:WORD
0000                                   EXTRN   X2:BYTE
0000   00 00                          L1   DW      0
0002                              EQX1 EQU     ES:X1
0002                              EQL1 EQU     ES:L1
0002                              EQX2 EQU     ES:X2
0002   26 A1 00 00        E              MOV     AX,ES:X1
0006   2E A1 00 00        E              MOV     AX,EQX1 ;ERROR
000A   26 A1 00 00        R              MOV     AX,ES:L1
000E   2E A1 00 00        R              MOV     AX,EQL1 ;ERROR
0012   26 A0 00 00        E              MOV     AL,ES:X2
0016   2E A0 00 00        E              MOV     AL,EQX2 ;ERROR
```


KPR #: D200102624 **CONTINUED**

```
001A                                   PROG ENDS            ;+line+
0000                                        END
```

Since an explicit override works to generate the segment override,
use a macro instead of an EQU to work around this problem.

KPR #: D200102632 Product: 8086 ASSEMB 300 64871S004 01.00

One-line description:
sigint and sigquit wrong

Problem:
Sigint and sigquit are not always handled correctly by as86,ap86 & ld86.
When ld86 is run in background with an ampersand (&), it should NOT
be able to be killed by a foreground signal because the foreground
signal probably was intended for a foreground process and not for the
background process.
If the user should happen to begin to list a file with 'more' and then
wishes to terminate the listing with a sigint (cntl C) or with a
sigquit (cntl |), the signal sent in foreground to halt the
foreground 'more' should NOT halt the background load job.
But ld86 running in background is killed by foreground signals.
The user can avoid this problem by trying to prevent any foreground
use of signals or by always running loads in a separate shell.

Ap86 demonstrates the exact same problem and can be killed by a
foreground signal while it is running in background.

As86 correctly ignores foreground signals while running in background,
but it unfortunately also ignores a sigquit (cntl |) while it is
running in foreground as well.
So as86 must be interrupted in foreground with a sigint (cntl C)
or with a kill -9.

KPR #: D200103176 Product: 8086 DQ EMUL 300 64220S004 01.30

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103523 Product: 8086 DQ EMUL 300 64220S004 01.30

One-line description:
Running w/o monitor will corrupt the offset value on INT 0 vector.

Problem:
Running the emulator from reset without a monitor loaded will cause the value 0080h to be written to address 0h. This location is the offset value of the vector for interrupt 0 (divide by zero).

The problem is that when the state of the emulator is changed from RESET to RUNNING the routine check_state() is called. This calls i80xx_monitor_status() which does an ARE_YOU_THERE. If the monitor control word is not defined, this i80xx_monitor_command() should return FAILED before doing the ARE_YOU_THERE (since the monitor control word is undefined, 0080h is being written to address 0h).

Temporary solution:
There are two work arounds. The first is to make sure a monitor is loaded. This is the easiest. If a monitor cannot be loaded, then the global symbol "MONITOR_CONTROL" must exsist and be a word location in emulation memory that can be written to without messing anyting up.

KPR #: D200098244 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103192 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: 5000440743 Product: 8086/8 C 64818 03.80

Keywords: PROBLEM ON 9000/S300

One-line description:
Incorrect code being generated for array assignment

KPR #: D200099416 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S500

One-line description:
Call to function using LONGS uses wrong segment.

Problem:
The compiler does not use correct segment. The DS segment is used instead of SS segment. The examples is as follows.

```
"C"
"80188"
main()
{
 long c;
 long sub();
 sub(c);
   /*PUSH    SS:[BP-00004H]
     PUSH    SS:[BP-00006H]
     LEA     BA,SS:[BP-0000AH]     <---- SS is used here.
     PUSH    BX
     CALL    NEAR PTR sub
     ADD     SP,#+00006H  */
}
 long  sub(c);
 long  c;
 {
   return (c);
     /*PUSH    SS:[BP+00008H]
       PUSH    SS:[BP+00006H]
       POP     SS:[BP-00006H]
       POP     SS:[BP-00004H]
       JMP     NEAR PTR sub03_0   */
 }
/* sub03_0
       PUSH    SS:[BP-00004H]
       PUSH    SS:[BP-00006H]
       MOV     BX,SS:WORD PTR [BP+00004H]
       POP     DS:[BX]                    <-----| DS is used here.
       POP     DS:[BX+00002H]             <-----| SS should be used.
 */
```

KPR #: D200099440 Product: 8086/8 C 64818 03.80

One-line description:
Pointer deref. followed by type conv., generates bad code.

KPR #: 5000422782 Product: 8086/8 PASCAL 64814 03.50

One-line description:
Our products on the VAX do not implement directory default protection

Problem:
When a VMS directory contains an ACE which specifies a default protection mask, all files created within that directory should take on that default. Instead, when our products write files they take on the UIC's default.

The problem seems to be caused by the C libraries supplied by DEC. These are used by all our tools (being written in C)

KPR #: 5000430454 Product: 8086/8 PASCAL 64814 03.60

Keywords: PROBLEM ON VAX

One-line description:
Error messages are not specific.

Problem:
Attempting to compile a program in a directory in which the user does not have write privelege results in an unexpected error. For example:

```
comp [dir]file.p  --- results in error "comp : failed; too many
                        errors in pass 3"
```

If the user sets the default directory to be the directory in question and compiles the file locally a different error is generated.

```
set def [dir]
comp file.p  ---> results in error "ioerror : comp cann not open
                  GLOBALS file"
```

Neither of these error seem to indicate the true problem. Please modify the messages to better describe the problem.

KPR #: 5000430611 Product: 8086/8 PASCAL 64814 03.60

Keywords: PROBLEM ON VAX

One-line description:
DIV uses incorrect segment.

Problem:
The following DIV statements produce bad code:

```
"80186"
$EXTENSIONS ON$
PROGRAM DIVTEST;
VAR
  Day_Of_Year : SIGNED_16;
  Date1,Date2,MonthNo : SIGNED_8;
BEGIN
```

KPR #: 5000430611 **CONTINUED**

```
  Date1 := MonthNo DIV 10;
  Date2 := Day_Of_Year DIV 10;
      MOV AX,DS:WORD PTR DDIVTEST
      CWD                          << Sign extend Day_Of_Year thru DX
      MOV DX,#+0000AH              << Overload DX
      MOV BX,DX
      IDIV BX                            << Divide with corrupt DX
      MOV DS:BYTE PTR DDIVTEST+00003H,AL
END.
```

Bob Poulsen - NARC/atl

I verified the bug with rev 3.60 on the S300. The error only occurs when the SIGNED_8 division precedes the SIGNED_16 division and the divisor in each division statement is the same. If the divisors are different, the value is loaded immediately into BX instead of instead of first being loaded into DX.

Temporary solution:
Change the SIGNED_8 variables to SIGNED_16.

KPR #: 5000431965 Product: 8086/8 PASCAL 64814 03.60

Keywords: PROBLEM ON VAX

One-line description:
ES register destroyed when accessing external 32 bit pointer.

Problem:
The compiler destroys the contents of the ES register when accessing an external variable with 32-bit pointers. The following code illustrates:

```
"80186"
$EXTENSIONS ON,POINTER_SIZE 32,FAR_EXTVARS$
PROGRAM IFTEST;
TYPE
  SYS_STRING = RECORD
                 LENGTH : UNSIGNED_8;
                 CHARS : ARRAY [1..40] OF CHAR;
               END;
  TABLEENTRY_T = RECORD
                   TABLEENTRIES : UNSIGNED_8;
                   TABLEKEYS : UNSIGNED_8;
                 END;

TABLE_T = ARRAY [0..1000] OF TABLEENTRY_T;
STTABLEPRT_T = ^TABLE_T;

VAR
$EXTVAR ON$
  CHARBUFFER : SYS_STRING_T;
  STARTOFTBLEARR : STTABLEPTR_T;
  LOCTABLEPTR : INTEGER;
```


KPR #: 5000431965 **CONTINUED**

```
BEGIN
  IF CHARBUFFER.LENGTH = STARTOFTBLEARR^[LOCTABLEPTR].TABLEKEYS
     MOV  AX,SEG LOCTABLEPTR
     MOV  ES,AX
     MOV  BX,IS:WORD PTR LOCTABLEPTR
     SHL  BX
     MOV  AX,SEG STARTOFTBLEARR
     MOV  ES,AX
     LES  SI,ES:DWORD PTR STARTOFTBLEARR  << ES points to STARTOFTBLEARR
     ADD  BX,SI
     MOV  AX,SEG CHARBUFFER
     MOV  ES,AX                           << ES points to CHARBUFFER
     MOV  AL,ES:BYTE PTR CHARBUFFER
     CMP  ES:BYTE PTR [BX+00001H], AL     << compare using wrong segment
     JE   SHORT $+00005H
     JMP  NEAR PTR IFTEST00_0
  THEN
       LOCTABLEPTR := 0;

  IF STARTOFTBLEARR^[LOCTABLEPTR].TABLEKEYS = CHARBUFFER.LENGTH
  THEN
       LOCTABLEPTR := 1;

END.
```

The second comparison generates good code, i.e. ES is not overwritten.

Temporary solution:
Use a temporary assignment to a local variable. Or, don't use 32-bit pointers or far external variables.

KPR #: D200099028 Product: 8086/8 PASCAL 64814 03.60

Keywords: PROBLEM ON 9000/S300

One-line description:
Unable to assign a one dim array to a two dim array

Signed off 02/03/89 in release A03.90

KPR #: D200099432 Product: 8086/8 PASCAL 64814 03.60

One-line description:
Vector ref. followed by array ref. generates bad code.

Problem:
Detailed Listing for Defect Number LSDqf05399
Text:
  Vector ref. followed by array ref. generates bad code
.....
This is a hot-site defect that was caused as a side effect of the fix for LSDqf03604. The problem is in subroutine DOM_R in file opmutil.P.

problem

KPR #: D200099432 **CONTINUED**

```
  1 0000  1  "80188"PREPROCESS
  2 0000  1  $EXTENSIONS ON$
  3 0000  1
  4 0000  1  PROGRAM MEM;
  5 0000  1  $RECURSIVE OFF$
  6 0000  1
  7 0000  1  TYPE
  8 0000  1
  9 0000  1     INTEGER = SIGNED_16;
 10 0000  1     ADD_REC = ARRAY[23..29] OF REAL;
 11 0000  1
 12 0000  1  VAR
 13 0000  1     $EXTVAR ON$
 14 0000  1     PROG_BUF_ADD : ARRAY[0..25] OF ADD_REC;
 15 0000  1     LINE_IM      : ARRAY[1..43] OF REAL;
 16 0000  1
 17 0000  1     $FAR_EXTVARS$
 18 0000  1     LINE_PNT     : ARRAY[1..29] OF BYTE;
 19 0000  1     $EXTVAR OFF$
 20 0000  1
 21 0000  1     I,J          : INTEGER;
 22 0000  1
 23 0000  1
 24 0000  1  PROCEDURE SAVE(BENDING:BYTE);
 25 0000  2
 26 0000  2  VAR J : BYTE;
 27 0004  2
 28 0004  2  BEGIN
 29 0004  2           J := LINE_PNT[I];
       0004              MOV    BX,DS:WORD PTR DMEM
       0008              MOV    AX,SEG LINE_PNT
       000B              MOV    ES,AX
       000D              MOV    AL,ES:BYTE PTR LINE_PNT[BX-00001H]
       0012              MOV    DS:BYTE PTR DSAVE,AL
 31 0015  2           PROG_BUF_ADD[BENDING,J] := LINE_IM[I];
       0015              MOV    AL,#+0001CH
       0017              IMUL   SS:BYTE PTR [BP+00004H]
       001A              MOV    SI,AX
       001C              LEA    SI,DS:PROG_BUF_ADD[SI]
       0020              SUB    SI,#+0005CH
       0023              SHL    BX,#+00002H
       0026              MOV    DI,BX
       0028              MOV    AX,SI
       002A              LEA    SI,DS:LINE_IM[DI-00004H]
       002E              MOV    BX,AX
       0030              MOV    CX,AX
       0032              MOV    AL,DS:BYTE PTR DSAVE
       0035              CBW
       0036              SHL    AX,#+00002H
       0039              ADD    BX,CX      <--- should be ADD  BX,AX
       003B              LEA    DI,DS:[BX]
       003D              MOV    CX,#+00004H
       0040              PUSH   DS
       0041              POP    ES
```


KPR #: D200099432 **CONTINUED**

```
       0042              CLD
       0043              REP    MOVSB

 33 0045  2
 34 0045  2  END;
 35 0058  1
 36 0058  1  BEGIN
 37 0058  1  END.
```

KPR #: D200102822 Product: 8086/8 PASCAL 64814 03.60

Keywords: PROBLEM ON VAX

One-line description:
Procedure call after vector ref causes 1006 error on VAX

KPR #: D200102830 Product: 8086/8 PASCAL 64814 03.60

Keywords: PROBLEM ON VAX

One-line description:
Variable function call inside IF statement generates bad code.

Problem:
The following program produced bad code:

```
"C"
"8086"
$POINTER_SIZE 32$
$FAR_LIBRARIES$
$FAR_PROC ON$
$FIXED_PARAMETERS OFF$
$ENTRY OFF$
$FAR_EXTVARS ON$
$SEPARATE_CONST OFF$

int f1(p1,p2)
int *p1;
char **p2;
{
  int j;
  j = *p1;
  return(j);
}

struct str1{
              int mem1;
              int mem2;
              int (*mem3[2])();
           } *work;

main()
{
  int i;
  char **a;
```

KPR #: D200102830 **CONTINUED**

```
  a=0x10;
  for (i=0; i<2; i++) {
                        work->mem3[i]=f1;
                        if ((work->mem3[i])(&i,&a[i]) != 0x0)
                            ; /*  produced bad code  */
                            ;
                       }
   ;
}
```


KPR #: 5000283937 Product: 8086/88 C M 64818-90905 00.00

One-line description:
Enhance to include a disc. on the symbol limitations of the compiler.

Problem:
Request to enhance the documentation to include a discussion on the symbol limitations of the compiler. Specifically, to change the manual to discuss the limitations on external variables and constants with the compiler. The only documentation on these limitations is in the form of error messages listed in the appendix.

Temporary solution:
There is no workaround available.

KPR #: D200103184 Product: 8088 DQ EMUL 300 64221S004 01.20

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103531 Product: 8088 DQ EMUL 300 64221S004 01.20

One-line description:
Running w/o monitor will corrupt the offset value on INT 0 vector.

Problem:
Running the emulator from reset without a monitor loaded will cause the value 0080h to be written to address 0h. This location is the offset value of the vector for interrupt 0 (divide by zero).

The problem is that when the state of the emulator is changed from RESET to RUNNING the routine check_state() is called. This calls i80xx_monitor_status() which does an ARE_YOU_THERE. If the monitor control word is not defined, this i80xx_monitor_command() should return FAILED before doing the ARE_YOU_THERE (since the monitor control word is undefined, 0080h is being written to address 0h).

Temporary solution:
There are two work arounds. The first is to make sure a monitor is loaded. This is the easiest. If a monitor cannot be loaded, then the global symbol "MONITOR_CONTROL" must exsist and be a word location in emulation memory that can be written to without messing anyting up.

KPR #: D200098277 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103226 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200091249 Product: EMUL TERMINAL IF 64740-90901 01.00

One-line description:
"step" doesn't work when CMP is active; (need to change the manual)

Problem:
When the Coordinated Measurement Bus (CMB) is being actively controlled by another emulator, the step command ("s") does not work correctly. The emulator may end up running in user code (NOT stepping).

Temporary solution:
Disable CMB interaction ("cmb -d") while stepping the processor.

KPR #: D200096834 Product: EMUL TERMINAL IF CMB 64306-90901 01.00

One-line description:
"step" doesn't work when CMP is active; (need to change the manual)

Problem:
When the Coordinated Measurement Bus (CMB) is being actively controlled by another emulator, the step command ("s") does not work correctly. The emulator may end up running in user code (NOT stepping).

Temporary solution:
Disable CMB interaction ("cmb -d") while stepping the processor.

KPR #: D200098392 Product: F9450 EMUL 300 64286S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200099424 Product: GENERIC ANALYSIS M 64740-90909 01.00

One-line description:
Measurements between the external/internal analyzers aren't synchronized

Problem:
The 68000 emulator does not "synchronize" measurements between the external and internal analyzers when the "xtmo -e" command is issued. The Terminal Interface User's Reference explains that the xtmo -e will ".. synchronize measurements made by the two analyzers."
An example that shows that this is not the case can be shown by plugging into the CSA DEMO BOX and mapping all resources to target. Use "xtmo -e" to synchronize the external analyzer with the internal analyzer. Monitor the R/W- line with one of the external bits, and it will always show as a one, even on write cycles.

Signed off 01/26/89 in release X00.00

KPR #: D200098038 Product: GENERIC EMULATION FW 64700 01.00

One-line description:
Combining CMB trigger driving and receiving can hang the system

---

KPR #: D200098632 Product: GENERIC EMULATION FW 64700 01.00

One-line description:
Base 10 output of a 32-bit analyzer field contains the character ":".

KPR #: 5000417014 Product: GENERIC EMULATION FW 64700-90901 01.00

One-line description:
Manual needs to clarified concerning fg mon and user prog loading

---

KPR #: 5000423681 Product: HOST SOFTWARE 300 64883 01.20

One-line description:
4K .A file high speed link failure

KPR #: D200098020 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
HPIB SELECT PORT PROBLEM

Problem:
Use an FSD function to set the Human Interface HPIB port to select code 7 for all HP9000/3x0 except 310.

Signed off 02/03/89 in release A01.30

KPR #: D200098657 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
Output warning msg. if number of ndilbuffers is <13

Problem:
A warning message will be output by csib if the hp-ux kernel is configured with out adequate number of "ndilbuffers". This should help the user who has build a hp-ux kernel on a diskless cnode, using dfile.cnode or dfile.cnodemin. The warning message suggests that the kernel needs at least 13 ndilbuffers.

Signed off 02/03/89 in release A01.30

KPR #: D200102772 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
A warning in a filelist transfer will abort the transfer.

Problem:
A filelist transfer which generates a warning (eg. file not found) will not attempt to transfer any of the subsequent files in the list.

KPR #: D200102798 Product: HOST SOFTWARE VAX 64882 02.40

One-line description:
A warning in a filelist transfer will abort the transfer.

Problem:
A filelist transfer which generates a warning (eg. file not found) will not attempt to transfer any of the subsequent files in the list.

Signed off 02/03/89 in release A02.50

KPR #: D200102806 Product: HOST SOFTWARE VAX 64882 02.40

One-line description:
Recompile and relink on VMS 5.0 with 5.0 runtime libraries

Problem:
It is necessary to recompile and relink both the hosted development system and transfer under VMS 5.0 using the 5.0 runtime libraries.

Signed off 02/03/89 in release A02.50

KPR #: 1650069583 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
Checker does not work properly for SEM cell with multiple actions

Problem:
If a State Event Matrix cell is defined as

    action1 ; action2 / next_state

and action1 and action2 are shown as two distinct Control Flows on the parent Data Flow Diagram then the syntax checker finds
    action1 out of C-spec unmatched
    action2 out of C-spec unmatched
    action1.action2 out of SEM unmatched

Temporary solution:
No know temporary solution.

KPR #: 1650074849 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
twk2hpgl can't convert the postscript output generated by twk_image

Problem:
twk2hpgl can't convert the postscript output generated by twk_image.

twk2hpgl gives error message "extra arguments on stack" and the produced hpgl file is incomplete and incorrect.

Temporary solution:
You must print the object from within Teamwork and save the output in a file.

KPR #: 5000423236 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
Formatted P-Spec prints FOOTER_TEXT on second page.

Problem:
When I print a formatted P-SPEC on the laserjetII the footer text is placed on the top of the second page instead of on the bottom of the first page.

Temporary solution:
Bob Poulsen - NARC/Atl

Under HP-UX 6.2 the model script for the laserjet printer was modified to change the default page length from 66 lines to 60 lines. When the default page size is used in /usr/hp64000/hptwk/spooler/spooler.config (8.0 x 11.0) the footer text is placed on line 63 of a formatted P-SPEC. This causes the text to be placed on the next page.

There are two workarounds. First, the model script can be modified to change the page length from 60 lines to 66 lines (see line 488 of rev 56.16 model script). Second, the page size can be modified in the

KPR #: 5000423236 **CONTINUED**

spooler.config file.

Please change the formatted output to print on 60 lines or document the necessary changes in the manual.

KPR #: 5000435248 Product: HP TEAMWORK 300 64711S004 03.00

One-line description:
Text is left on display after leaving a spawned shell

Problem:
Text is left on display after returning to TEAMWORK from a forked shell process.
If the text and background colors on the console are not left the default white text on black background, then the TEAMWORK display will not be properly redrawn when the shell is exited. For example, if the console is using cyan text and black background, and the shell (created by typing CNTL-I) is exited, then the text from the shell is left on the display and appears green. This problem does not occur when using xtwk.

Temporary solution:
Use only white text on black background for a shell spawned from Teamwork.

KPR #: 5000439620 Product: HP TEAMWORK 300 64711S004 03.00

One-line description:
Index editor dies when doing a copy subtree

Problem:
The index editor (/usr/hp64000/hptwk/twk/bin/xed) is killed when the "copy subtree" command is used from the process index. Teamwork must be stopped and restarted to resume operation.

Temporary solution:
No temporary solution.

KPR #: D200096859 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
Cannot run dc_server on machine which has file system nfs mounted

Problem:
The dc_server will not run on a system which has nfs mounted the teamwork software from another system.

Temporary solution:
The dc_server must be executed on the system which physically has the file system mounted.

KPR #: D200102194 Product: HP TEAMWORK 300 64711S004 03.00

One-line description:
Capital file names not accepted by where referenced report.

Problem:
After selecting where_referenced from the Whole_DD menu the user is prompted for a Report file name. The default file name is /usr/hp64000/hptwk/reports/cruisaaaXXXX. Changing this default works fine except when the file name is all capitals. The problem: the data is indeed dumped to the capital file name. However, the verify window attempts to open the respective file in lower case and displays a "file does not exist error".

Temporary solution:
Use lower case letters.

KPR #: D200102913 Product: HP TEAMWORK 300 64711S004 03.00

One-line description:
SECURITY VIOLATION occurs when ID Module is not first in hil chain

Problem:
SECURITY VIOLATION occurs when Teamwork ID Module is not the first in the hil chain when doing copy subtree.

Temporary solution:
Teamwork id module must be first in hil chain

KPR #: D200077883 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
load_dd -c false_file_name causes an unknown error.

Problem:
Executing a "load_dd" and passing a false file name for the configuration file causes a strange error such as:

```
*******************************************************
EXC_PANIC -- Raise: No context
Exception = ' DC_MAILBOX_NAME_CONFIG_ERROR'
Traceback:
*******************************************************
```

This error message is not documented in the manual.

Temporary solution:
see problem text.

KPR #: 5000235150 Product: HP-UX OP SYS M 64801-90903 01.00

One-line description:
Add documentation for configuring swap space.

Problem:
NOTE: this SR is submitted against the HP-UX series 300 Hosted Software Installation Manual. Product number 64801-90907 was not recognized by RCSTARS, so the SR was submitted against the software 64801.
The manual should be updated to include guidelines for configuring swap space. The HP-UX System Administrator's Manual refers the user to the application manual when swap space requirements need to be determined. The manual should include information on emulation systems, language tools, etc. Discussions should include guidelines for different size files. Edbuild is of particular interest.

Temporary solution:
No workaround at this time.

KPR #: 5000293530 Product: HS ST/TIM ANAL M 64610-90901 01.00

One-line description:
Remove HP2225 and HP2631G printers for support from the manual.

KPR #: 5000441485 Product: P1750 EMUL 300 64288S004 01.10

One-line description:
Incorrect disassembly of traces

Problem:
When using the P1750 emulator, a single instruction loop using the SOJ instruction is disassembled incorrectly.

KPR #: D200098400 Product: P1750 EMUL 300 64288S004 01.10

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

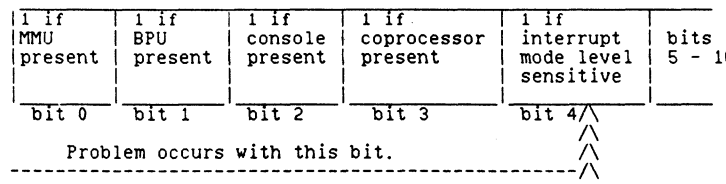
The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103077 Product: P1750 EMUL 300 64288S004 01.10

One-line description:
The emulator does not configure the sys. config. register properly.

Problem:
The 64000-UX version of the F9450 emulator does not configure the system configuration register properly for the interrupt mode question.
If the level sensitive mode is selected, then bit 4 of the System Configuration Register should be one, and a zero if the edge sensitive mode is selected. The emulator configures the register for 0 with level sensitive and 1 with edge sensitive. This can be seen by issuing a display io_port 8410H.

SYSTEM CONFIGURATION REGISTER



But, if the emulator configuration questions are answered to set the system to level-sensitive mode, the content of the system configuration register should be x8xxH (where x means "don't care).

KPR #: D200103077 **CONTINUED**

But instead it contains x0xxH. Bit 4 has reverse presentation of what it should be.

Temporary solution:
There is no workaround available at this time.

KPR #: 1650074567 Product: PROM PROGRAMMER 64501 01.10

One-line description:
Intel D2764A PROMs blowing up

KPR #: 5000430421 Product: PROM PROGRAMMER 300 64501S004 01.50

One-line description:
Customer unable to program NMC27C32 EPROMs

KPR #: D200096214 Product: PROM PROGRAMMER 300 64501S004 01.30

One-line description:
Adden an enhancement to the checksum stuff. checksum file.

Problem:
Detailed Listing for Defect Number LSDqf05034
Text:
Added enhancement to the checksum stuff. checksum file.

.....
I fixed the checksum command in the prom programmer to correctly add in the spots in the prom that are not in the file. This now works exactly like a CROCK.

KPR #: 5000429795 Product: ROM EMULATION 64272 01.04

One-line description:
Some 64502A revisions cause the 64272 to fail its option test

Problem:
Some revisions of the 64502A prom module cause the ROM emulator to fail option test.

KPR #: D200098665 Product: RS-232 TRANSFER 300 64885 01.30

One-line description:
Output warning msg. if number of ndilbuffers is ‹13

Problem:
A warning message will be output by csib if the hp-ux kernel is configured with out adequate number of "ndilbuffers". This should help the user who has build a hp-ux kernel on a diskless cnode, using dfile.cnode or dfile.cnodemin. The warning message suggests that the kernel needs at least 13 ndilbuffers.

Signed off 02/03/89 in release A01.50

KPR #: D200102814 Product: RS-232 TRANSFER VAX 64886 01.60

One-line description:
Recompile and relink on VMS 5.0 with 5.0 runtime libraries

Problem:
It is necessary to recompile and relink both the hosted development system and transfer under VMS 5.0 using the 5.0 runtime libraries.

Signed off 02/03/89 in release A01.70

KPR #: 5000417063 Product: STATE 80386 64659 01.00

One-line description:
SPECIFYING DISASSEMBLER IS CUMBERSOME

KPR #: D200097626 Product: TMS32020 EMUL FW 64786 01.02

One-line description:
A ROVM instruction following a DINT is skipped when single stepping

Problem:
When single stepping, an instruction sequence consisting of a DINT followed by a ROVM is treated as a single step. The ROVM seems to be executed but the single step display skips it. In other words the PC skips from the address of the DINT to the address of the instruction following the ROVM.

KPR #: D200103358 Product: USER DEF EMUL 300 64274S004 01.20

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200097071 Product: USER DEF EMULATION 64274 01.06

One-line description:
UDE displays incorrect data in emulation memory when read fails

Problem:
The UDE memory display command can display data which is not equal to the data actually stored in memory. This has been observed in a system where a large percentage of attempted emulation memory reads failed because the emulation processor was HALTed.

KPR #: D200094854 Product: USER DEFIN ASM M 64851-90904 01.00

One-line description:
Assembler reference manual should explain EQU and MACROs better.

KPR #: D200099390 Product: USER INTERFACE 300 64808S004 02.10

One-line description:
Pmon flags legitimate option for lnk (for 64859) as syntax error

Problem:
the 64859 linker for the 80286B (protected mode) has to be invoked with a special option -b (not -h).
The "pmon" interface revision 2.10 however will flag this option as a syntax error.

Temporary solution:
Invoke linker using shell escape: !lnk -b

KPR #: D200098350 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103309 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103234 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200103242 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200098368 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Monitor commands may not complete execution correctly with target sys.

Problem:
Monitor cmds may not be given enough time to finish when the emulator is used with a target system. The monitor cmds are any command executed by the monitor from the wait loop (ARE_YOU_THERE, COPY, EXIT_MONITOR, etc.)

If the target system halts the emulator or runs extremely slow, the monitor command will not complete before the aries card cage gives up.

The card cage should wait 100 ms for the command to complete. Many of the emulators wait much les than this.

KPR #: D200103317 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Emulation hangs on shell commands that run quickly.

Problem:
On a lightly or moderately loaded system, the emulator will hang after using the "!" feature to run a command that terminates quickly. For instance I have had the emulator hang numerous times after
"!pwd" and "!cd ." commands
It hangs immediately after Unix writes the output of the commmand to the screen. At this point, the emulator should respond with "Press return to continue.", but no response is given, and no keystrokes can continue the session (not even <CTRL> - \ to try to terminate the session.) The only way to terminate the session is to "Kill" the process.

The problem may be a timing issue. This problem only appears when I am using an emulator attached to the host CPU that I am logged in to. I cannot get the problem to appear when using an emulator on a host other than my own. Possibly the delay when using "VT" across the net keeps the problem from surfacing.

KPR #: D200082057 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by
```
cat < ptyxx
```
This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200085878 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                                  ; end locks the emulation session
3. <system name> <module name>          ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086264 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088229 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090688 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095539 Product: 6301V EMULATION 300 64206S004 01.20

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size

KPR #: D200095539 **CONTINUED**

is too small".

KPR #: D200095828 Product: 6301V EMULATION 300 64206S004 01.20

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: 5000420208 Product: 6301V/03R EMUL 64206 01.01

One-line description:
symbol tables get corrupt.

KPR #: D200088088 Product: 6301V/03R EMUL 64206 01.01

One-line description:
6301V/03R module cannot be accessed with HP-UX 6.01

Problem:
In A 131 bundled system the 6301V/03R fails to operate with HP-UX 6.01.
Error message reads: 6301-00c: could not open module - check HPIB
and power to card cage.

Temporary solution:
The current workaround requires that the emulation software be put
in the software debug mode by creating a file:

usr/hp64000/log/adb.X.0

where "X" represents the select code for the HPIB interface.

Then standard error must be redirected to /dev/null when the emulator
is invoked.

KPR #: D200092122 Product: 6301V/03R EMUL 64206 01.01

One-line description:
Illegal opcode error occur when displaying memory repetetively

---

KPR #: D200082065 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed
some of the measurement systems processes are left running. Please
change the behaviour of the products so that these processes die
nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can
release the processes by
cat < ptyxx
This causes the pending output to be flushed, and the processes will die
naturally.

KPR #: D200085886 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the
following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

The symbols will not be displayed even if you try to re-execute step
number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the
problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200086272 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter
emulation, reentry will not be possible, and there will be a continuous
error message stating that the 64120 will not accept a download. This
is a problem with emulation core, and exists in all emulators.

KPR #: D200088237 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090696 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095547 Product: 6301X EMULATION 300 64207S004 01.10

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size

KPR #: D200095547 **CONTINUED**

is too small".

KPR #: D200095836 Product: 6301X EMULATION 300 64207S004 01.10

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200090704 Product: 6301Y EMULATION 300 64208S004 00.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095554 Product: 6301Y EMULATION 300 64208S004 01.10

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095844 Product: 6301Y EMULATION 300 64208S004 01.10

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200081596 Product: 6301Y/03Y EMUL 64208 01.00

One-line description:
Emulator can't work when external clock is selected and E clock = 160khz

Problem:
Emulator can not work when external clock is selected and system clock rate (E clock) is 160 KHz.
Same configuration with 64100A (Pisces I) can perform correct emulation.

KPR #: D200090472 Product: 64000-UX OP-ENV 300 64801S004 01.80

One-line description:
EDB problems with scoping of locals from new com/asm/linker

Problem:
Detailed Listing for Defect Number LSDqf03783
Text:
edb problems with scoping of locals from new comp/asm/linker.

Symbols that are expected to be scoped relative to function are sometimes included with the file local symbols, causing unexpected and varying results in inverse-assembly of tracelists as well as confusion in command entry.

Several of the function entry link symbols are not scoped to the function, while others are. This seems to be random depending on the file being used. Everything "appears" to be ok from the language tools and in the .A files.

Temporary solution:
There is no workaround available.

KPR #: 5000240580 Product: 64HP-UXVMS8086/8 A M 64853-90908 02.03

Keywords: MANUAL

One-line description:
Need Manual change to explain AC and PH phase errors.

Duplicate Service Requests: 5000240960

KPR #: 5000269779 Product: 650X ASSEMB 64843 01.00

Keywords: PROBLEM ON 9000/S300 LINKER

One-line description:
LNK does load NOLOAD files.

Problem:
linker DOES load and link NO LOAD objects.

EXAMPLE
```
    object ?   abc_a.R,(abc_b.R)
    library?
    load address ... ... ....
```

abc_b.R relocatable file is loaded and linked.

Temporary solution:
The linker is operating correctly. The linker listing file is in error in that it shows that the (noload) file has been loaded when it actually has not.

KPR #: 5000219865 Product: 6800 C 64821 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Libraries cause write to ROM

Problem:
The PUSHX:D6800 library will cause a write to ROM if the library is linked in a ROM section. The library declares a local variable SAVEX using the RMB directive in an area defined as PROG. It then writes to that variable. This variable should be in a DATA section.
An example of code that produces this problem:

```
"C"
"6800"
$RECURSIVE-$
int  func1(i)
int  i;
{    return(i);
}
main()
{     int i;
      i = func1(i);
}
```

Temporary solution:
Rex Mayne - Atlanta Response Center

See submitter text

WORKAROUND

Use the slib libraries, or link the dlib libraries in a RAM section.
Rex Mayne - Atlanta Response Center

See submitter text

WORKAROUND

Use the slib libraries, or link the dlib libraries in a RAM section.

KPR #: 5000293779 Product: 6800 C 64821 02.10

One-line description:
Switch statement causes infinite loop.

Problem:
Switch statement causes infinite loop.

Example code:

```
"C"
"6800"
#$FIXED_PARAMETERS ON$
#$INIT_ZEROS OFF$
#$RECURSIVE OFF$
#$SEPERATE ON$
#$UPPER_KEYS ON$
```

KPR #: 5000293779 **CONTINUED**

```
#$SHORT_ARITH ON$
#$WARN OFF$

#define ZLT_ALOCK_LI  0x0001
#define ZLT_POLL_LI   0x0002
#define ZLT_PLU_LI    0x0004
#define ZLT_PARAM_LI  0x0008
#define ZLT_MBT_LI    0x0010
#define ZLT_ACSH_LI   0x0020
#define ZLT_ICSH_LI   0x0040
#define ZLT_SALP_LI   0x0080
#define ZLT_DEPT_LI   0x0100
#define ZLT_SURVEY_LI 0x4000
#define ZLT_CONFIG_LI 0x8000
#define ZDC_ACK_C     6

PROC1 () {

        UNSIGNED *TYP_UP;

        SWITCH (*TYP_UP & ~ZLT_ALOCK_LI)  {

             CASE ZLT_PARAM_LI:

             CASE ZLT_MBT_LI:

             CASE ZLT_ACSH_LI:

             CASE ZLT_ICSH_LI:

             CASE ZLT_SALP_LI:

             CASE ZLT_DEPT_LI:

             CASE ZLT_SURVEY_LI:

             CASE ZLT_CONFIG_LI:
                 *LAN_BTBUF_AC = 0x80;


             DEFAULT:       ;
            }
}
```

Temporary solution:
This only fails for this specific example. So, the only workaround is to simply not use this specific code.

Duplicate Service Requests: D200091348

KPR #: D200068197 Product: 6800 C 64821 01.06

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

"C"
"processor"

```
struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200069823 Product: 6800 C 64821 01.06

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

"C"
"processor"

```
$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);              /* See comment below. */
```

KPR #: D200069823 **CONTINUED**

```
#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200074989 Product: 6800 C 64821 01.07

One-line description:
USE OF MANY FUNCTION CALLS WITH CONSTANT PARAMATERS MAY CAUSE ERR #1007

Problem:
C programs with many function calls with constant parameters may cause Pass 2 Error 1007 - expression too complicated.

The workaround is to use variables to pass parameters. Since the error is related to the compiler attempting to create logical temporaries in order to have these parameters passed as the proper size, the problems can be avoided by using variables instead of constants as parameters.

If only a few instances of constants being passed as parameters is encountered, the use of type casting the constants to int may be sufficient to allow the program to compile.

This is related to the known 6800 compiler limitation which can cause Pass 2 Error 1010 - Too many constants.

When passing constants as parameters, the 6800 code generator will eventually run into the 256 constant limit which produces the 1010 error. The use of variables is the only solution in this instance.

In most instances which produced the 1010 error in previous revisions of the C/6800 compiler, the 1007 error will now be produced. In some instances which previously produced no errors, the 1007 error may be produced.

The following program fragment when duplicated can cause the 1007 error:

```
"C"
"6800"
 extern funct();
 main()
```


KPR #: D200074989 **CONTINUED**

```
 {
   funct(1,2,3,4,5,6,7,8);
     /*  ...  */
   funct(1,2,3,4,5,6,7,8);
 }
```

The following program fragment illustrates the workaround solutions:

```
"C"
"6800"
 extern funct();
                /* Variables initialized to constants */
 int  one=1; int two=2; int three=3; int four=4;
 int five=5; int six=6; int seven=7; int eight=8;

 workaround()
 {

      /* Use functional type change of short int const to int */
   funct((int)1,(int)2,(int)3,(int)4,(int)5,(int)6,(int)7,(int)8);

      /* Use int variables with constant values */
   funct(one,two,three,four,five,six,seven,eight);

 }
```

Temporary solution:
See problem text.

KPR #: D200079624 Product: 6800 C 64821 01.07

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}
```

KPR #: D200079624 **CONTINUED**

```
memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
       }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081539 Product: 6800 C 64821 01.07

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )     /* gives "Illegal type of operand(s) */
          ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;
```

OR

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

KPR #: D200081539 **CONTINUED**

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200085787 Product: 6800 C 64821 02.10

One-line description:
Type cast of constant to (char *) in pointer expression error

Problem:
Text:
Type cast of constant to (char *) in pointer expression error

Problem: Use of CHAR pointer type casting may cause error in pointer expressions.

```
"C"
"6800"
 char *st_ptr;
main(){
 st_ptr = "any_string" - (char *)1; /* Computes bad address */
/* WORKAROUND 1 */
 st_ptr = "any_string";
 st_ptr -= 1;              /* Correct decrement -1 */
/* WORKAROUND 2 */
 st_ptr = "any_string" - 1;      /* Correct decrement -1 */
}
```

EXPANDED CODE EXAMPLE:

```
  st_ptr = "any_string" - (char *)1; /* Computes bad address */
             LDX  #CONST_prog
             LDAA #0FEH          ;Should be #0FFH
             LDAB #0FBH          ;Should be #0FFH
             JSR  LEAX_D_X
             STX  Dstatic
 /*WORKAROUND*/
  st_ptr = "any_string";
             LDX  #CONST_prog
             STX  Dstatic
```

KPR #: D200085787 **CONTINUED**

```
  st_ptr -= 1;
            DEX
            STX  Dstatic
  /* or this */
  st_ptr = "any_string" - 1; /* Computes correct address */
            LDX  #CONST_prog
            DEX
            STX  Dstatic
?                        ^515 /* Ignore warning  code OK*/
 }
515: Warning: integer not pointer size
```

Temporary solution:

```
"C"
"6800"
 char *st_ptr;
main(){
 st_ptr = "any_string" - (char *)1; /* Computes bad address */


/* WORKAROUND 1 */
 st_ptr = "any_string";
 st_ptr -= 1;              /* Correct decrement -1 */
/* WORKAROUND 2 */
 st_ptr = "any_string" - 1;     /* Correct decrement -1 */


}
```

KPR #: D200085803 Product: 6800 C 64821 02.10

One-line description:
SHORT_ARITH OFF for some short experssions used as conditional branch

Problem:
Text:
SHORT_ARITH OFF for some short expressions used as conditional branch

Problem: With the SHORT_ARITH option OFF, the 6800 compiler does not execute full K&R C code correctly for certain mixed aritmetic operations when used in "if" expressions.
Problems occur when 8-bit (short) aritmetic is used, rather than full expansion to 16 bit values to performa operations as in the standard K&R.

EXAMPLE:

```
"C"
"6800"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
```


KPR #: D200085803 **CONTINUED**

```
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */
 /* WORKAROUND */
 if ((int)s<<4); /*Result is 64*4=1024 which is <>0 should branch here*/
  else  ;
}
```

The 6800 C compiler computes mixed expressions correctly, as in assignment statements and parameter expressions.
This defect appears only when mixed expressions are used without assignment as conditional branching expressions.

This problem may be generated with other operators besides the "<<" as in the example, such as ">>" , "/" and "%

EXPANDED example:

```
  "C"
                          EXTERNAL entry
  "6800"
  short s,ss;
  main(){
        main
   $SHORT_ARITH OFF$
   if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
            LDAB Dstatic  ; This is correct ONLY $SHORT_ARITH OFF$
            ASLB
            ASLB
            ASLB
            ASLB
            BNE  main01_7
            JMP  main01_1
        main01_7
            JMP  main01_2
        main01_1
    else  ;  /* Code branches here, due to use of byte arithmetic. */
        main01_2
   /* WORKAROUND */

   if ((int)s<<4); /*Result is 64*4=1024 which <>0 should branch here*/
            LDAB Dstatic
            JSR  SEXtend
            JSR  TFR_DtoX
            LDAB #004H
            JSR  Zwshift
            JSR  TFR_DtoX
            CPX  #00000H
            BNE  main01_8
            JMP  main01_3
        main01_8
            JMP  main01_4
        main01_3
    else  ;
        main01_4
```

KPR #: D200085803 **CONTINUED**

Temporary solution:
EXAMPLE:

```
"C"
"6800"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */


 /* WORKAROUND */
 if ((int)s<<4); /*Result is 64*4=1024 which is <>0 should branch here*/
  else  ;
}
```

KPR #: D200059980 Product: 6800 PASCAL 64811 01.09

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:
I am expanding this to all pascal compilers. The C compilers list the correct offset. $FAR ON$ only applies to the 68000 cross compiler. The other compilers exhibit the defect w/o any options on.

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200093534 Product: 6800 PASCAL 64811 02.00

One-line description:
Type casting the ADDR function to SET for masking may cause error

Problem:
Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

will geneate incorrect code.

KPR #: D200093534 **CONTINUED**

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

HERE is an expanded example:

```
"PASCAL"
"6800"
PROGRAM Error;
$EXTENSIONS$
TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;
VAR
    S : SET_OF_BITS;
    Byte1,Byte2: BYTE;
    I : SIGNED_16;
PROCEDURE BadADDRsetMASK;
 BEGIN
  Byte1:=BYTE(
  (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
           ^505
        LDAA #000H
        LDAB #0FFH
        JSR  Zuintmul          <----------Should be AND function
        STAA DLSDqf02563
        STAB DLSDqf02563+00001H
        LDAB DLSDqf02563
        STAB DPTEST110+00002H
  END;
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
TempADDR := ADDR(variable);

Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: D200093542 Product: 6800 PASCAL 64811 02.00

One-line description:
Large Sets may produce invalid results for elements outside set range

Problem:

KPR #: D200093542 **CONTINUED**

The set inclusion operation may test undefined bit when the element being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT   {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}     { Branch should always be FALSE,        }
    ELSE   {...}     {  But the result is due to invalid bit test}
END.
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the the full 256 bit set implementation.

Instead of defining the large set as:

```
DIG = SET OF '0'..'9';
```

It could be rewritten:

```
digch = SET OF CHAR;
```

With the sets now using a full 256 bits, all bits will be set and tested properly.

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT   {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}     { Branch should always be FALSE,        }
    ELSE   {...}     {  But the result is due to invalid bit test}
END.
```

KPR #: D200093682 Product: 6800 PASCAL 64811 01.90

One-line description:
Compare using var pointer to first record item fails.

Problem:
When accessing the first item in a record (passed as a VAR pointer parameter), in a comparison expressions, the 6800 code generator fails to call the run time comparison routine properly.

This defect can occur with any date item of size greater than 1 byte (i.e. INTEGER, REAL, LONGREAL).

This defect is only reproducible in the HOSTED compilers:
(64811S001 S500/HPUX, 64811S004 S300/HPUX, 64811S003 VAX/VMS)
It occurs on compilers Rev. 1.90 & Rev. 2.00.

This defect appears to create correct code on the HP64811A 64100 (Rev. 1.90 & Rev 2.000) compiler.

The following source program produces the incorrect code.

```
PROGRAM PTEST32;
TYPE
   WORD = RECORD
            KEY : SIGNED_16;
            LEFT, RIGHT : REF;
          END; (* RECORD *)
    REF = ^WORD;

PROCEDURE POINTERROR ( X : SIGNED_16; VAR P : REF );
BEGIN
   IF X < P^.KEY THEN  ;  (P^.KEY passed to Zintles incorrectly)
END;
```

Temporary solution:
WORKAROUND:
Use a temporary variable:

```
    temp:= P^.KEY;
    IF X < temp THEN ...
```

KPR #: D200093708 Product: 6800 PASCAL 64811 01.90

One-line description:
Asignment of string to double dereference string pointer causes error

Problem:
When attempting to assign a string to a double dereferenced string pointer, the 6800 code generator fails to call the run time string move routine properly.

This defect is only reproducible in the HOSTED compilers:
(64811S001 S500/HPUX, 64811S004 S300/HPUX, 64811S003 VAX/VMS)
It occurs on compilers Rev 1.90 & Rev 2.00.

This defect appears to create correct code on the HP64811A 64100

KPR #: D200093708 **CONTINUED**

(Rev 1.90 & Rev 2.00) compiler.

The following source program produces the incorrect code.

```
PROGRAM doublepointererror;
PROCEDURE BAD_STR_ASN;
TYPE
   STR_ARR = PACKED ARRAY [0..7] OF CHAR;
   ARR_PTR = ^STR_ARR;
VAR
   PTR2 : ^ARR_PTR;
BEGIN
   PTR2^^ := "XY";  {======= BAD call to STmove routine ======}
END;    {BAD_STR_ASN}
BEGIN
END.
```

Temporary solution:
Use a temporary pointer to the final string:

```
  VAR PTR1: ARR_PTR;
  ...
  PTR1:= PTR2^;  PTR1^:= "XY"
```

KPR #: D200093716 Product: 6800 PASCAL 64811 01.90

One-line description:
Pointer dereference of VAR pointer to structure as a parameter fails.

Problem:
A parameter passed in as a VAR pointer to a structure can not be passed properly as a parameter to another routine. The 6800 code generator fails to call the routine properly.

This defect is only reproducible in the HOSTED compilers:
(64811S001 S500/HPUX, 64811S004 S300/HPUX, 64811S003 VAX/VMS)
It occurs on compilers Rev 1.90 & Rev 2.00.

This defect appears to create correct code on the HP64811A 64100 (Rev 1.90 & Rev 2.00) compiler.

The following source program produces the incorrect code.

```
PROGRAM VARerror;
TYPE
  ARTIKEL = RECORD
            ELE1 : INTEGER;
            ELE2 : INTEGER;
            END;
  ARTIKEL_PTR = ^ARTIKEL;
PROCEDURE VarParamRec(VAR ART : ARTIKEL);  EXTERNAL;
PROCEDURE VarParam(VAR ART : ARTIKEL_PTR);
BEGIN
VarParamRec(ART^);  { Parameter NOT passed properly }
END;
```

KPR #: D200093716 **CONTINUED**

```
BEGIN
END.
```

Temporary solution:
Use a temporary pointer variable.

```
  VAR Temp_ptr: ARTIKEL_PTR;
  ...
  Temp_ptr:= ART;
  VarParamRec(Temp_ptr^);
```

KPR #: 5000151050 Product: 6800-03 ASSM M 64841-90905 01.15

One-line description:
Mask pseudo works incorrectly in certain cases.

Problem:
The mask psuedo does not work in the following program.

```
"processor"

        MASK    0FFH
        LDX     #'AB'       ;'A' is masked with 0, and B is masked
                            ;with F.
```

It appears that the assembler is using the required leading zero as a mask value.

Temporary solution:
Specify a four-byte mask value.

```
"processor"

        MASK    0FFFFH
        LDX     #'AB'
```

KPR #: 5000221200 Product: 6800-03 ASSM M 64841-90905 01.15

Keywords: MANUAL

One-line description:
Support OIM, AIM, EIM, TIM

Problem:
This request was originally that we support the AIM, OIM, TIM, and EIM instructions. We do support those instructions, but, this fact is not documented in the manual. The manual should be updated to include this information.

Temporary solution:
No temporary solution.

KPR #: 5000117002 Product: 6800/2 ASSEMB 64841 01.13

One-line description:
Comments are listed in the xref table when not delimited by a ;

Problem:
Comments are listed in the cross reference table as labels when the comments are not delimited from the code with a semi-colon(;).

```
       "processor name"
                    opcode
                    opcode
                    opcode          This is a comment
                    opcode
```

"This", "is", "a", and "comment" will be listed in the cross reference table as labels.

Temporary solution:
To avoid having comments listed in the cross reference table, used a semi-colon (;) to delimit the comment from the code.

Duplicate Service Requests: D200065664 D200065854 D200065862

KPR #: 5000166983 Product: 6800/2 ASSEMB 64841 01.15

One-line description:
External MASKS are not handled properly by the assembler.

Problem:
The bit instructions (BSET, BCLR, BTST) do not handle externally defined masks properly.

"6301"

```
MASK            EQU             7
                EXTERN          MASK2


                BCLR            MASK    30
                BCLR            MASK2   30
                BCLR            7       30
                END
```

MASK2 is an external symbol, but, the assembler interprets it as a zero value therefore it generates a mask of FEH.
The linker then adds the actual mask value and a legal range error is generated.

Temporary solution:
Use an include file to define the MASK (bit) values. This does not accomodate the linker XREF desires of the customer.

KPR #: 5000226563 Product: 6800/2 ASSEMB 64841 01.40

Keywords: PROBLEM ON 9000/S500

One-line description:
Xref table is not listing all symbol references.

Problem:
The following program when assembled with option XREF will demonstrate a problem. The problem is that the XREF will not list all symbol references.

"6301"

```
          ORG           80H
MAP       RMB           1
          ORG           0C000H
          AIM           80H  MAP
          OIM           80H  MAP
          TIM           80H  MAP
```

Temporary solution:
No temporary solution.

KPR #: 5000255752 Product: 6800/2 ASSEMB 64841 00.00

Keywords: PROBLEM ON 9000/S500

One-line description:
Very long file causes problems with xref listing on a 2563B

Temporary solution:
No temporary solution at this time.

KPR #: 5000273458 Product: 6800/2 ASSEMB 64841 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
6301 AIM instruction with ".NT." operator causes LR error.

Problem:
6301 AIM instruction with ".NT." operator causes LR error.

Example assemble list

```
                      1 "6301"
                <00E2> 2 RXFLG   EQU   0E2H
      0000 867F       3          LDAA  #.NT.80H
      0002 717FE2     4          AIM   #.NT.80H RXFLG
  ERROR-LR                               ^
      0005 717FE2     5          AIM   #07FH RXFLG
                      6          END
```

I talked to Dave Ritchie about this - he said it WAS a problem. However, I was unable to obtain a manual for the 6301 to verify if this is correct op-code syntax.

KPR #: 5000273458 **CONTINUED**

David Landoll

Temporary solution:
No workaround available.

KPR #: 5000273474 Product: 6800/2 ASSEMB 64841 01.10

One-line description:
Assembler allows the inst. "LDA A". "LDA A" isn't a valid instruction.

Problem:
```
"6800"
LAB0  EQU  0
LAB1  EQU  1
      LDA A LAB0
      LDA A LAB1
```

Assembler generates the xref of "LDA A LAB0(1)" on 64000, but not on 64000ux. If I change "LDA A" to "LDAA", it generates the xref on 64000ux ,too.

Temporary solution:
There is no workaround available.

KPR #: 5000415786 Product: 6800/2 ASSEMB 64841 01.80

Keywords: PROBLEM ON 9000/S300

One-line description:
XREF missing some labels.

Problem:
Assembler does not generate xref for "LDA A label".
Example:

```
     1  "6800"
     2  label      EQU    1
     3             LDA A label

    xref
        LINE#    SYMBOL    TYPE     REFERENCE
          2      label       A               <--- 3 should be here.
```

Temporary solution:
None.

KPR #: 5000417808 Product: 6800/2 ASSEMB 64841 01.80

Keywords: PROBLEM ON 9000/S300

One-line description:
XREF not properly generated.

Problem:

KPR #: 5000417808 **CONTINUED**

Assembler does not generate xref table for the following instructions.
ORA, AND, BIT, EOR, CMP, SBC, STA, SUB, ADC

Example.
```
"processor"
label   FCB  10
        ORA A label   ---
        AND A label   ---   Xref table does not include these line
         ......             numbers.
```

Temporary solution:
Do not insert space between the instruction and the register name.
Use " ORAA label" instead of "ORA A label".

KPR #: D200089631 Product: 68000 12MHZ EMUL FW 64742 00.04

One-line description:
Slow Clock interferes with configuring monitor... Poor error messages.

Problem:
When an external clock is selected, but the target system is not powered-up, the configuration command

```
cf mon=fg..1000H@s
```

fails, with the message

```
!ERROR ! Invalid configuration value: fg..1000H@s
```

The message does not give any hint that the real problem is the fact that there's no clock.

This is a problem for the HPUX interface, because there is no message indicating slow clock. Clock source and monitor type are set up within configuration, and incompatibilities are not found until the entire config has been entered; the user has no clue to the order in which the config commands were sent to the pod.

Temporary solution:
There is no workaround.

KPR #: D200091587 Product: 68000 12MHZ FUI DOS 64742S006 01.01

One-line description:
All states requested from emtrdata should be valid

Problem:
There is a problem with the emtrdata() input structure. The values contained by "startline" and "endline" should always be valid, regardless of their magnitude. Any value that is out of bounds should be accepted and changed to the maximum range acceptable by the specific analyzer.

Temporary solution:
There is no workaround available.

KPR #: D200092452 Product: 68000 12MHZ FUI DOS 64742S006 01.01

One-line description:
emtrdata() does not work correctly if upload too big linearray

---

KPR #: 5000243048 Product: 68000 ASSEMB 64845 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
Missing whitespace is not flagged.

Problem:
The following code should cause an error to be generated because no whitespace is included. Correct code is generated.

```
"68000"

        DC.B08
        DC.W10
        DC.L15
```

KPR #: 5000247437 Product: 68000 ASSEMB 64845 01.12

One-line description:
Size qualifiers in cross reference.

Problem:
The instruction size qualifiers are listed in the xref as undefined symbols (64000 only).

```
"68000"

               MOVE.B   D0,D1
```

'B' is listed in xref.

Signed off 02/03/89 in release A02.11

KPR #: 5000258590 Product: 68000 ASSEMB 64845 01.13

One-line description:
Math operators not working on 64100.

Problem:
Math operator not working properly on 68000 assembler on 64000 host. Example:

```
"68000"
LAB    DC.W   ((-1*12A9H.SL.0.SR.0).SL.15)/10000
  ^ ERROR LR
LAB1   DC.W   0F6AB8000H/10000  ; should be same answer as above
  ^ ERROR LR
LAB3   DC.W   -156532736/10000 ;works OK
       END
```

These all assemble without error on the 300 and VAX, and return the correct result (0C2DB).

Temporary solution:

KPR #: 5000258590 **CONTINUED**

Make the calculation yourself as is done in the last expression (for LAB3).

KPR #: 5000270637 Product: 68000 ASSEMB 64845 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
No A5 prompt when non-existient .R file specified.

Problem:
The error output is not correct when the following linker command file is used.

```
 segment
 object files         test1.R test2.R test3.R
 library files
 load address         1000H,2000H,0,0 <----------68000 file A5
```

The problem is that test3.R does not exist. Rather than reporting that test3.R does not exist the linker gives the following error:

```
    3 load address must be specified.
```

Temporary solution:
No temporary solution.

Duplicate Service Requests: 5000270215

KPR #: D200086678 Product: 68000 ASSEMB 64845 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
TITLE directive inserting garbage control characters.

Problem:
The following program demonstrates a problem with the titlte directive. If you 'vi' the output file you will see that garbage characters are in the output file.

```
"68000"


         TITLE      "i changelength"

         MOVE       D0,D1
```

Temporary solution:
No temporary solution at this time.

KPR #: 5000163048 Product: 68000 C 64819 01.08

One-line description:
Pass 3 error 1113 flagged.

Problem:
Error 1113 is flagged in the following program.

```
"C"
"68000"

main()
{

  int  intc;
  int  *int_ptr;

  int_ptr = & intc;

  *(int_ptr + intc)--;          /* Error 1113 flagged. */
  intc++;                       /* Doesn't appear in listing. */
  intc--;                       /* Appears in listing. */

}
```

Temporary solution:
Break the instruction into two parts as demonstrated below.

```
"C"
"68000"

main()
{
  int   intc;
  int   *temp,*int_ptr;

  int_ptr = &intc;

  temp = int_ptr + intc;
  *(temp)--;

}
```

KPR #: 5000173815 Product: 68000 C 64819 01.09

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"
```


KPR #: 5000173815 **CONTINUED**

```
struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: 5000192054 Product: 68000 C 64819 01.09

One-line description:
Fields of a structure are dereferenced incorrectly (if fields are big).

Problem:
Structure ponters are not being calculated correctly when relative addressing requires offsets of large sizes. See following code.

```
"C"
"68000"

$FAR$

struct this{
         unsigned short int first[256][256];
         unsigned short int second[256][256];
} one,*bufptr;

unsigned short int *desptr;

main()
{
 bufptr = &one;
 destptr = bufptr->first[0][0];
 destptr = bufptr->second[0][0];   /* Same address assigned. */
 }
```

NOTE: The 8086 line of compilers will generate a pass three error for this code. "Program Counters Disagree. "

Temporary solution:
For the 68000 family of cross compilers you may use the '.' operator instead.

KPR #: 5000192054 **CONTINUED**

"C"
"680XX"

FAR

```
struct  this {
         unsigned short int  first [256][256];
         unsigned short int  second[256][256];
} Structure,*ptrToStruct;

unsigned short int *destptr;
main()
{

 destptr = &Structure.second[0][0];
}
```

KPR #: 5000220418 Product: 68000 C 64819 01.10

One-line description:
Address is not incremented past 0xFFFF for data areas > 32k.

Problem:
In the expanded listing the address of the
variable declarations is not shown when the size of the
data area is greater than FFFFH. I am refering to the
top of the listing where the C declarations of the
arrays are made.
In the expanded listing the address of the
variable declarations is not shown when the size of the
data area is greater than FFFFH. I am refering to the
top of the listing where the C declarations of the
arrays are made.

Temporary solution:
No temporary solution at this time.

KPR #: 5000226530 Product: 68000 C 64819 01.10

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself
as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
        ;
}
```

Customer feels that this variable should be evaluated to see if it
is a non-zero float value.

WORKAROUND:

KPR #: 5000226530 **CONTINUED**

Use `if( x != 0.0 ) ;`

OR

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

"C"
"processor"

```
main()
{

float i;

if ( i != 0)
;

}
```

KPR #: 5000264481 Product: 68000 C 64819 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Problem with Type Name cast - causes Pass 1 error.

Problem:
Type Name cast causes Pass 1 error with no info other than:
"comp: C Pass 1 cannot recover from errors, parsing stopped at
line .."
Example:
"C"
"68010"

```
extern  unsigned short *list[];
int   main()
{
    unsigned short (*maddr)[6];
    maddr = (unsigned short(*)[6])list; /* this line causes the */
}                                       /* problem             */
```

This operation is discussed on pages 199 and 200 of K&R.

The problem seems to come from an error in the way the pointers are
assigned, i.e. if the program is changed to:

```
. . .
{
```

KPR #: 5000264481 **CONTINUED**

```
    unsigned short *maddr[6];
    maddr = (unsigned short (*)[6])list; /* . . .
. . .
```

it seems to compile.
But, according to K. & R., this should be an lvalue error.

Temporary solution:
Change program structure to form a new block - then redefine the variable type as needed inside this block. Upon exiting this block the original type cast will resume.

e.g.

```
"C"
"68010"
extern unsigned short *list;
int main()
{
       {           /* new block */
          unsigned short (*maddr)[6];
          unsigned short (*list)[6]; /* redefined var. type */
          maddr = list ;  /* simplified assignment statement */
        }
     /* Rest of program
        goes here . . . note that the global type defn.
        for the variable "list" now takes over */
```

This isn't pretty, but it is functional.

KPR #: 5000269407 Product: 68000 C 64819 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
The EXT.L command does not work properly.

Problem:
The problem seems to be with the EXT.L instruction itself - when the code is executed D0 contains 'FFFF8000h' (it should contain '00008000h'. Somehow the negative flag in the status register is getting set. Then, when the EXT.L is executed, the computer thinks the number it is expanding is negative - and it expands it with 2's compliment using 'F's instead of zeros.

NOTE: This problem has come up in other places, too. See SR #5000271957.

Temporary solution:
Cast to an unsigned long;

```
    ul = (unsigned long)0x8000;
```

KPR #: 5000269415 Product: 68000 C 64819 02.10

One-line description:
Bad code is generated when a char var is compared to a negative number

Problem:
Bad code generated when a var that is type char is compared to a negative number. Example:

```
"C"
"68000"
char  c;
main()
{
    if( c == -1 )
              c = 1;
}
```

the problem is that 8 bit value is moved to a register via a MOVE.B instruction, but then compared to 0FFFFH via a CMPI.W instruction.

Temporary solution:
Cast the -1 to a char:

```
    if( c == (char)-1 )
```

KPR #: 5000271957 Product: 68000 C 64819 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
Problem with EXT.L command.

Problem:
When we assigns the address value to the pointer, the address is converted to a long by EXT.L. The address is 8000H thru 0FFFFH. In result, the pointer points to unexpected memory location.
example;

```
          "C"
          "68000"
           main()
             {
               int *a;
               a=0x8000;
                  EXT.L is used here.
             }
```

The problem seems to be with the EXT.L instruction itself - when the code is executed D0 contains 'FFFF8000h' (it should contain '00008000h'. Somehow the negative flag in the status register is getting set. Then, when the EXT.L is executed, the computer thinks the number it is expanding is negative - and it expands it with 2's compliment using 'F's instead of zeros.

NOTE: This problem has come up in other places, too. See SR #5000269407.

Temporary solution:
workaround:

KPR #: 5000271957 **CONTINUED**

Use a cast and change the type of address.
a=(long)0x8000
This statement generates MOVE.L instruction instead of MOVE.W and EXT.L.

KPR #: 5000406058 Product: 68000 C 64819 02.10

One-line description:
Cannot combine 'shift' with '&' in same statement for structured var.

KPR #: 5000407197 Product: 68000 C 64819 02.10

One-line description:
"variable = variable = constant" causes failure.

Problem:

```
int func (key)
char key [];
{
key[0] = key[1] = 0;
}
```

fails with too many errors.

Temporary solution:

```
{
key[0] = 0;
key[1] = 0;
}
```

KPR #: D200004929 Product: 68000 C 64819 00.21

Keywords: CODE GENERATOR

One-line description:
Multiple assignments may cause compiler to reuse an overwritten reg.

Problem:
Statements requiring extensive use of address reisters may cause the compiler to use all available registers and then reuse a register that has been over-written. Such a situation is encountered in the following source line where address register A0 originally contains the base address of the array of structures xx[]. By the time that the assignment xx[i].x = xx[i].y is made A0 has been over written to contain the address of xx[i], but it is used as if it still contained xx[0].

```
struct { double x, y, z;
       } xx[2];
int i;
double d;
main()
{
  xx[i].x = xx[i].y = xx[i].z = d;
}
```


KPR #: D200004929 **CONTINUED**

Temporary solution:
Break such complex statements up into simpler statements; i.e., three separate assignments.

KPR #: D200014399 Product: 68000 C 64819 01.07

Keywords: CODE GENERATOR

One-line description:
Bad code using $OPTIMIZE$ and successive uses of the same pointer.

Problem:
Sometimes bad code is produced when $OPTIMIZE ON$ is in effect and one dereferences a pointer, updates the pointer, and dereferences the pointer again. For example,

```
char *p,c;
main()  {  $OPTIMIZE ON$
   c = *p;     /* Dereference a pointer */
     MOVEA.L Dstatic[A5],A0      ; p is loaded into A0
     MOVE.B  [A0],Dstatic+4[A5]
   p += 1;     /* Update the pointer */
     MOVEA.L Dstatic[A5],A1
     LEA     1[A1],A2            ; updated value of p is in A2
     MOVE.L  A2,Dstatic[A5]
   c = *p;     /* Dereference the pointer again */
     MOVE.B  [A0],Dstatic+4[A5]  ; ERROR - A0 contains the old value
                                 ; of p, not the updated value.
```

Temporary solution:
Turn $OPTIMIZE OFF$ around the operations of the above type.

KPR #: D200032045 Product: 68000 C 64819 01.07

One-line description:
Compiler uses MSB of word containing char value rather than LSB.

Problem:
When a character is passed as a parameter, its value is loaded into the lower byte of a word. The most significant byte of this word is passed as the address of this character. Our compiler doesn't realize that the actual ascii value is in the lower byte. So, when this character is referenced, the compiler uses the contents of the most significant byte rather than the lower byte which contains the ascii value. See code and comments below.

```
"C"
"68000"
/*The following sequence of function calls will duplicate the problem.*/
main()
{ putchar('5');
}

putchar(parm)
char parm;
{
```

KPR #: D200032045 **CONTINUED**

```
write(&parm);
}

write(p1)
char *p1;
{
 char temp;
 temp = *p1;       /* temp is loaded with zero rather than the ascii
}                     value for '5' */
```

The intermediate function call to putchar is needed to duplicate this problem. Simply calling write from main will not duplicate the problem.

Temporary solution:
Do not declare parameter to be of type character. Instead define it as an integer and declare a local variable which is intialized to the parameters. SEE CODE BELOW:

```
"C"
"68000"
main L)
{
  putchar ('5');
}

putchar (parm)
int  parm;          *1 Here is the change *1
{
  local_variable=parm;

  Write (&local_variable);
}
Write (p1);
char *p1;
{
  char temp;
  temp=*p1;
}
```

KPR #: D200063115 Product: 68000 C 64819 01.09

One-line description:
Shift of wrong sized value in register.

Problem:
Shift right sometimes generates incorrect result.

```
"C"
"68000"

int fct1(param)
unsigned short param;
{
unsigned short data;
data = param;
```


KPR #: D200063115 **CONTINUED**

```
/* Zeroes are shifted in for the right shift below. */

data = data >> 7 | data << 1;
return (data);
}


int fct2(param)
unsigned short data;
data = param;

/* The word size of param is in a register. The next statement
   uses this word size value of param and this results in the
   upper byte of the word being shifted into data instead
   of zeros. */

data = data >> 7 | data << 1;
return(data);
}
```

Temporary solution:
No temporary solution at this time.

KPR #: D200065193 Product: 68000 C 64819 01.09

One-line description:
An "if" statement may cause the compiler to go astray.

Problem:
An if statement causes a branch to be generated in case the condition is false. If the code space below the if statement is >32 in size then the branch goes off into the weeds. Use the following program to demonstrate the problem.

```
"C"
"680000"

main()
{
   int i;

   if(i==1) {

       /*This code must be >32K */
       i++;i++;i++;i++;i++;i++;i++;i++;   /*Repeat 1035 times. */
  }
}
```

Temporary solution:
Don't create such large blocks within an if statement. Perhaps the code below the if can be put in a sub-routine.

KPR #: D200069674 Product: 68000 C 64819 01.09

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede
the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which
is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function.
For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200071829 Product: 68000 C 64819 01.09

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside
in DATA space. This prevents these libraries from being used in
a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing
constants that are used by the routine SINCOS:NS8086. The "DATA"

KPR #: D200071829 **CONTINUED**

pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be
in the DATA area, but the majority of the constants are also being
loaded in the DATA area. Since the libraries are shipped in relocatable
form only, the customer must wait for the factory to send the sources
to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory,
remove the DATA pseudo, and reassemble.

KPR #: D200076513 Product: 68000 C 64819 01.10

One-line description:
Address comparisons for variables located on negative base-page may fail

Problem:
Address comparisons may not work with $BASE_PAGE$ variables loaded in
the address range 0FF8000H-0FFFFFFH.

In particular, addresses in that range generated when initializing
pointers will be created by the linker with the upper byte set to
00H for the ABSOLUTE address. They will fail if compared to
ABSOLUTE_SHORT addresses (which perform sign extension on the
address setting the upper byte to 0FFH) generated with PEA or
LEA instructions executed at run time.

The following program illustrates the problem:

```
      "C"
      "68000"
      $BASE_PAGE$
       int other[16];
       int *p =other;
       unsigned int mask16=0xffff;
       main()
       {
0000      main
0000           LINK     A6,#0
       if ( p!=other )  ;
0004           ABSOLUTE_SHORT
0004           MOVE.L   Dstatic+00020H,D0  ### 00FF8XXXH
0008           LEA      Dstatic,A0          ### FFFF8XXXH
000C           CMP.L    A0,D0              #FAIL Compare HERE#
000E           BEQ      main01_1

      /* Workaround # mask addresses if $BASE_PAGE$ */
0012      main01_1
      if ( ( (int)p&(mask16) ) != ( (int)other&(mask16) ) )  ;
0012           MOVE.L   Dstatic+00020H,D1  ### 00FF8XXXH
```

KPR #: D200076513 **CONTINUED**

```
0016          AND.W    Dstatic+00024H,D1  ### 0000FFFFH
001A          LEA      Dstatic,A1         ### FFFF8XXXH
001E          MOVE.L   A1,D2
0020          AND.W    Dstatic+00024H,D2  ### 0000FFFFH
0024          CMP.W    D2,D1              #OK# 0000XXXXH (D2)
0026          BEQ      main01_2           #==# 0000XXXXH (D1)
002A      main01_2
      }
0000          DATA
0000      Dstatic
0000          DC.B     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0010          DC.B     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0020          DC.L     Dstatic     ### Produces 00FFXXXXH value in link
0024          DC.W     -1          ### 0FFFFH
```

Temporary solution:
See problem text for a work around.

KPR #: D200079590 Product: 68000 C 64819 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
```


KPR #: D200079590 **CONTINUED**

```
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
       }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081505 Product: 68000 C 64819 01.10

One-line description:
Libraries generate incorrect code 68010 processor.

Problem:
The 68000/08/10 compilers all share the same library routines. In the function Zenter_trap an ADDA.L #6 instruction is used to point to the parameters in the trap procedure. The problem is the 68010 pushes the Vector base register so the instruction should use an immediate value of 8.

Temporary solution:
Contact your local Applications Engineer to obtain a copy of the sources for the libraries.

KPR #: D200085373 Product: 68000 C 64819 01.10

One-line description:
SHORT_ARITH OFF use of mixed short int in conditionals may not work

Problem:
With the SHORT_ARITH option OFF, the 68000 compiler does not execute full K&R C code correctly for certain mixed arithmetic operations when used in "if" expressions.

EXAMPLE:

```
  main() {
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
  else
    /* This statement executes erroneously */
             /* But using byte arithmetic  produces FALSE=0 */
```

Detailed Listing for Defect Number LSDqf02058
Text:

KPR #: D200085373 **CONTINUED**

SHORT_ARITH OFF use of mixed short int in conditionals may not work

Problem: With the SHORT_ARITH option OFF, the 68000 compiler does not execute full K&R C code correctly for certain mixed arithmetic operations when used in "if" expressions.

EXAMPLE:

```
  main() {
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */
  /* WORKAROUND */
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
   else;
  }
```

The 68000 C compiler computes mixed expressions correctly, as in assignment statments and parameter expression.
This defect appears only when mixed expressions are used without assignment as conditional branching expressions.

This problem may be generated with other operators besides the "+" as in the example.

EXPANDED example:

```
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
          MOVE.W   #0FFFDH,D0
          MOVE.B   D0,-2[A6]
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
          ADDQ.B   #3,D0
          BEQ      main01_1
          BRA      main01_2
      main01_1
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */

  /* WORKAROUND */
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
          CLR.L    D1
          MOVE.B   -2[A6],D1
          ADDQ.W   #3,D1
          BEQ      main01_3
          BRA      main01_4
      main01_3
```


KPR #: D200085373 **CONTINUED**

```
   else;
      main01_4
```

Temporary solution:
EXAMPLE:

```
  main() {
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */



  /* WORKAROUND */
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
   else;
  }
```

KPR #: D200085399 Product: 68000 C 64819 01.10

One-line description:
SHORT_ARITH OFF with unsigned short int in conditional branch error

Problem:
This problem may be generated with other operators besides the "+" as in the example.

EXAMPLE:

EXPANDED example:

```
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
          MOVE.W   #0FFFDH,D0
          MOVE.B   D0,-2[A6]
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
          ADDQ.B   #3,D0
          BEQ      main01_1
          BRA      main01_2
      main01_1
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */
```

Temporary solution:

```
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
```

KPR #: D200085399 **CONTINUED**

```
         CLR.L    D1
         MOVE.B   -2[A6],D1
         ADDQ.W   #3,D1
         BEQ      main01_3
         BRA      main01_4
     main01_3
  else;
     main01_4
```

KPR #: D200093294 Product: 68000 C 64819 02.10

One-line description:
Zlongreal_sub library error.

KPR #: 1650064923 Product: 68000 C M 64819-90902 01.09

One-line description:
error 1113

KPR #: 5000184374 Product: 68000 C M 64819-90902 01.09

One-line description:
List library link range in manuals.

Problem:
The 68000 library routines must reside within +/- 32K of each other. This applies to both A5_LIB and ABS_LIB. This caused problems when a customer linked in some of the libraries and burned them into EPROM. He then used that link_sym when linking subsequent code. A new library routine, not previously linked, was loaded. It made reference to a previously loaded library using a BSR which is limited to a +/-32k branch.

Either the libraries should be changed, or the manuals should be updated to reflect this restriction.

Temporary solution:
No temporary solution at this time.

KPR #: D200095182 Product: 68000 DQ EMUL 300 64243S004 01.30

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

Problem:
Detailed Listing for Defect Number LSDqf04997
Text:
  Open of file pvxxxxfile_asmb causes pv failure on long file name syste
.....
PV software opens a file called "pvxxxxfile_asmb", where xxxx is the idcode of the hardware. We ship a file called "pvxxxxfile_asm". On a short file name system, these two files are considered the same. However, on a long filename system, the open fails, and therefore, pv will not run.

Applies to 64243S004, 64244S004, and 64245S004

Temporary solution:
A tempory fix can be obtained by linking the two files.

KPR #: D200095679 Product: 68000 DQ EMUL 300 64243S004 01.30

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095968 Product: 68000 DQ EMUL 300 64243S004 01.30

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200081372 Product: 68000 EMUL 12.5 MHZ 64243 01.01

One-line description:
State IA generates wrong instruction for Adr Reg. Indirect w/Indexing

Problem:
The assembler syntax of the address register indirect with index addressing mode is d(An,Rn). Rn (the index register) can be either an address or a data register. If the code uses an address register for Rn, the disassembler incorrectly displays the data register of the same number. Refer to the following example:

```
Source Code:        MOVE.W  0[A0,A1].D0
Disassembled Code:  MOVE.W  000H[A0,D1.W],D0
                                 ^----------incorrect value
```

Note that the index register was incorrectly disassembled as D1 instead of A1.

Temporary solution:
No workaround at this time.

KPR #: D200067637 Product: 68000 EMULATION 64242 01.07

One-line description:
Load of more than 1 abs. to targ. mem. not allowed when restricted to RT

Problem:
When restricted to real time, loading more than one absolute file to target memory gives an error when attempting the second load, "ERROR: Command causes break- runs restricted to real time".

The emulator is actually running in the monitor at this time but the emulation software seems to think that it is executing target memory.

Example:

```
emulate CONFIG load MON ;restricted to real-time, load monitor
load FILE1              ;FILE1 mapped to target memory
load FILE2              ;FILE2 mapped to target memory <<<<<
                                 Error occurs here >>^
```

Temporary solution:
Add a "break" between each load

Example

```
emulate CONFIG load MON
load FILE1              ;mapped to target memory
break
load FILE2              ;mapped to target memory
break
load FILE3              ;mapped to target memory
etc.
```

KPR #: D200069484 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Duplicate Service Requests: D200069500 D200069492

KPR #: D200080606 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080903 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

KPR #: D200080903 **CONTINUED**

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081679 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Tracing on status int_ack does not work.

Problem:
Tracing on status int_ack does not work.
To observe an interrupt service routine, "trace about status int_ack" does not work. One must instead trace about an address in the interrupt vector table (0h thru 2ffh).

Temporary solution:
Trace on an address (range) in the interrupt vector table. This will result in capturing nearly the same amount of information.

KPR #: D200081885 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082180 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082180 **CONTINUED**

KPR #: D200082594 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Memory breaks during stepping are not detected

Problem:
Memory breaks (write to ROM, etc.) which occur while stepping will not be noted on the status line.

Temporary solution:
No workaround at this time.

KPR #: D200082610 Product: 68000 EMULATION 300 64242S004 01.00

Keywords: BREAKPOINT

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

Temporary solution:
Do not use software breakpoints in target memory. If software breakpoints are required to debug a certain section of code, make sure that section is mapped to emulation memory.

KPR #: D200083196 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085993 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200090811 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095661 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095950 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200013110 Product: 68000 HL SOFT ANAL M 64331-90902 01.00

One-line description:
Tracing a variable declared as a pointer to a function doesn't work in C

Problem:
If a variable is declared as a pointer to a function (using C) and you then try and trace that variable in Spider an "Invalid type encountered" message will occur. No known workaround situation is currently known.

KPR #: 1650006700 Product: 68000 PASCAL 64815 00.00

One-line description:
Immediate operand's value is altered when doing a logical and.

Problem:
Incorrect code is generated for the below program. Specifically, the immediate value is altered when the logical AND is done.

```
"68000"
$EXTENSIONS$

PROGRAM TEST;

TYPE
  MASK = 0..15;
  INTEGER = SIGNED_16;

VAR
  CNTL_REG, I : INTEGER;


BEGIN
  IF INTEGER(MASK(CNTL_REG)*MASK(SIGNED_16(0FH)) > 9
     THEN;

END.
```

Temporary solution:
The April SMS has fixed this problem. The revisions involved are 1.3 on the 9000, 1.4 on the VAX and 1.10 on the 64100A.

KPR #: 5000161182 Product: 68000 PASCAL 64815 01.12

One-line description:
Bad code when taking ADDR of record element when using WITH.

Problem:
Code generated by the 68000 Pascal compiler is wrong in the following example:

```
"68000"

PROGRAM  test;
$EXTENSIONS ON, EXTVAR ON$

TYPE   x_type = RECORD
         f1   : BYTE;
         f2   : BYTE;
       END;

VAR  x   :  x_type;
     i   :  INTEGER;

BEGIN

   WITH  x  DO
```

KPR #: 5000161182 **CONTINUED**

```
      i :=  INTEGER (ADDR(F2)) - INTEGER(ADDR(F1));

END.
```

The value of F1 is subtracted rather than the address of F1. This problem seems to be unique to the first element of the record.

Temporary solution:
Add a negative first element rather than subtracting it.

```
"68000"
PROGRAM  test;

$EXTENSIONS ON, EXTVAR ON$

TYPE   x_type  =   RECORD
          F1   :   BYTE;
          F2   :   BYTE;
       END;

VAR    x  :  x_type;
       i  :  INTEGER;

BEGIN
   WITH  x  DO
     i = - INTEGER(ADDR(F1)) + INTEGER(ADDR(F2));

END.
```

KPR #: 5000169250 Product: 68000 PASCAL 64815 01.11

One-line description:
Declaring a boolean array may cause an out of bounds error.

Problem:
The Pascal compiler generates an invalid out of bounds error for the following program.

```
"processor"

PROGRAM  OUTOFBOUNDS;
$RANGE+$

TYPE    I               =       INTEGER;
        BOOL_ARRAY      =       ARRAY[1..2] OF BOOLEAN;

VAR     B_ARRAY         :       BOOL_ARRAY;
        B               :       BOOLEAN;

BEGIN

        B  := B_ARRAY[1];
```


KPR #: 5000169250 **CONTINUED**

```
END.
```

If the array type is not boolean then this code compiles correctly. Also, you can put the array declaration first and the invalid error goes away.

Temporary solution:
Declare the boolean array first.

```
PROGRAM  BOOLEAN_ARRAY;

TYPE     BOOL_ARRAY      =       ARRAY[1..2] OF BOOLEAN;

VAR      B_ARRAY         :       BOOL_ARRAY;
         BOOL            :       BOOLEAN;
         INT             :       INTEGER;

BEGIN
   BOOL := B_ARRAY[1];
END.
```

KPR #: 5000183913 Product: 68000 PASCAL 64815 01.11

One-line description:
Casting address to int and adding a signed_16 var generates bad code.

Problem:
In the following program the address to integer conversion is not always done correctly. In particular if you cast an address to an integer and add a signed_16 variable incorrect code is generated.

```
"68000"

$EXTENSIONS ON$
$GLOBPROC ON$

PROGRAM TEST;

VAR   INT  :  INTEGER;
      SHORT:  SIGNED_16;


PROCEDURE  PROC(VAR PARAM: INTEGER);

VAR   INT1,INT2   : INTEGER;
      SHORT       : SIGNED_16;

BEGIN

   SHORT := 4;
   INT2  := 8;
```

KPR #: 5000183913 **CONTINUED**

```
   INT1  := INTEGER(ADDR(PARAM)) + SHORT;
END;

BEGIN
END.
```

Instead of adding the address of PARAM to SHORT we add the value of PARAM.

Temporary solution:
Cast the signed_16 variable, in this example SHORT, to an integer.

KPR #: 5000196428 Product: 68000 PASCAL 64815 01.11

One-line description:
Bytes sign extened in a case statement.

Problem:
The foloowing program generates wrong code on the 64000 (old) system (Seems to be correct on 300 series host)
"68000"

```
PROGRAM T;
$EXTENSIONS ON$;
VAR A : BYTE;

BEGIN
  CASE A OF
    BYTE(55H) : ;
    BYTE(0F5)H: ;
  END;
END.
```

"A" is extended to a word in 64000 Pascal (longword on 300) and the comp are is done on a word too (byte on 300).If A=0F5H then it is extended to 0FFF5H and compared with 0F5H, which fails.

Temporary solution:
Use a signed_16 variable for the test case.

```
$EXTENSIONS ON$

PROGRAM  TEST;

VAR   A:  SIGNED_16;

BEGIN

    CASE  A OF
      0F5H:  ;
    END;
END.
```

KPR #: D200013359 Product: 68000 PASCAL 64815 01.08

Keywords: CODE GENERATOR

One-line description:
Compiler generates incorrect code for set inclusion check.

Problem:
The following program will display a compiler code generation problem when testing for SET inclusion.

```
  PROGRAM TEST;
  $EXTENSIONS$

     TYPE
        DIG = SET OF '0'..'9';

     VAR
        DIGIT : DIG;
        DA : STRING;
        A, B : INTEGER;

     BEGIN
        DIGIT := DIG['0','1','2','3','4','5','6','7','8','9'];
        DA := 'AAA';
        IF DA[1] IN DIGIT THEN
     END.
```

Temporary solution:
As a temporary work-around use the following TYPE definition for DIG.

```
  TYPE
     DIG = SET OF CHAR;
```

KPR #: D200014332 Product: 68000 PASCAL 64815 01.09

Keywords: CODE GENERATOR

One-line description:
Bad code using $RANGE$ or $DEBUG$ with $CALL_PC_LONG$ or LIB_PC_LONG

Problem:
Bad code is generated when calling functions and the compiler directives $RANGE ON$ or $DEBUG ON$ are used in combination with the directives $CALL_PC_LONG$ or LIB_PC_LONG. For example,

```
   $DEBUG ON, LIB_PC_LONG$     VAR I:SIGNED_16;
   FUNCTION F:SIGNED_16; BEGIN F := 0; END;
   BEGIN   I := F * 2;  {PRODUCES BAD CODE}
        BSR   F          ;CALL F
        MULS  #2,D7      ;MULTIPLY RESULT IN D7 TIMES 2
        MOVE.L D7,-[A7] ;PUSH PARAMETER FOR Zoverflow_s16
        MOVE.L #Zoverflow_s16[PC],D7 ;ERROR!! D7 DESTROYED!!
        JSR   -6[PC,D7.L] ;CALL Zoverflow_s16 VIA PC LONG METHOD
        MOVE.W D7,I      ;WRONG VALUE STORED, D7 CONTAINS BAD DATA!!
```

KPR #: D200014332 **CONTINUED**

Temporary solution:
Avoid the combination of functions, $RANGE$ or $DEBUG$, and $CALL_PC_LONG$ or LIB_PC_LONG. The example above may be rewritten to achieve the same functionality.

```
   I := F;   {STATEMENT DOES NOT CAUSE CALL TO OVERFLOW ROUTINE}
   I := I * 2;   {OVERFLOW ROUTINE CALLED HERE BUT DATA IS NOT IN D7}
```

KPR #: D200049882 Product: 68000 PASCAL 64815 01.10

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200060103 Product: 68000 PASCAL 64815 01.10

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
```


KPR #: D200060103 **CONTINUED**

```
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200060343 Product: 68000 PASCAL 64815 01.10

One-line description:
Compiler generates a LEA instruction with an illegal source operand.

Problem:
The following program causes an illegal 68000 instruction to be generated.

```
"68000"
PROGRAM  CGR_1;
$EXTENSIONS ON$


TYPE    REC     =       RECORD;
        REG0,
        REG1,
        REG2,
        REG3,
        REG4,
        REG5,
        REG6,
        REG7:            UNSIGNED_8;
        END;

VAR     REC1,
```

KPR #: D200060343 **CONTINUED**

```
       REC2:           REC;

       V1  :           UNSIGNED_8;

BEGIN
     REC1.REG0 := SHIFT(V1,5);
     REC2 := REC1;
END.
```

The 'REC2 := REC1' instruction causes a 'LEA D0,A0' 68000 instruction to be generated. This is an illegal instruction because the LEA instruction cannot have a data register as its source operand.

Temporary solution:
Turn $AMNESIA ON$ above the instruction which makes the structure access.

"68000"

```
PROGRAM  CGR_1;
$EXTENSIONS$

TYPE      REC      =         RECORD
                   REG0,
                    .
                    .
                    .
                   REG7:     UNSIGNED_8;
          END;

VAR       REC1,
          REC2:   REC;

          V1  :   UNSIGNED_8;

BEGIN
$AMNESIA OFF$

      REC1.REG0 = SHIFT(V1,5);
      REC2 := REC1;
$AMNESIA OFF$

END.
```

KPR #: D200065045 Product: 68000 PASCAL 64815 01.11

One-line description:
The WARN option cannot be turned off.

Problem:
The WARN option cannot be turned off on the hosts. With WARN off, the host machines (9000/500, 9000/320 and VAX) still generate warning messages on the screen and in the listing file. The 64100 will generate some warning messages (only the duplicate

KPR #: D200065045 **CONTINUED**

macro warning in the following example).

```
"processor"  PREPROCESS

$WARN OFF$

#define byte  SHORT
#define byte  UNSIGNED_8

PROGRAM GENERATE_WARN;

VAR      LSD:                                                    :INT

{The above is intended to generate warning 512 }

$LSD +$

BEGIN
END.
```

Temporary solution:
No temporary solution at this time.

KPR #: D200071696 Product: 68000 PASCAL 64815 01.11

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory, remove the DATA pseudo, and reassemble.

KPR #: D200073007 Product: 68000 PASCAL 64815 01.11

One-line description:
Problems with routine STRWRITE & $BASE_PAGE$ mode with ASPIOLIB

Problem:
I/O error # 2 may occur when using STRWRITE in a program compiled with $BASE_PAGE$ option, in conjunction wtih using the the Pascal IO library ABSPIOLIB.

If the user allows the DATA area for Pascal program variables to be loaded in the memory range 00FF8000H to 00FFFFFFH, an unexpected run time I/O error #2 may occur.

When using the Pascal/64000 string function STRWRITE, the compiler generates a global variable STRfile and calls the routines Pstringopen and Pwrite_string. As part of its error checking mechanism, Pwrite_string performs an address comparison on the variable STRfile as passed by the caller (done as a sign extened SHORT address due to the $BASE_PAGE$ option [0FFFF6238H]) and as created by Pstringopen (done as a LONG address due to the FAR option used in the compililng the ABSPIOLIB routines [000FF6238H]). Since the addresses are found to be different the routine generates the I/O error #2 message.

This error can be avoided by always using the FAR option for any user program which wants to be used with the ABSPIOLIB.

More Information:

I/O error # 2 may occur when using STRWRITE in a program compiled with $BASE_PAGE$ option, in conjunction wtih using the the Pascal IO library ABSPIOLIB.

If the user allows the DATA area for Pascal program variables to be loaded in the memory range 00FF8000H to 00FFFFFFH, an unexpected run time I/O error #2 may occur.

When using the Pascal/64000 string function STRWRITE, the compiler generates a global variable STRfile and calls the routines Pstringopen and Pwrite_string. As part of its error checking mechanism, Pwrite_string performs an address comparison on the variable STRfile as passed by the caller (done as a sign extened SHORT address due to the $BASE_PAGE$ option [0FFFF6238H]) and as created by Pstringopen (done as a LONG address due to the FAR option used in the compililng the ABSPIOLIB routines [000FF6238H]). Since the addresses are found to be different the routine generates the I/O error #2 message.

KPR #: D200073007 **CONTINUED**

This error can be avoided by always using the FAR option for any user program which wants to be used with the ABSPIOLIB.

Workaround solution for this defect:

This error can be avoided by consistent use of the $BASE_PAGE$ and FAR options in user compiled code with the proper A5 or ABS libraries.

Always using the FAR option for any user program which will be used with the ABSPIOLIB.

Always using the $BASE_PAGE$ option for any user program which will be used with the A5_PIOLIB.

Temporary solution:
This error can be avoided by consistent use of the $BASE_PAGE$ and FAR options in user compiled code with the proper A5 or ABS libraries.

Always using the FAR option for any user program which will be used with the ABSPIOLIB.

Always using the $BASE_PAGE$ option for any user program which will be used with the A5_PIOLIB.

KPR #: D200076562 Product: 68000 PASCAL 64815 01.12

One-line description:
Subrange parameter not passed properly when function returning integer

Problem:
A multi - parameter procedure or function with a value parameter of type subrange may not be passed properly in $COMMON$ option mode when the parameter expression includes a function returning a subrange value.

The problem appears to be improper conversion of the 32 bit integer returned from the function and the truncation to 16 bits for the parameter passing.

########## Problem source text ##########

The defect can be reproduced with a call to routine PARMS defined:

```
PROCEDURE PARMS(SP:SUBRANGE; IP:INTEGER; RP:REAL; LP:LONGREAL);

FUNCTION FI1:INTEGER; BEGIN FI1:=1; END;

BEGIN
```

KPR #: D200076562 **CONTINUED**

```
 PARMS(FI1,FI1,FI1,FI1);  {Fails to pass the first parameter properly}
END;
```

Temporary solution:
The workaround would seem to be performing a functional type change on the integer function call:

```
 PARMS( SUBRANGE(FI1) ,FI1,FI1,FI1);  {--Will pass the first parameter
                                         properly. -- }
```

KPR #: D200093450 Product: 68000 PASCAL 64815 02.00

One-line description:
Type casting the ADDR function to SET for masking may cause an error.

Problem:
Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
    Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

will geneate incorrect code.

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

HERE is an expanded example:

```
"PASCAL"
"68000"
PROGRAM Error;
$EXTENSIONS$
TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;
VAR
    S : SET_OF_BITS;
    Byte1,Byte2: BYTE;
    I : SIGNED_16;
PROCEDURE BadADDRsetMASK;
 BEGIN
  Byte1:=BYTE(
  (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
```


KPR #: D200093450 **CONTINUED**

```
                    ^505                                          ^505
        LEA      DPTEST110+00004H[A5],A0
        MOVE.L   A0,-[A7]
        CLR.L    D0
        MOVE.W   #000FFH,D0
        MOVE.L   D0,-[A7]
        JSR      Zunsmult[PC]
        MOVE.W   D7,-2[A6]
        MOVE.B   -2[A6],DPTEST110+00002H[A5]
  END;
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
    Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
    TempADDR := ADDR(variable);
    Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: 5000239012 Product: 68000/08/10 ASM M 64845-90904 01.10

Keywords: MANUAL

One-line description:
Alter all assembler manuals to reflect new syntax.

KPR #: 5000242032 Product: 68000/08/10 ASM M 64845-90904 01.10

Keywords: MANUAL

One-line description:
Manual indicates EXT is a legal psuedo for an external declaration.

Problem:
The assembler outputs an error when assembling:

```
"68000"
       EXT        L1
                  ^
               error -IO
```

The 68000 assembler manual states EXT is an acceptable pseudo-op.

Temporary solution:
Use "EXTERNAL" rather than "EXT" when making external
symbol declarations.

KPR #: D200045864 Product: 68000/08/10 ASM M 64845-90904 01.00

One-line description:
Wrong offset calculated when using PC+index reg+ offset mode of addr.

Problem:
When using the PC relative with offset and index register mode of addressing the assembler may generate a legal range error. The error will be made if the offset symbol is at an absolute location greater than FFH. The correct address is generated.
"68000"

```
     ORG     010H
     MOVE    #0,D0
     JMP     TABLE[PC,D0]
     ORG     100H
TABLE        DS.W     10
```

Temporary solution:
No temporary solution.

Duplicate Service Requests: D200045351

KPR #: D200065565 Product: 68000/08/10 ASM M 64845-90904 01.00

One-line description:
Include support for BHS and BLO.

Problem:
BHS should assemble the same as BCC and BLO the same as BCS. This is more logical for the programmer and matches Motorola's assembler.

Temporary solution:
No temporary solution.

KPR #: 5000163808 Product: 68000/10 RT S-ANAL M 64341-90903 02.00

One-line description:
Non-adjacent symbols not traceable in some conditions.

KPR #: D200095190 Product: 68008 EMULATION 300 64244S004 01.30

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

Problem:
Detailed Listing for Defect Number LSDqf04997
Text:
Open of file pvxxxxfile_asmb causes pv failure on long file name syste
.....
PV software opens a file called "pvxxxxfile_asmb", where xxxx is the idcode of the hardware. We ship a file called "pvxxxxfile_asm". On a short file name system, these two files are considered the same. However, on a long filename system, the open fails, and therefore, pv will not run.

Applies to 64243S004, 64244S004, and 64245S004

Temporary solution:
A tempory fix can be accomplished by linking the two files.

KPR #: D200095687 Product: 68008 EMULATION 300 64244S004 01.30

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095976 Product: 68008 EMULATION 300 64244S004 01.30

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200069567 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200077545 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
State inverse assembler for 6801 does not work

Problem:
Todd Hatfield@Logic Systems Division

The problem with the state inverse assembler for the 6801 has been fixed by the lab engineer. The software being shipped currently with the product has been fixed to work. The bad software which was shipped will be updated with an SMS shipment in the latter part of the summer or the first part of the fall, 1987.

Duplicate Service Requests: D200077537

KPR #: D200080671 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080978 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081927 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082255 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by
cat < ptyxx

KPR #: D200082255 **CONTINUED**

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082727 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Under certain conditions the 6801 may not work correctly with SPA

Problem:
In certain configurations SPA doesn't work with the 68XX series emulators.

There was a similar problem with the JLO 68XX clones, the 63XX series. They implemented a software fix to take care of this problem. The lab is implementing the same fix for the 68XX series.

KPR #: D200083261 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086058 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the

KPR #: D200086058 **CONTINUED**

problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200086397 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088351 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090886 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles,

KPR #: D200090886 **CONTINUED**

relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095737 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200096024 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200095208 Product: 68010 EMUL 12.5M 300 64245S004 01.30

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name sys.

Problem:
Detailed Listing for Defect Number LSDqf04997
Text:
Open of file pvxxxxfile_asmb causes pv failure on long file name syste
.....
PV software opens a file called "pvxxxxfile_asmb", where xxxx is the idcode of the hardware. We ship a file called "pvxxxxfile_asm". On a short file name system, these two files are considered the same. However, on a long filename system, the open fails, and therefore, pv will not run.

Applies to 64243S004, 64244S004, and 64245S004

Temporary solution:
A tempory fix can be obtained by linking the two files.

KPR #: D200095695 Product: 68010 EMUL 12.5M 300 64245S004 01.30

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095984 Product: 68010 EMUL 12.5M 300 64245S004 01.30

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200069534 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200072496 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Incorrect breakpoint behaviour on continuing emulation.

Problem:
When using software breakpoints, and doing the following sequence, emulation does not behave as expected.

```
run LOOP
modify software_breakpoints set LOOP
# breaks into monitor, displays breakpoint and clears breakpoint
run
end  locked
# return to emulation
modify software_breakpoints set LOOP
# breaks into monitor, but does not display or clear breakpoint
# at second iteration, things return to normal.
```

KPR #: D200080648 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080945 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081893 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082222 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

KPR #: D200082222 **CONTINUED**

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082776 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Memory breaks during stepping are not detected

Problem:
Memory breaks (write to ROM, etc.) which occur while stepping will not be noted on the status line.

Temporary solution:
No workaround at this time.

KPR #: D200083238 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086025 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

KPR #: D200086025 **CONTINUED**

5. display trace mnemonic
6. display trace absolute

KPR #: D200087288 Product: 68010 G.P. EMUL 300 64249S004 01.00

Keywords: BREAKPOINT

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

Temporary solution:
Do not use software breakpoints in target memory. If software breakpoints are required to debug a certain section of code, make sure that section is mapped to emulation memory.

KPR #: D200090852 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095703 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095992 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: 5000285742 Product: 68020 ASSEMB 300 64870S004 01.00

Keywords: PROBLEM ON 9000/S300

One-line description:
NOPAGE option does not work for the 68000 assembler.

Problem:
This is the 68000 assembler directive option problem.
Please refer 68000/10/20 Assembler/Linker/Librarian reference manual.
(Manual part number : 64870-90901)
Chapter-6 assembler directive : 'NOPAGE' option could not work.
This option used , but all page eject and page headers were printed.
'NOPAGE' option were entered inthe file in the second(operation) field.

Temporary solution:
There is no workaround available at this time.

KPR #: 5000291294 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
68000 AXLS compiler generates x_ref to _strcpy when it isn't needed.

Problem:
The 68000 AXLS compiler makes an external reference to the library function "strcpy" or "strcmp" even though the routines are not called.
The compiler may generate in-line code for a strcpy or strcmp.
Therefore there may not be a need to generate an "XREF_strcpy" or "XREF_strcmp," but the compiler still does. This forces the user to to either link in the libc.a library, or receive an unresolved reference error when linking.

Example code :

```
char str1[10] = "Hello";
char str2[10] = "Bye";
main()
{
     strcpy(str1, str2);
}
```

Compiler listing:

```
1    char str1[10] = "Hello";
2    char str2[10] = "Bye";
3    main()
4    {
5           strcpy(str1,str2);
00000004 207C 0000 000A  R          MOVE.L  #_str2+0,A0
0000000A 227C 0000 0000  R          MOVE.L  #_str1+0,A1
                           L0_strcpy  <----- Compiler creates internal
                                      <----- strcpy function
 00000010 12D8                       MOVE.B  (A0)+,(A1)+
 00000012 66FC                       BNE.S   L0_strcpy
6    }
```

KPR #: 5000291294 **CONTINUED**

```
 00000014 4E71                  NOP
                        functionExit1
 00000016 4E5E                  UNLK    A6

                                .

                                .

                                .

                                XREF    _strcpy <--- Doesn't need
                                              to be generated.
```

Temporary solution:
To avoid getting the unresolved reference error, link in the libc.a library.

KPR #: D200089276 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Using asm psued END with numeric expression causes linker error.

Problem:
If you use a numeric expression in the assembler END pseudo the linker reports error 318.

```
     SECT  prog
main:  move.l d0,d1
     END  $1000
```

Temporary solution:
Rather than using a numeric expression use the form

```
               END    LABEL
```

and then load that file starting at the address you wish.

KPR #: D200089714 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Ar68k can not handle long list in command line options.

Problem:
DETAILED DESCRIPTION: Ar68k allows the -a, -d, -r, and -e options on the command line. These options may be followed by a list of files or modules.
If the list becomes too long OR if too many of the above options are issued
than ar68k acts badly, usually core dumps or sometimes giving a spurious error. The list seems to be limited to a few hundred characters

It would be better if a very long list were allowed. This would allow an easier interface with "make" as shown in the following example.

MODLIST = very long list of modules ...

KPR #: D200089714 **CONTINUED**

```
all:    $(MODLIST)
        rm -f lib.a
        ar68k -a "$(MODLIST)" lib
```

The following shell script illustrates the problem. It makes about 30 modules with long names. It then tries to archive them using the command line.

A test of the fix could be done using the script below and then comparing
the librarian listing.

```
list1=
list2=
for i in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
22 23
4 25 26 27 28 29 30 31 32
do
        list1="$list1 ar68klist$i.o"
        list2="$list2 -a ar68klist$i.o"
        as68k -o ar68klist$i <<EOF
        sect    a,,c
        xdef    g$i
g$i     dc.w    0
        end
EOF
done
        rm -f lib.a
        # try it as one option with long list
        echo $list1
        ar68k -a "$list1" lib
        # try it as many options with short list
        echo $list2
        rm -f lib.a
        ar68k $list2 lib
```

Temporary solution:
Workaround: Create a libraian command file. For example,

MODLIST = very long list of modules ...

```
all: $(MODLIST)
        rm -f lib.a
        echo create lib > arcmd
        for i in ar*.o ;\
        do echo addmod $i >> arcmd ;\
        done
        echo save >> arcmd
        ar68k < arcmd
```

KPR #: D200089722 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
PLEN directive does not work properly

Problem:
DETAILED DESCRIPTION: The PLEN directive is supposed to adjust the numbe
r
of lines in a listing page. The manual says that the number specified
in the directive should be the total number of lines on the page, includ
ing
headings. Two things are wrong.

1. The number specified does not include heading lines. If I specify PL
EN 50
then I get about 54 lines per page, 49 lines of source and 5 heading lin
es.

2. If I specify a PLEN greater than 55, then the first page is short.
Subsequent pages are OK obeying the rule stated in number 1 above.

Temporary solution:
None.

KPR #: D200089730 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
LLEN directive does not work properly with tab characters

Problem:
DETAILED DESCRIPTION: The LLEN directive is used to specify the width of
the
listing. However, when the source file contains tab characters, these
tabs are counted as 1 character for the purposes of truncation. The out
put
device usually expands these tabs into more than one character. This ma
kes
the listing wider than specified sometimes causing jamming problems with
certain printers.

Temporary solution:
None.

KPR #: D200089748 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Temporary files should be created in /tmp directory

Problem:
DETAILED DESCRIPTION: Ld68k (and perhaps other tools) create temprorary
files
in the pwd.

1. If the program terminates abnormally (core dump) than the temporary f
ile
is left around. (A interrupt and some kill signals seem to clean up

KPR #: D200089748 **CONTINUED**

properly.

2. The program cannot be executed in a directory without write permissio
n
(where the output files are created elsewhere).

We would like as68k, ld68k, and ar68k to create ALL temporary files in
/tmp. Tmpfile(3S) and tmpname(3S) are available to expedite this proces
s
on HPUX.

Temporary solution:
None.

KPR #: D200089763 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Incremental link and strip results in corrupted relocatable

Problem:
When performing an incremental link (-i option) in
combination with the strip flag (-f nos), the resulting IEEE relocatable
file is corrupt. The corruption seems to be in the IMAGE part.

The fix may be tested in the incremental link directory by comparing
ieee relocatable files. Example files follow.

```
------------------------- eco242a.s -----------------------------
        sect    a
        xdef    ga
        xref    gb
ga      dc.l    gb
        end
------------------------- eco242b.s -----------------------------
        sect    a

        xdef    gb
        xref    ga
gb      dc.l    ga
        end
------------------------- eco242.lc -----------------------------
* strip & incremental link produce corrupt relocatable output
* incremental link (-i option) must be specified on command line
nlist s
load eco242a,eco242b
end
------- partial prnieee dump of resulting relocatable file -------
PRNIEEE: Printer version 3.0, reader version 3.0.0
         Reading file eco242.o
...
IMAGE LOAD PART:
---------------
(0142) SB : Current section for loading is L01.

       VVVVV !!!!!!!!!!!
```

KPR #: D200089763 **CONTINUED**

```
(0144) Record warning: (01) 0144 AS: extra fields found at end of record
.(0144
 ASP: Set load address for section L01 to (R01 + 0000)
(014D) ASR: R01 Base offset is (R01 + 0004)
(0154) SB : Current section for loading is L01.

       VVVVV !!!!!!!!!!!
(0156) Record warning: (01) 0156 AS: extra fields found at end of record
.(0156
 ASP: Set load address for section L01 to (R01 + 0000)
(015F) ASR: R01 Base offset is (R01 + 0004)
...
```

Temporary solution:
None.

KPR #: D200089771 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Reference to label in empty section causes ld68k error

Problem:
Detailed Listing for Defect Number LSDqf03193

ONE LINE DESCRIPTION: Ref. to label in empty sect. causes ld68k error.

DETAILED DESCRIPTION: Originally Microtec's eco #243. I am entering this so that I can keep track of it in the usual way.

An empty section is one which does not contain any code or data. It may, however, contain a label. If code in some other section refers to the label in the empty section, then ld68k generates an INTERNAL ERROR (318) when attempting to link the resulting relocatable file.

The problem is seen while linking while the fix, I am told, will be made in the assembler. The fix can be tested in either of two ways. By comparing the relocatable files produced by the assembler or by comparing linker listing files.

DEFECT OWNED BY: Paul Malek

Temporary solution:
There is no known workaround.

KPR #: D200089789 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Section mismatch causes bad info in HP link_sym file

Problem:
Section mismatch causes bad info in HP link_sym file

Ld68k flags a "Section Mismatch" when a global symbol
is defined in one section and referenced using a different section name.

KPR #: D200089789 **CONTINUED**

Under certain conditions when this warning occurs, there is bad information
in the HP link_sym file. Specifically, there is a bad "memory space" record for the referencing module.

This bad memory space record has a range from 0x00000000 thru 0xffffffff.
This causes problems for the HP emulators because they think the whole memory belongs to this module.

The only way to explain this is with several examples...

Temporary solution:
None.

KPR #: D200092312 Product: 68020 ASSEMB 300 64870S004 01.00

Keywords: MACROS

One-line description:
>37 parameters in a MACRO heading and it silently does not expand.

Problem:
If more than 37 parameters are declared in the MACRO heading, it insidiously declines to expand without generating any warnings.

Temporary solution:
None.

KPR #: 5000409094 Product: 68020 ASSEMBLER 64870-90901 01.00

One-line description:
WARNING: (335)

KPR #: D200086801 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
"at_execution run" may fail to run upon execution,

Problem:
"at_execution run ..." may ignore a subsequent execute command and fail to initiate a run.

Please contact the factory if you encounter this problem.

Joanne Carlson (719) 590-5840
-or- 590-5576

KPR #: D200091306 Product: 68020 EMUL 300 64410S004 02.00

One-line description:
Leading comma in some addtess indirect assembly is not needed

Problem:
The leading comma in some indirect address disassembly is not needed.

Example code listing:

```
*** ORIGINAL SOURCE FILE ***
     CHIP 68020
     ORG 0000H

     CLR.B          ([A0],$12345678)
     CLR.B          ([A0.W],$12345678)

*** ASSEMBLY LISTING ***
1                                       CHIP 68020
2                                       ORG 0000H
3
4    00000000 4230 8193 1234            CLR.B ([A0],$12345678)
                5678
5    00000008 4230 8193 1234            CLR.B ([A0.W],$12345678)
                5678

*** DELTA 68020 INVERSE ASSEMBLY ***
         0  42308193+  CLR.B            ([,A0.W],$12345678)
         8  42308193+  CLR.B            ([,A0.W],$12345678)
   ^
```

An unnecessary comma is displayed when no offset is present.

Temporary solution:
There is no workaround available.

KPR #: D200095778 Product: 68020 EMUL 300 64410S004 02.10

One-line description:
Emulation core dumps when run in a small window.

Problem:

KPR #: D200095778 **CONTINUED**

When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200096065 Product: 68020 EMUL 300 64410S004 02.10

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200088427 Product: 68020 EMUL 300 64416S004 02.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: 1650020396 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
LR error flagged for legal expression of the form 'label-value'.

Problem:
The 6809 assembler flags a legal range error if you use an external label in the following manner.

"6809"

```
      EXT     LSD
      JMP     LSD-2
```

The JMP LSD-2 causes a legal range error to be generated.

Temporary solution:
Jump to an equivalent positive offset. In this example you would use

"6809"

```
        EXT   LSD
        JMP   LSD+0FFFEH
```

Duplicate Service Requests: 1650044578

KPR #: 1650071050 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
LEAX [WORD] fails.

Problem:
The whole family of LEAx with x = X,Y,U,S has the problem
Example :

"6809"

```
            ORG       0D400H
LABEL1      LEAX      [WORD]
            LDA       [WORD]
            ORG       0D500H
WORD        FDB       0AABBH
```

The LDA instruction is compiled correctly to A6 9F D5 00
For the LEAX instruction an IO Error is flagged, though the processor supports extended indirect for the LEA family of instructions.

Temporary solution:
None.

KPR #: 5000143628 Product: 6805/9 ASSEMB 64844 01.10

One-line description:
Label in IF stmnt. does not appear in XREF

Problem:
The following shows a condition where a label is not listed in the cross reference table that should be there.

"processor name"

```
TOTO            EQU         0
                IF          TOTO
LABEL0          LDA         1000H
                ELSE
LABEL1          LDA         2000H
                ENDIF
```

When TOTO equals 1, everything is correct. The cross reference list both TOTO and LABEL0. When TOTO equals 0, the cross reference only list TOTO, not LABEL1.

KPR #: 5000150292 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
HEX pseudo causes byte counter to quit incrementing in certain cases.

Problem:
The byte counter is not incremeted after the 1EH in the following program.

"6809"

```
LABEL           HEX             1B,EC,1E,20,30
```

Temporary solution:
Use the FCB pseudo instead.

"6809"

```
LABEL       FCB     1BH,0ECH,1EH,20H,30H
```

KPR #: 5000164012 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
Arithmetic expression is not being evaluated correctly.

Problem:
When you offset a relocatable label by -1 the assembler flags an out of range error.

"6809"

```
        EXT     TABLE,ENDTABLE

        LDA     TABLE
```

KPR #: 5000164012 **CONTINUED**

```
        LDA     ENDTABLE-1                      ;LR error flagged
        LDA     TABLE+0FFFFH                    ;No error flagged
```

KPR #: 5000294207 Product: 6805/9 ASSEMB 64844 01.40

Keywords: CODE GENERATOR PROBLEM ON 9000/S300

One-line description:
BRSET range not checked.

Problem:

```
                                1"6805"
                    <0001>      2RSW            EQU         1
                                4               ORG         0FFH
     00FF                       5FLSW0          RMB         1
     0100                       6FLSW1          RMB         1
                                7               ORG         250H
     0250 02FF 05               8               BRSET       RSW,FLSW0,C1011
(1)->0253 0200 02               9               BRSET       RSW,FLSW1,C1011
     0256 A6 01                10               LDA         #1
     0258 9D                   11               NOP
  Errors= 0
```

Assembler should output error on (1).

Temporary solution:
None.

KPR #: D200063164 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
NT operator not operating consistiently.

Problem:
In the following program the .NT. operator will work in the first case, but, not in the second.

"6809"

```
START   EQU     80H
        LDA     #.NT.00001111B
        LDA     #.NT.11110000B          ;LR ERROR FLAGGED
        LDA     #.NT.START              ;LR error flagged.
```

Temporary solution:
AND THE VALUE WITH 0FFH.

"6809"

```
       LDA        #.NT.11110000B.AN.0FFH
```

KPR #: D200076950 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
BEXT address is not calculated correclty.

Problem:
In the following program the base page external DISP_MASK is given two different values in the two STA commands.

"6809"

```
        BEXT    DISP_MASK
        EXT     CLOCK,PROC

N_CALC  EQU     CLOCK+2
S_CALC  EQU     CLOCK_+1

        PROG
        SETDP   0

CALC:   CLR     S_CALC
        CLS     N_CALC
        STA     DISP_MASK
        JSR     PROC
        STA     DISP_MASK
        RTS


"6809"

        GLB     CLOCK,DISP_MASK,PROC

        DATA
        BASE_SEG
CLOCK   RMB     10
DISP_MASK       RMB     1
        BASE_END

        PROG
PROC:  NOP
       RTS
```

NOTE: In the first module all lines are necessary. If you remove the JSR PROC, for example, the problem goes away.

If you look at the .X file created you will note that the STA DISP_MASK's instructions have different destinations.

Temporary solution:
In module one (ie the one that declares DISP_MASK as BEXT and has the STA commands) declare DISP_MASK as a regular external and turn on the DIRECT pseudo above the STA instructions.

"6809"

```
        EXT     DISP_MASK,CLOCK,PROC
```

KPR #: D200076950 **CONTINUED**

```
N_CALC          EQU     CLOCK+1
S_CALC          EQU     CLOCK+2

                PROG
                CLR     S_CALC
                CLR     N_CALC

                DIRECT
                STA     DISP_MASK
                EXTEND
                JSR     PRO
                DIRECT
                STA     DISP_MASK
                RTS
```


KPR #: D200068239 Product: 6809 C 64822 01.07

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

"C"
"processor"

```
struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200069864 Product: 6809 C 64822 01.07

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

"C"
"processor"

```
$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);              /* See comment below. */

#if ibis
```

KPR #: D200069864 **CONTINUED**

```
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function.
For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200073171 Product: 6809 C 64822 01.07

One-line description:
Use of address (&) stack vars on right side of conditional expression

Problem:
C 6809 defect with the address(&) function:

Comparisons using the address(&) function with local variables or parameters in $RECURSIVE ON$ may generate incorrect code.

The use of the &(variable) function on the right hand side of comparison expressions can cause incorrect code to be generated.

eg. The statement:

```
IF ( pointer <> &local_var ) ...
```

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of functions compiled with $RECURSIVE ON$(the default value for C functions).

No problem occurs for static variables, any external variables, or local variables and parameters of functions compiled with $FIXED_PARAMETERS ON$ and $RECURSIVE OFF$.

The simple work around solution is to only use the &(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

The following listing illustrates the problem.

```
  "C"
  "6809"
  extern int a,b,c;
  Recursive_function(p,q,r)
```


KPR #: D200073171 **CONTINUED**

```
   int p,q,r;
  {  int i,j,k; $LIST_CODE ON$
                /* Static variables */
   if (&a !=  b) ; /* Works */
               LDX  #a
               CMPX b
               LBEQ Recursi01_1
          Recursi01_1
   if ( a != &b) ; /* Works */
               LDX  a
               CMPX #b
               LBEQ Recursi01_2
          Recursi01_2
   if (&a != &b) ; /* Works */
               LDX  #a
               CMPX #b
               LBEQ Recursi01_3
          Recursi01_3
                /* Local parameters */
   if (&p !=  q) ; /* Works with & on left side */
               LEAX 00000000CH,S

               CMPX 00000000EH,S
               LBEQ Recursi01_4
          Recursi01_4
   if ( p != &q) ; /* Fails with & on right side */
               LDX  00000000CH,S
               CMPX 00000000EH,S *** Value of q, NOT address of q ***
               LBEQ Recursi01_5
          Recursi01_5
   if (&p != &q) ; /* Fails with & on both sides */
               LEAX 00000000CH,S
               CMPX 00000000EH,S *** Value of q, NOT address of q ***
               LBEQ Recursi01_6
          Recursi01_6
                /* Local variables */
   if (&i != &j) ; /* Fails with & on both sides */
               LEAX 000000002H,S
               CMPX 000000004H,S *** Value of j, NOT address of j ***
               LBEQ Recursi01_9
          Recursi01_9
  }
```

Workaround:

The simple work around solution is to only use the &(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

KPR #: D200073171 **CONTINUED**

Temporary solution:
See problem text.

Duplicate Service Requests: D200073163

KPR #: D200075036 Product: 6809 C 64822 01.08

One-line description:
Some C programs using pointer & structure dereferences cause error #1006

Problem:
Some C programs using structure with pointers in expressions may cause pass 2 error 1006 - Compiler Error.

Some expressions with multiple use of pointer and structure dereferences may cause this error.

These errors did not appear on previous versions of the compiler.

The workaround solution is to break up the expression into smaller statements.

The following program illustrates the problem:

```
  "C"
  "6809"

  typedef unsigned int (*FP)();
  typedef struct hs {
      FP          load;
      FP          reset;
  } HEADER;

  int _app_set_valid(h)
  HEADER *h;
  {  FP p;  /* Temporary variable */
    h->load = (char*)h+(int)h->load;  /* This works */
    h->reset = (char*)h+(int)h->reset; /*  This causes 1006 error */
**** Pass 2 ERROR ?? 1006
  }
```

The following program shows some workaround solutions to the problem:

```
"C"
"6809"

typedef unsigned int (*FP)();
typedef struct hs {
    FP          load;
    FP          reset;
} HEADER;
```


KPR #: D200075036 **CONTINUED**

```
int _app_set_valid(h)
HEADER *h;
{  FP p;  /* Temporary variable */
  h->load = (char*)h+(int)h->load;  /* This works */
  /*h->reset = (char*)h+(int)h->reset;      This causes 1006 error */
    h->reset += (int)(char*)h;                /* This works: using += */
    p =  h->reset; /* This works: temporary assignment */
    h->reset =  (int)p+(int)(char*)h; /* This works: temprorary assignme
nt */
}
```

Temporary solution:
See problem text.

KPR #: D200075663 Product: 6809 C 64822 01.08

One-line description:
Programs with duplicate goto labels may fail in Pass 3 on VAX&HPUX C

Problem:
C programs with duplicate user labels(for goto's)may fail in pass3.

The current SUDS C compilers may produce the error
"comp: failed; too many errors in pass 3."
from some C programs which previously compiled correctly.

This problem did not appear in any C compilers before April 1987.

In C it is valid to use the same goto label symbol in different functions, since they have a logical different scope.

However, the HP64000 C cross will inform the user that these symbols are duplicate in the pass3 on the compiler. These symbols would produce duplicate label definitions when defined the ASM_FILE output is assembled. In addition the emulation products will only find one of these symbols.

The duplicate symbol detection algorithm on the HPUX/300, HPUX/500 and VAX/VMS C language compilers has an error which causes the compiler to fail.

However, the duplicate symbol checking is done after all of the relocatable and asmb_sym files have been produced. These output files are equivalent to those produced in the HP64000 version compilers. Thus, the output of the compilers is still correct, except for some trailing lines in the listing file.

The following program will cause this defect to occur:

```
"C"
"6800"
/*************************************************************/
```

KPR #: D200075663 **CONTINUED**

```
/*   TEST  file for problem with duplicate local labels     */
/*----------------------------------------------------------*/
/*   This program fails in pass 3 on VAX & HPUX/500 &/300   */
/*      While checking for duplicate asmb_sym symbols       */
/*      due to the "duplicate" error_exit labels            */

int i;
test1()
{
    /* ... */
  if (i == 77) goto error_exit;
    /* ... */
  error_exit:

       i = -1;
    /* ... */
}
/* duplicate symbol should be created */

test2()
{
    /* ... */
  if (i == 137) goto error_exit;
    /* ... */
  error_exit:

       i = -1;
    /* ... */
}
```

Temporary solution:
Do not use a local symbol more than once per module.

KPR #: D200079632 Product: 6809 C 64822 01.08

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;
```


KPR #: D200079632 **CONTINUED**

```
memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
       }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081547 Product: 6809 C 64822 01.08

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
          ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

Use `if( x != 0.0 ) ;`

OR

cast the variable to an int:

`if ( (int)x);`

KPR #: D200081547 **CONTINUED**

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200086611 Product: 6809 C 64822 01.80

One-line description:
Compare error using address of local variable on right of expression

Problem:
Problem: Compiler generates illegal instruction when performing an address compare of a stack relative local variable on the right hand side of an expression.

The compiler needs to use a load effective address instruction to create the proper address. This can not be done in one instruction with a compare.

```
"C"
"6809"
test(){
 int t,*q;
  q = &t;
  $AMNESIA$
  if (q != &t) ; /* This will not work */
/*WORKAROUND*/
  if (&t != q) ;
  }
```

EXPANDED EXAMPLE :

```
 test(){
  int t,*q;
   q = &t;
            LEAX 000000002H,S
            STX  000000004H,S
   $AMNESIA$
```


KPR #: D200086611 **CONTINUED**

```
   if (q != &t) ;
            LDX  000000004H,S
            CMPX 000000002H,S   /* This is not correct */
            LBEQ test01_1       /* Comparing to contents NOT address */
         test01_1
 /*WORKAROUND*/
   if (&t != q) ;
            LEAX 000000002H,S
            CMPX 000000004H,S
            LBEQ test01_2
         test01_2
   }
```

Temporary solution:

```
"C"
"6809"
test(){
 int t,*q;
  q = &t;
  $AMNESIA$
  if (q != &t) ; /* This will not work */


/*WORKAROUND*/
  if (&t != q) ;
  }
```

KPR #: D200086629 Product: 6809 C 64822 01.80

One-line description:
SHORT_ARITH OFF expressions in branches may not work as K&R

Problem:
Problem: With the SHORT_ARITH option OFF, the 6809 compiler does not execute full K&R C code correctly for certain mixed arithmetic operations when used in "if" expressions. Problems occur when 8-bit (short) arithmetic is used, rather than full expansion to 16 bit values to perform operations as in the standard K&R.

EXAMPLE:

```
"C"
"6809"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */
 /* WORKAROUND */
 if ((int)s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
  else  ;
}
```

KPR #: D200086629 **CONTINUED**

The 6809 C compiler computes mixed expressions correctly, as in assignment statements and parameter expressions.
This defect appears only when mixed expressions are used without assignment as conditional branching expressions.

This problem may be generated with other operators besides the "<<" as in the example, such as ">>" , "/" and "%

EXPANDED example:

```
  $SHORT_ARITH OFF$
  if (s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
             LDB  Dstatic
             LSLB
             LSLB
             LSLB
             LSLB
             LBEQ main01_1
             LBRA main01_2
        main01_1
   else  ;  /* Code branches here, due to use of byte arithmetic. */
        main01_2
/* WORKAROUND */
  if ((int)s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
             LDB  Dstatic
             SEX
             TFR  D,X
             LDB  #004H
             LBSR Zwshift
             CMPD #00000H
             LBEQ main01_3
             LBRA main01_4
        main01_3
   else  ;

        main01_4
```

Temporary solution:

```
"C"
"6809"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */


 /* WORKAROUND */
 if ((int)s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
  else  ;
}
```


KPR #: D200093575 Product: 6809 C 64822 01.80

One-line description:
Switch statement using unsigned int values 0 and 0xFFFF creates error

Problem:
Using a switch statement where the expression is an unsigned int, and the case values include small numbers and very large numbers, may create bad object code.

It appears that the compiler is attempting to decide whether to generate a jump table for the switching instead of individual case tests, which would be better for this example.

The code generated will not be able to jump to the "large", although apparently small (0xffff may look like -1) actual values.

For the following example, the generated code will not correctly jump to the case 0xffff statement.

```
main()
{
   unsigned I;
   I = 0xffff;
   switch( I )
   {
      case 0xfffd:
           break;
      case 0x0000:
           break;
      case 0x0001:
           break;
      case 0xffff: break;
      default :
           break;
   }
}
```

Temporary solution:
The workaround for this switch statement would be to test the cases with individual if statements.

```
   if (I == 0xfffd) ...
    elseif (I == 0xffff) ...
    elseif (I == 0) ...
    elseif (I == 1) ...
```

KPR #: 1650051649 Product: 6809 PASCAL 64813 01.30

One-line description:
If >39 functions declared; following funcs may include bad code.

Problem:
Define 39 procedures with PASCAL 6809, and define this procedure

```
PROCEDURE A40;
VAR CARAC_1:CHAR;
    BEGIN
        WRITE(CARAC_1);
    END;
```

and define another procedure A41 with the same instructions. If you look at the code generated by the compiler there is one more instruction with A40. This instruction is STD DA40+0394BH, between a SEX and a TFR D,Y.
You have the same problem with WRITE(REAL), WRITE(INTEGER), WRITE(UNS.), etc...
You also have the same problem with procedures #42,44,46,48 but not with procedures # 41,43,45,47,...

Temporary solution:
Limit each module of your project to under 40 functions.

s

KPR #: 5000184317 Product: 6809 PASCAL 64813 01.10

One-line description:
Records of pointers to text not handled correctly.

Problem:
The following program causes incorrect code to be generated for writes to TEXT files.

```
"6809"
$EXTENSIONS ON$
$RECURSIVE OFF$
$SEPARATE OFF$$
$GLOBPROC ON$

PROGRAM test;

TYPE
   files    = RECORD  F0,F1 : ^TEXT;
              END;

VAR     CH   :CHAR;
        f    :files;
        g    :TEXT;


PROCEDURE  doit (VAR f:files);

BEGIN
    WRITE(f.f0^,CH);      {LOOKS OK }
    WRITE(f.f1^,CH);      {LOOKS BAD.  NO CALL IS EVEN MADE TO
```


KPR #: 5000184317 **CONTINUED**

```
                          Pwrite_char. }

END;


BEGIN { MAIN }

WRITE(f.f1^,CH);
WRITE(g,CH);

END.
```

Temporary solution:
No temporary solution.

KPR #: 5000409821 Product: 6809 PASCAL 64813 01.60

One-line description:
Incorrect test generated for more than the 256th label.

Problem:

```
"6809"
PROGRAM TEST;

FLAG:BOOLEAN;
BEGIN
FLAG:=FALSE;
 WHILE FALSE DO BEGIN END;                repeat this line 127 times.

 WHILE NOT FLAG DO BEGIN END;
     the compiler generates the following code:
       DTEST00_255
       CLRA
       LBSR Zintneg
       CMPD #00000H
       LBEQ DDTEST00_256
instead of     CLR FLAG
               LDB FLAG
               EORB #000001H
which is correct.
```

If the number of labels is more than 512, the error 1004 occurs on the same instruction.

Temporary solution:
None.

KPR #: D200060020 Product: 6809 PASCAL 64813 01.09

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

KPR #: D200060020 **CONTINUED**

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200073155 Product: 6809 PASCAL 64813 01.10

One-line description:
ADDR function for stack relative variables in right side conditionals

Problem:
ADDR function for stack relative variables in right side of conditiona ls

Pascal 6809 defect with the ADDR function:

Comparisons using the ADDR() function with local variables or parameters with $RECURSIVE ON$ may generate incorrect code.

The use of the ADDR(variable) function on the right hand side of comparison expressions can cause incorrect code to be generated.

eg. The statement:

```
IF pointer <> ADDR(local_var) THEN ...
```

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of procedures

KPR #: D200073155 **CONTINUED**

compiled with $RECURSIVE ON$(the default value).

No problem occurs for static variables, which may be outer PROGRAM block variables, any external variables, or local variables and parameters of procedures compiled with $RECURSIVE OFF$.

The simple work around solution is to only use the ADDR(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

The following listing illustrates the problem.

```
  "PASCAL" PREPROCESS
  "6809"
  PROGRAM ADDRbug;
  $EXTENSIONS$
  $RECURSIVE ON$
    VAR GLBi, GLBj: ^INTEGER;
  PROCEDURE RecursiveON;
   VAR
      I,J:^INTEGER;
   BEGIN  {Procedure RecursiveON}
  $LIST_CODE ON$
     IF ADDR(I)<>     J  THEN ;  { With ADDR on left, This works. }
              LEAX 000000002H,S
              CMPX 000000004H,S
              LBEQ RecursiveON01_1
           RecursiveON01_1
     IF       I <>ADDR(J) THEN ;  { With ADDR on right, it FAILS ! }
              LDX  000000002H,S
              CMPX 000000004H,S  *** Value of J, NOT address of J ***
              LBEQ RecursiveON01_2
           RecursiveON01_2
     IF ADDR(I)<>ADDR(J) THEN ;  { With ADDR both sides ALWAYS FAILS !!
}
              LEAX 000000002H,S
              CMPX 000000004H,S  *** Value of J, NOT address of J ***

              LBEQ RecursiveON01_2
           RecursiveON01_3
  $LIST_CODE OFF$
   END;
  BEGIN
  $LIST_CODE ON$
     IF ADDR(GLBi)<>     GLBj  THEN ;  { With static vars it works. }
              LDX  #DADDRbug
              CMPX DADDRbug+00002H
              LBEQ ADDRbug00_4
           ADDRbug00_4
     IF       GLBi <>ADDR(GLBj) THEN ;  { With static vars it works. }
              LDX  DADDRbug
```

KPR #: D200073155 **CONTINUED**

```
             CMPX #DADDRbug+00002H
             LBEQ ADDRbug00_5
          ADDRbug00_5
     IF ADDR(GLBi)<>ADDR(GLBj) THEN ;  { With static vars it works. }
             LDX  #DADDRbug
             CMPX #DADDRbug+00002H
             LBEQ ADDRbug00_6
          ADDRbug00_6
  $LIST_CODE OFF$
  END.
```

Workaround:

Pascal 6809 defect with the ADDR function:

eg. The statement:

```
IF pointer <> ADDR(local_var) THEN ...
```

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of procedures compiled with $RECURSIVE ON$(the default value).

No problem occurs for static variables, which may be outer PROGRAM block variables, any external variables, or local variables and parameters of procedures compiled with $RECURSIVE OFF$.

The simple work around solution is to only use the ADDR(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

Temporary solution:
IF pointer <> ADDR(local_var) THEN ...

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of procedures compiled with $RECURSIVE ON$(the default).

No problem occurs for static vars, which may be outer PROGRAM block variables, any external variables, or local variables and parameters of procedures compiled with $RECURSIVE OFF$.

The simple work around solution is to only use the ADDR(local_var) function on the left hand side of the comparison expression.
If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will

KPR #: D200073155 **CONTINUED**

be required.

KPR #: D200075010 Product: 6809 PASCAL 64813 01.11

One-line description:
With statements used in FOR loops on records may cause error #1006

Problem:
With statements used in FOR loops may cause pass 2 error 1006 on VAX & HPUX 6809 Pascal compilers.

This problem does not occur on 64000 versions of the 6809 Pascal compilers.

The following program illustrates the problem:

```
"PASCAL"
"6809"
PROGRAM P1006;
$EXTENSIONS$
TYPE
    RECORDTYPE = RECORD
       FIELD1, FIELD2, FIELD3 : BYTE ;
    END ;
VAR
    VARTYPE : ARRAY [1..5] OF RECORDTYPE ;
    J,K,L,t1,t2   : BYTE ;

BEGIN   { MAIN }
     FOR J := 1 TO 5 DO
     BEGIN
        WITH VARTYPE[J] DO
          BEGIN
             FOR K := FIELD2 TO FIELD3 DO
**** Pass 2 ERROR ?? 1006
                L := L + 1 ;
         END;
    END;
END.    { MAIN }
```

Temporary solution:
The workaround is to assign the WITH variables to temporary variables for use in the FOR loop boundary conditions.

```
       WITH    VARTYPE[J]  DO
         BEGIN
            t1 := FIELD2;
            t2 := FIELD3;
            FOR K := t1 TO t2 DO
               L := L +1;
         END;
```

KPR #: D200075010 **CONTINUED**

Duplicate Service Requests: D200075002

KPR #: D200082446 Product: 6809 PASCAL 64813 01.11

One-line description:
Compiler incorrectly assumes the value of a var is in the D register.

Problem:
The compiler assumes it know the value of a variable which it has loaded in the D register, but, the D register is modified by a library call to Zwinset. The pascal code has the following logic:

```
   IF (VARIABLE in arrayOfRecords[].set)  THEN

       IF(array[VARIABLE] = someValue)
```

When the compiler tests the first condition VARIABLE is loaded into the D register and a call is made to Zwinset. Zwinset modifies the D registers. Next, when the compiler is testing the second condition it assumes VARIABLE is still in register D.

Temporary solution:
Anytime the 64000 compilers incorrectly assume a value is in a register try turning AMNESIA on around the offending statements.

```
$AMENESIA ON$

      IF (VARIABLE in arrayOfStructures[].set)

         IF (array[VARIABLE] = someValue)

$AMNESIA OFF$
```

KPR #: D200093468 Product: 6809 PASCAL 64813 01.70

One-line description:
Type casting the ADDR function to SET for masking may cause an error.

Problem:
Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
    Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

will geneate incorrect code.

The context of the pascal expression is clear that the AND operation

KPR #: D200093468 **CONTINUED**

is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

HERE is an expanded example:

```
"PASCAL"
"68000"
PROGRAM Error;
$EXTENSIONS$
TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;
VAR
    S : SET_OF_BITS;
    Byte1,Byte2: BYTE;
    I : SIGNED_16;
PROCEDURE BadADDRsetMASK;
 BEGIN
  Byte1:=BYTE(
  (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
            ^505
        LDD  #DPTEST110+00004H
        LDX  #000FFH
        LBSR Zuintmul              <----- Should be AND operation
        STB  DPTEST110+00002H
  END;
```

Temporary solution:
The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
    Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
    TempADDR := ADDR(variable);
    Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: D200093526 Product: 6809 PASCAL 64813 01.70

One-line description:
Large Sets may produce invalid results for elements outside set range

Problem:
The set inclusion operation may test undefined bit when the element being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

KPR #: D200093526 **CONTINUED**

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,        }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```

Temporary solution:
Detailed Listing for Defect Number LSDqf04487
Text:
Large sets may produce invalid results for elements outside set range

---

The set inclusion operation may test undefined bit when the element being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,        }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```

WORKAROUND:

The workaround for this defect is to separate the use of the the full 256 bit set implementation.

Instead of defining the large set as:

```
DIG = SET OF '0'..'9';
```

It could be rewritten:

```
digch = SET OF CHAR;
```

With the sets now using a full 256 bits, all bits will be set and tested properly.

KPR #: D200093526 **CONTINUED**

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,        }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```

KPR #: D200096735 Product: 6809 PASCAL 64813 01.70

One-line description:
Write statement causes ERROR 1006.

Problem:
The following program will produce a pass 2 ERROR #1006 . Note: the dots of course refer to External Procedure declarations P03-P37. When P40 is taken out of the comments the error does not occur. Also, when less than 39 procedures are declared the error does not occur.

```
"6809"
PROGRAM BAD;

PROCEDURE P01; EXTERNAL;
PROCEDURE P02; EXTERNAL;
.
.
.
.
.
.
.
PROCEDURE P38; EXTERNAL;
PROCEDURE P39; EXTERNAL;

(* PROCEDURE P40; EXTERNAL; *)

PROCEDURE BUG;
BEGIN
  WRITE('X');
END;

BEGIN
  REWRITE(OUTPUT);
END.
```

Temporary solution:
Need more than or less than 39 external procedures declared.

KPR #: 5000093708 Product: 6809 PASCAL M 64813-90903 00.02

One-line description:
Parameter passing thru the registers has changed.

Problem:
Further explanation on how we use registers for parameter passing is needed.

Temporary solution:
No temporary solution.

KPR #: D200094599 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Pressing <shift><home> in high level code window does not work.

Temporary solution:
There is no known work around at time.

KPR #: D200094870 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Section pragma causes unpredictable behavior.

Problem:
Using section pragma with compiler or assembler will cause unpredictable results.

Temporary solution:
Do not use section pragma.

KPR #: D200094888 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Illegal use of Expression Monitor Value corrupts array without warning

Problem:
Assume the following declaration:

```
main()
{
...
static char     array[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g'};
...
}
```

After stepping into main(), array[] has been initialized. At this point, the command "Expression Monitor Value array" displays the elements of array[n], and they are correct. Now, the expression *++array is syntactically incorrect, since "array" is a constant, not a variable. Entering the command "Expression Monitor Value *++array" generates no error message and corrupts the elements of "array".

Temporary solution:
Do not use Expression Monitor Value *++array.

KPR #: D200094896 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Floating pt values are rounded and/or displayed as integral values.

Problem:
If a float is displayed in the monitor window the following results are seen.

```
----------------
darray[1]=9999999
```


KPR #: D200094896 **CONTINUED**

```
darray[2]=10000000

Monitors as:
darray[1]       1E+07
darray[2]       1E+07


----------------
darray[1]=0.99999
darray[2]=0.999999

Monitors as:
darray[1]       0.99999
darray[2]       1

----------------
darray[1]=7999991

Monitors as:
darray[1]       7.99999E+06
----------------
darray[1]=7.999999

Monitors as:
darray[1]       8
```

Temporary solution:
No known work around at this time.

KPR #: D200094904 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Comment lines echoed to journal window with Command Echo OFF.

Problem:
Execution of command files with Debugger Option Command_Echo OFF still echos blank lines to journal file.

Temporary solution:
No known work around at this time.

KPR #: D200094938 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Debugger will not break on access of inport address.

Problem:
The following illustrates the problem:

```
----- test_port.c ----------------------------------------

#include <stdio.h>
/* Declare ports */
unsigned char  input;
unsigned char  output;
```

KPR #: D200094938 **CONTINUED**

```
int main()
{

  short i;

  /**
    Simple test - read from input port until bit 3 is set - then
    echo to output port.
  **/

  for (;;) {

    i = input;

    printf("Input value is: %2x\n",(unsigned)i);

    if (i & 0x8) {
      output = i;
    }
  }

  return(0);
}
```

After compiling, run the debugger and use the following command sequence:

- o Program Load Default test_port.x
- o Memory Inport Byte Address &input Source_Is File test_port.c
  (Same result with Source_is Journal, and Data_String "...")
- o Memory Outport Assign Byte &output Destination_Is Journal_Window
- o Breakpoint Access &input (also tried B Read &input)
- o Breakpoint Access &output (also tried B Write &output)
- o Program Run

Program reads first data item, prints data into stdio window, if item had bit 3 set, shows output in journal window, and then breaks on 'output' access (though code shows break at line following 'output = i'). If the input was to come from Journal

Window, it did ask for input.

The program does not break on the access of the input port.

It does break on the access of the output port.

Temporary solution:

KPR #: D200094938 **CONTINUED**

No known work around at this time.

KPR #: D200094961 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
No error when writing to a file opened for read.

Problem:
Open a user window as a readable file using:

File User_Fopen Read 51 File <filename_of_existing_file>

and then write to it using:

Expression Fprintf 51,"%d\n",10

The debugger does not issue an error message at this point. The data seems to be ignored. It is good that the debugger does not modify the file. Please add an error indication for this action.

Temporary solution:
No work around necessary.

KPR #: D200094979 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Macros can not have parameter names that have underscores

Problem:
The following macro declaration will not work:

Debugger Macro Add cnvrt(current_temp)

When using the command: File Command cnvrt(current_temp)

The debugger will come up with the error message:
Invalid argument, expected:)
while pointing at the 'D' in Debugger

Temporary solution:
Do not use underscores in macro parameter names.

KPR #: D200095117 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
If push Step_Over after trying Step_Into printf, pgm runs to completion

Problem:
If you try to step into a Printf function (or any function in the HP supplied libraries) and then hit 'Step Over' key, the program will run to completion, if no other breakpoints are set.

Temporary solution:
You will not be able to step into a routine supplied by HP (such as printf). The source libraries are not available. If this

KPR #: D200095117 **CONTINUED**

happens, set a breakpoint at the stack level above the HP library call, and then run.

KPR #: D200096172 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Too many memory map commands yields erroneous breakpoint error.

Problem:
DEFECT:

If use more than 30 different memory map commands, the wrong error message appears:

'Breakpoint limit exceeded.'

Temporary solution:
There is no known work around at this time.

KPR #: D200096180 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
"pi" register is not maintained across Save_state and Load_state.

Problem:
If you execute a "Debugger Execution Save_State" command, then at some later time use a "Debugger Execution Load_State" to restore the debugger to the previous state, the "pi" register is not restored.

Temporary solution:
There is no known work around at this time.

KPR #: D200096198 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
Unsetting TERM in the environment causes core dump.

Problem:
Unsetting TERM in the environment causes core dump

Temporary solution:
Do NOT unset the TERM environment variable.

KPR #: D200096610 Product: 680XX DEBUG/SIM 300 64360S004 00.00

One-line description:
These do not evalueate to the same result: "1+(&test)" and "(&test)+1"

Problem:
Using "Expression Display_Value", the following results are obtained. "increment" is a DS.L within an assembly file.

case	debugger command		debugger result	correct result	
a	Expression Display_Value &increment		103ch	103ch	-- ok
b	Expression Display_Value sizeof(incret)		4	4	-- ok
c	Expression Display_Value &increment+1		1040h	1040h	-- ok
d	Expression Display_Value &increment+2		1044h	1044h	-- ok
e	Expression Display_Value &increment+3		1048h	1048h	-- ok
f	Expression Display_Value 1+&increment	**	40f1h	1040h	-- WRONG
g	Expression Display_Value 2+&increment	**	40f2h	1044h	-- WRONG
h	Expression Display_Value 3+&increment	**	40f3h	1048h	-- WRONG

KPR #: D200096610 **CONTINUED**

Note that the results for cases c-e should be the same as the results for cases f-h. Use of parenthesis has no effect on the results (i.e. 1+&increment shows the same value as 1+(&increment)). These results are the same for breakpoint setting, and memory map specifications.

Another result is shown by the following cases:

case	debugger command		debugger result	correct result	
a	Expression Display_Value &increment		103ch	103ch	-- ok
b	Expression Display_Value sizeof(incret)		4	4	-- ok
c	Expression Display_Value &increment+1		1040h	1040h	-- ok
d	Expression Display_Value &increment+2		1044h	1044h	-- ok
e	Expression Display_Value &increment+3		1048h	1048h	-- ok
f	Expression Display_Value &increment-1	**	4155t	4152t	-- WRONG
g	Expression Display_Value &increment-2	**	4154t	4148t	-- WRONG
h	Expression Display_Value &increment-3	**	4153t	4144t	-- WRONG

Temporary solution:
WORK AROUND: Must use ptr+scalar.

Expressions of form ptr - scalar work incorrectly.

Use form ptr + (-scalar)

KPR #: 1650048355 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
68HC11 will work alone as a measurement system.

KPR #: D200082271 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083287 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086066 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

KPR #: D200086066 **CONTINUED**

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200086413 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088377 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090902 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

KPR #: D200090902 **CONTINUED**

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095752 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200096040 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200096727 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
Target system resets on "display memory" command.

Problem:
When the command line parser for the emulator configuration encounters a "16" as a parameter for the COP watchdog timer, it loads a "1" instead

Temporary solution:
Because the leading "1" in "16" is what is read, using any of the other values "4" or "64" will work OK.

KPR #: D200082313 Product: 70108 EMUL 300 64295S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200086108 Product: 70108 EMUL 300 64295S004 01.10

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200088393 Product: 70108 EMUL 300 64295S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.

KPR #: D200088393 **CONTINUED**

This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200082305 Product: 70116 EMUL 300 64294S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200086090 Product: 70116 EMUL 300 64294S004 01.10

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                                  ; end locks the emulation session
3. <system name> <module name>          ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200088385 Product: 70116 EMUL 300 64294S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.

KPR #: D200088385 **CONTINUED**

This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200082339 Product: 70208 EMUL 64297S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200088419 Product: 70208 EMUL 64297S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: 5000242818 Product: 70216 EMUL 64296 01.00

One-line description:
V50 Disassembler generates "illegal" opcode for "POP PS" instruction

---

KPR #: 5000251363 Product: 70216 EMUL 64296 01.00

One-line description:
Can not specify needed trigger specification.

KPR #: D200082321 Product: 70216 EMUL 64296S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200088401 Product: 70216 EMUL 64296S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is. In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200089847 Product: 80186 EMUL FW M 64764-90901 01.00

One-line description:
The Manual says that step is not allowed in real time mode.

Signed off 01/26/89 in release X00.00

KPR #: D200090167 Product: 80186 EMUL FW M 64764-90901 01.00

One-line description:
The "stty" command doesn't work correctly for baud rate <= 1200.

Problem:
If you toggle the xon parameter when running at 1200 baud and below, the stty command will return invalid characters.

```
>stty
 stty A 1200 xon
>stty -xon
 [#!,*&^junk characters
```

Since the PC interface calls the stty command upon startup, this problem will make the PC interface fail at startup with a datacomm error at 1200 baud (all lower baud rates are not supported by the PC interface).

Temporary solution:
To get around this problem, just set switch 13 on the emulator's back panel (enable xon). The stty parameter will not be toggled and PC interface will startup successfully.

From the terminal-mode interface, just enter another carriage-return to regain proper communications.

Signed off 01/26/89 in release X00.00

KPR #: 1650044016 Product: 80186 EMULATION 64224 00.00

One-line description:
"run from <addr>", "modify reg <reloc>" generates 16 extra I/O writes.

Problem:
The commands "Run From <addr>", "modify register <name>", where <name> is a relocatable register such as UMCS, LMCS, generates a series of 16 I/O writes on addresses from 0 to 1EH incrementing by 2 (0,2,4,6,8....) from the start of the Perpheral Control Block.

KPR #: 5000211557 Product: 80186 EMULATION 64224 01.04

One-line description:
"disp. memory mnemonic" shows incorrect inv. assembly for JMP NEAR inst

Problem:
"display memory mnemonic" shows incorrect inverse assembly for JMP NEAR instruction when MOV for segment register appears before the JMP NEAR.

```
 EXAMPLE source code:  MOV DS,BX
                       JMP NEAR PTR 1000H

   disassembled code:  MOV DS,BX    JMP 0FFFH
```

Temporary solution:
There is no workaround available.

KPR #: 5000225748 Product: 80186 EMULATION 64224 01.05

One-line description:
LODS instructions with segment override not properly disassembled.

Problem:
LODS instructions with segment override not properly disassembled. Example:

The code

```
          2EH,0ACH  represents a    LODSB  CS:[SI] instruction
```

when displayed mnemonic the opcode is shown as:

```
          LODS   ES:BYTE PTR[DI],CS:[SI]
```

other LODS instructions with segment override disassemble incorrectly also.

Temporary solution:
There is no workaround available.

KPR #: D200095463 Product: 80186 EMULATION 300 64224S004 01.20

One-line description:
Open of file pvxxxxfile_asmb causes pv failure on long file name systems

Problem:
When running performance verification (pv), one of the following two files are used: (1) pv0032file_asm, or (2) pv0033file_asm. If a short file name system is being used, then these two files are considered the same, and pv will run correctly. However, if a long file name system is being used, these two files are NOT considered to be the same, and PV will NOT run correctly.

This applies to the 64224S004 and 64225S004 software.

Duplicate Service Requests: D200095455 D200095471

KPR #: D200065805 Product: 80188 EMULATION 64225 01.03

Keywords: USER MEMORY

One-line description:
Emulator would not recover from errors during display memory repetitive.

Problem:
The problem occurs when displaying user memory repetitively. An error condition such as slow clock or guarded memory access would cause the 64000 station to reboot or to display extraneous data at the top of the screen. When the screen had been written to at the top, the only action to delete the characters was resetting the station.

Temporary solution:
There is no workaround other than avoiding the error conditions during a repetitive display of user memory.

KPR #: 5000240259 Product: 80286 EMULATION 64228 01.02

One-line description:
trace only <Odd Address> data 0: analyzer doesn't qualify properly.

Problem:
Trigger and store function in 80286 emulator does not function properly when an odd address is specified with a data qualifier. The problem is that the data quailifier appears to be ignored. For example:
"trace only address 1463DH data 0"
will show ALL accesses to 1463DH, not just those with data = 0H

also:
"trace only address 1463Dh data 03EXXH"
will show all accesses to 1463DH

also:
"trace only address 1463DH data 0XX3EH"
will not capture any data, however
"trace only address 1463DH data 0XX00H" will capture all accesses
(The program used to test this writes consectutive values from 0-FFH to location 1463DH)

KPR #: 5000244343 Product: 80286 EMULATION 64228 01.02

One-line description:
80286 emul. fails to run programs mapped as user memory at the target.

Problem:
The 80286 emulator fails to run programs mapped as user memory at the target system if the target system makes frequent hold requests. This problem results from a cpu misinterpretation of an ACK signal. The emulator generates a HOLD-ACK signal at the end of every emulator controller's hold cycle. If the target system makes a hold request near the HOLD-ACK signal, the target system may misuse this ACK as its own ACK, and immediately initiates BUS master operation. But the 80286 cpu continues to execute normal bus cycles. The CPU's read/write instruction fails because the data/address bus is used by another bus master.

Temporary solution:
There is no workaround available at this time.

KPR #: 5000273250 Product: 80286 EMULATION 64228 01.02

One-line description:
80286 Emulator may not display proper Interrupt Type number.

Temporary solution:
There is no workaround available at this time.

KPR #: 5000273268 Product: 80286 EMULATION 64228 01.02

One-line description:
trace abt addr 0:0E0H status rd mem triggers on addresses 0E0h, 0C0H.

Problem:

KPR #: 5000273268 **CONTINUED**

Special combination of address and status causes improper analyzer spec in 80286 emulator.

The trace command:

"trace about address 0:0E0H status read_mem" will trigger on addresses 0E0H and 0C0H. It should only trigger on 0E0H.

The commands:

"trace about address 0:0E0H" AND
"trace about address 0E0H status read_mem"

work properly.

KPR #: 5000275727 Product: 80286 EMULATION 64228 01.02

One-line description:
"list printer memory" command gives wrong addresses using seg:offset.

Problem:
The "list printer memory" command does not function properly. Two errors have been observed.

1. If the "list printer memory" command is issued with the address specified in the segment:offset format the addresses on the printout do not have a colon separating the segment from the offset. The resulting addresses are not the addresses specified.

2. If the "list printer memory" command is issued multiple times after a "display memory" command (again with the address specified as segment:offset) the addresses in the printout are not the addresses specified.

KPR #: D200080127 Product: 80286 EMULATION 64228 01.02

One-line description:
First PV cycle shows failure with some 64155B cards, if PV'd 1st on 228.

Problem:
When the 64228 is in the same cage as a 64155B, and PV is run on the 228 before being run on the 155B, the first cycle of PV will fail with certain 155B cards, and not with others. PV passes when perfomed on the 155B card before the 228.

Temporary solution:
This problem does not influence the operation of the emulator, and is merely an inconvenience, especially however, for those who do not expect it to occur.

KPR #: 5000141747 Product: 80286 UDE 64227 01.00

Keywords: DISPLAY MEMORY

One-line description:
Inverse assembler does not work properly during display memory mnemonic.

Problem:
The "MUL " instruction is not correct when shown by a "display memory mnemonic" command. The inverse assembler cosiders it a 3 byte instruction rather than a 2 byte instruction. For example

```
3015    MOV BX,#0010H
3018    MUL BX
301B    MOV 3002H,AX        * the correct address is 301A
301E    JMP s 03004H        * this address is correct
```

The trace disassembly is correct.

KPR #: 5000162651 Product: 80286 UDE 64227 01.00

Keywords: DISPLAY MEMORY

One-line description:
The IDIV instruction is not correct during a display memory mnemonic.

Problem:
The trace disassembly is correct. The problem is only with display memory mnemonic. An example follows.

```
3009    MOV BX,#0100H
300C    IDIV BX
300F    MOV 3000H,AX     * address should be 300E
3012    MOV ax,#0100H    * this address is correct
```

KPR #: 5000181131 Product: 80286 UDE 64227 01.00

Keywords: DISASSEMBLER

One-line description:
Incorrect data is returned on a trace about an I/O port.

KPR #: D200046714 Product: 80286 UDE 64227 01.00

Keywords: INSTRUCT. EXECUTION

One-line description:
Single step function does not work after a software breakpoint.

KPR #: D200068775 Product: 80286B ASSEMB 64859 01.02

One-line description:
Aliases not allowed in the linker to specify library paths.

Problem:
Path specifications are not allowed for libraries in the linker on old assembler, you could use aliases; you can not use aliases now.

Temporary solution:
No known temporary solutions.

KPR #: D200092734 Product: 80286B ASSEMB 64859 01.40

One-line description:
MODULE pseudo generates random relocation type

Problem:
MODULE pseudo-op in 80286 extensions does not set the TYPE variabl to a known value prior to generating code and so may yield different checksums when running through regression tests.

This results in extra work when running tests because the checksums may not be the same as previously. THIS DOES NOT AFFECT THE RESULTANT EXECUTABLE CODE. This will be modified prior to the next release to make running regression tests easier.

KPR #: 5000132662 Product: 8048 ASSEMB 64846 01.00

One-line description:
Error message LR generated on valid JMP instruction

Problem:
The 8042 processor allows jumping through 2K blocks called pages
The following example generates a LR error for a valid JMP opcode.

```
"8042"
      ORG    401H
LABEL   NOP
        NOP
        PROG
        JMP     LABEL       {opcode 8401 - is valid, p.14-19
                ^LR error       Micontroller Handbook}
```

Temporary solution:
No known temporary solution.

KPR #: 5000169995 Product: 8051 ASSEMB 64855 01.08

One-line description:
Assembler inconsistant in permitting forward referencing

Problem:
The assembler does not always allow forward referencing. It is
not clear why it allows forward referencing sometimes but not others.

"8051"

```
     MOV SYMBOL,C  ; no error
     MOV C,SYMBOL  ; DE error why?
     EXT SYMBOL

     ORL A,SYMBOL2 ; no error
     EXT SYMBOL2
     END
```

Temporary solution:
Define all externals before referencing them.
(In this case customer does not like this workaround, because a large amount of code was written under rev 1.06 and these errors did not occur.)

KPR #: 5000171470 Product: 8051 ASSEMB 64855 01.08

One-line description:
Defining a transfer address causes an ET error

Problem:
The following program generates an ET error.

"8051"

```
start   NOP
        NOP
        END start
```

Therefore, a transfer address cannot be defined. This is a critical need for our customers for emulation

Temporary solution:
No known temporary solution.

KPR #: 5000240929 Product: 8051 ASSEMB 64855 01.20

Keywords: CODE GENERATOR PROBLEM ON 9000/S300

One-line description:
Special operator "HIGH" does not work with DS pesudo opcode

Problem:
The special operator "HIGH" does not work correctly when the label is defined using the DS pseudo opcode: Example:

KPR #: 5000240929 **CONTINUED**

```
"8051"
        ORG   1234H
LABEL1  EQU   $
LABEL2  EQU   3344H
LABEL3  DS    1
        MOV   A,#HIGH(LABEL1)  ;correct - moves 12H into A
        MOV   A,#HIGH(LABEL2)  ;correct - moves 33H into A
        MOV   A,#HIGH(LABEL3)  ;WRONG - moves 34H into A
        MOV   DPTR,#LABEL3     ; correct
        END
```

Temporary solution:
There is no known work around at this time.

KPR #: D200068379 Product: 8051 ASSEMB 64855 01.08

One-line description:
HIGH operator does not function correctly

Problem:
The HIGH operator does not work when accessing data memory. It assumes that external ram is 8 bits instead of the 16-bits. So no matter what you use as an example, you always get the LOW byte.

Temporary solution:
No known temporary solution.

KPR #: D200081570 Product: 8051 ASSEMB 64855 01.08

Keywords: CODE GENERATOR

One-line description:
HIGH does not work

Problem:
HIGH operator does not work

Temporary solution:
There is no known work around.

KPR #: D200091710 Product: 8051 ASSEMB 64855 01.08

One-line description:
CONT in linker will overwrite addresses of variables in different module

Problem:
The CONT command used during link will overwrite variable addresses declared in different modules.

---------- MOD1 ----------

```
"8051"
        EXT LAB1,LAB2

        PROG
```


KPR #: D200091710 **CONTINUED**

```
        MOV     LAB1,#01H
        MOV     LAB2,#02H
```

---------- MOD2 ----------

```
"8051"
        GLB LAB1,LAB2,LAB3

LAB1    DS      1
LAB2    DS      1
LAB3    DS      1
```

---------- MOD3 ----------

```
"8051"
        GLB LAB4,LAB5
LAB4    DS      1
LAB5    DS      1
```

After assembling, link all three modules together using the CONT command for the addresses of the last two modules. The XREF listing will look like this:

```
          LAB1       D   0020  <-----
          LAB2       D   0021
          LAB3       D   0022
          LAB4       D   0020  <----- Should continue, not start
          LAB5       D   0021         over
```

Temporary solution:
Define all variables in one module, or declare the addresses during link and don't use CONT.

KPR #: D200092098 Product: 8051 ASSEMB 64855 01.08

One-line description:
Cross reference goes into endless loop on macro reference.

KPR #: 5000206458 Product: 8051 ASSM M 64855-90902 01.05

Keywords: CODE GENERATOR

One-line description:
In the manula pg 8-2 states the BIT instruc. shows operand is address.

KPR #: D200086439 Product: 8051 ASSM M 64855-90902 01.07

Keywords: MANUAL

One-line description:
The assmblr manual needs to be updated w/ information in reference manul

---

KPR #: 5000183475 Product: 8051 EMUL M 64264-90901 01.01

One-line description:
Manual enhancement to reflect Port display info in more detail.

KPR #: 1650042655 Product: 8051 EMULATION 64264 00.00

One-line description:
Cannot load absolute file using remote file access.

Problem:
Cannot load absolute file using remote file access(RFA).

KPR #: 5000285536 Product: 8051 EMULATION 64264 01.04

One-line description:
MODIFY EXTERNAL MEMORY WITH ODD INITIAL ADDRESS DOES NOT WORK CORRECTLY

Problem:
Configuration : External data memory mapped as emulation memory.

Problem : "modify external data memory ADDR1 thru ADDR2 to 0"
command does not modify some values to 0 if ADDR1 is
an odd address.

Example : modify external data memory 0A001H thru 0A7FFH to 0

address A2EE is 4, A3E7 is 4, A3E8 is B, A4E0 is 4, A4E1 is B

If command is reexecuted from address 0A003H (offset of 2),
the addresses with incorrect values are also offset by 2.

KPR #: D200095745 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before
the status line comes up and generates a core file. It should
leave the emulator locked and display a message, "Display size
is too small".

KPR #: D200096032 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file
the command fails silently. The error message "permission denied" never
shows up.

KPR #: 5000219220 Product: 8080/5 ASSEMB 64840 01.00

Keywords: CODE GENERATOR

One-line description:
xref incorrect with conditional assmbly IF when code generated for false

Problem:
Cross reference listing is incorrect when using IF statement of conditional assembly. When code is generated for FALSE part of IF, the cross reference may be for the wrong value.

Temporary solution:
No known solution at this time.

KPR #: 5000129023 Product: 8085 B PASCAL 64825 01.01

Keywords: PASS 1

One-line description:
$Range ON$ causes incorrect code to be generated for a test operation.

Problem:
The following program when compiled with the $RANGE ON$ option wil cause incorrect code to be generated.

```
"B8085" | "BZ80"
$EXTENSIONS$
$RANGE ON$

PROGRAM  BOOLREAL;

VAR  A,B,C    :   REAL;
     L        :   BOOLEAN;

BEGIN
     A := 10.0;
     B := 15.0;
     C := 12.0;


     L := (C < (B+.5)) AND ((C + .5) > A);
END.
```

The two intermediate results "(C < (B +.5))" and "((C+.5) >A)" are anded together and this result is compared with the value two. Thus the case is never true. With RANGE OFF correct code is generated.

Temporary solution:
It is necessary to turn $RANGE OFF$ to obtain correct code. Simply breaking up the expression will not work.

KPR #: D200060228 Product: 8085 B PASCAL 64825 01.02

One-line description:
Incorrect data offsets in listing file.

Problem:
I am expanding this to all pascal compilers. The C compilers list the correct offset. $FAR ON$ only applies to the 68000 cross compiler. The other compilers exhibit the defect w/o any options on.

```
"processor name"
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
```

KPR #: D200060228 **CONTINUED**

```
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200093641 Product: 8085 B PASCAL 64825 01.90

One-line description:
Type casting the ADDR function to SET causes error #1006 on the VAX

Problem:
Type casting the ADDR function to type SET causes 1006 error on VAX.

Temporary solution:
Break up the expression by isolating ADDR:

```
TempADDR := ADDR(variable);
Byte := BYTE(SET_OF_BITS(TempADDR)*SET_MASK);
```

KPR #: D200069948 Product: 8085 C 64826 01.03

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
   i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200081562 Product: 8085 C 64826 01.04

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
          ;
}
```

KPR #: D200081562 **CONTINUED**

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

Use if(x != 0.0) ;

OR

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```


KPR #: 5000398396 Product: 8085 EMULATION 64203 01.07

One-line description:
64203A (8085) MEMORY MAPPING PROBLEMS

Duplicate Service Requests: 5000404988 5000298398

KPR #: D200095521 Product: 8085 EMULATION 300 64203S004 01.40

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095810 Product: 8085 EMULATION 300 64203S004 01.40

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200013334 Product: 8085 PASCAL 64810 00.70

Keywords: CODE GENERATOR

One-line description:
Compiler generates incorrect code for BOOLEAN assignment statement.

Problem:
The following program displays a code generation error with regard to BOOLEAN assignment statements. The BOOLEAN assignment statement overwrites a value in the H&L register pair which is relied upon later.

```
PROGRAM BAD_LOOP;

   VAR
      A, B : BYTE;
      ERROR : BOOLEAN;

   BEGIN
      REPEAT
      UNTIL A < B;
      ERROR := FALSE;      { overwrites H&L which contain 'A' }
      FOR A := 1 TO B DO   { uses H&L assuming 'A' still in register }
   END.
```

Temporary solution:
No known workaround at this time.

KPR #: D200081240 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:
Display Memory and Modify Memory will be incorrect at the segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the center of a line.

Modify Memory will be incorrect if done beyond the end of a segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment in the middle of the line being displayed.

Modify Memory will be correct if it is not extended through the end of a segment. For example:

modify memory 0FFFEH to 1,2 will be correct.

modify memory 0FFFEH to 1,2,3 will NOT be correct, because the third entry is in the next segment.

KPR #: D200081414 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

KPR #: D200082123 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed

KPR #: D200082123 **CONTINUED**

some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083139 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200084921 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

KPR #: D200085944 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step

KPR #: D200085944 **CONTINUED**

number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200090753 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095604 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095893 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: 5000254730 Product: 8086-89 ASSM M 64853-90907 02.01

Keywords: MANUAL

One-line description:
8086 Asm/linker manual doesn't doc. valid DQ and DT directives.

Problem:
The 8051 assembler/linker reference manual has paragraph with errors relating to the CSEG directive. The errors are typographical, but can lead to confusion:
Page 8-4 of the HP-UX hosted manual talks about the CSEG directive with a paragraph:

The code segment counters can be charged (changed) with the DS, DW, and DW (DB) pseudos, and with each instruction encoded. Each unit in the program relocatable counter represents one byte in the code address space within the range of 0 to 64. (0 to 64K)

(corrections are indicated in parenthesis)

KPR #: 5000247783 Product: 8086/8 ASSEMB 64853 02.30

Keywords: CODE GENERATOR PROBLEM ON 9000/S300

One-line description:
Assembler does not handle all string comparisons correctly.

Problem:
Assembler does not handle all string comparisons correctly:
"70108"

```
DEM01   MACRO   &PARM1
VAL1    SET     1
        .IF     "&PARM1" .EQ. ":char" saw_char
VAL2    SET     2
        .IF     "&PARM1" .EQ. ":long" saw_long
VAL3    SET     3
        .IF     "&PARM1" .EQ. ":int" saw_int
VAL4    SET     4
        .IF     "&PARM1" .EQ. ":ch" saw_ch
VAL5    SET     5
saw_char  .NOP
saw_long  .NOP
saw_int   .NOP
saw_ch    .NOP
        MEND
        DEM01    char  ;shouldn't match ":char", but does
     DEM01  :char   ;should match and does
     DEM01  long    ;shouldn't match, but does
     DEM01  :long   ; should match and does
     DEM01  int     ; shouldn't match - and doesn't INTERESTING
     DEM01  :int    ;should match and does
     DEM01  ch      ;shouldn't match - and  doesn't
     DEM01  :ch     ; should match and does
     DEM01  nothing ; shouldn't match and doesn't
     END
```

It appears that if the original macro string includes a colon, and the passed string has more than 3 chars that match the first 3 chars after the colon, the comparison will always be true.

Temporary solution:
Pass strings that are less that 4 characters.

KPR #: 5000283077 Product: 8086/8 ASSEMB 64853 02.70

Keywords: PROBLEM ON 9000/S300

One-line description:
CMP statement is producing wrong label address.

Problem:
The assembler generates a bad code, which counts a wrong address.

```
          "8086"
                PROG CS:PROG,DS:DATA
                GLOBAL POWER_ON
          POWER_ON
E80100          CALL   BBB
```

KPR #: 5000283077 **CONTINUED**

```
80BC000000      CMP    DS:AAA[SI],0
90        BBB   PROC
33S0            XOR    AX,AX
C3              RET
                DATA
0000      AAA   DBS    1
                END
```

In above program, "CALL BBB" generates a wrong code "E80100". This is a mistake of counting its address. ASM 8086 rev2.3 generates PH error, but we can not find where the wrong part is.

NOTE: For the CMP statement the assembler generates 1 byte during pass 1, and 7 bytes during pass 2.

Temporary solution:
Replace "CMP DS:AAA[SI],0" with "CMP DS:BYTE PTR AAA[SI],0" will achieve the desired results however forward referencing is still not recommended.

KPR #: D200096594 Product: 8086/8 ASSEMB 64853 02.80

One-line description:
EXT with the 70108 causes US error.

Problem:
The following code duplicates the problem.

```
"70108"
   EXT   TEST
         ^
         US - Undefined symbol
```

NOTE: Using EXTERNAL works fine. And, EXT works fine with the 8086.

Temporary solution:
Use EXTERNAL instead of EXT.

KPR #: 1650061572 Product: 8086/8 C 64818 03.70

One-line description:
Constant divided by short in function call generates wrong code.

Problem:

```
"C"
"8086"
#define A 800
extern short B
main()
{
   nothing(A/B);
}

nothing(D)
short D;
{
  int E;
  E = D;
}
```

generates the following (wrong) code

```
rev 3.70                                rev 3.20
MOV   AX,#0320H                         MOV  AL,3018H
CWD                                     CBW
MOV   AL,3018H (AL OVERWRITTEN !)       MOV CX,AX
CBW                                     MOV AX,#0320H
MOV CX,AX   (DIVISION RESULT IS         CWD
IDIV  CX     ALWAYS ONE )               IDIV CX
PUSH  AX                                PUSH AX
```

Temporary solution:
The customer is using Rev. 3.20 as a temporary fix. But, in Rev. 3.70 A/B can be placed into a temporary variable and then passing the temporary to the function.

KPR #: 5000135285 Product: 8086/8 C 64818 03.00

One-line description:
1006 message generated when referenced to unspecified array element

Problem:
Misleading error message generated when incorrect reference to an array in a structure is made. An 1006 (contact HP error) is generated

```
"C"
"8088"
struct CN {int f; int cnt[5];} cn;
struct CM {int i; struct CN *p;} cm;
main(){ cm.i = ++cm.p->cnt; }
                          ^1006 {invalid C, but misleading error msg}
```

Temporary solution:

KPR #: 5000135285 **CONTINUED**

Use array index in stmnt:

```
      cm.i = ++cm.p->cnt[3];
```

KPR #: 5000176891 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Bad code generated when casting a real constant into an integer

Problem:
Bad code generated when casting a real constant into an integer.
Example:

```
"C"
"processor"
main() {
        int   i;
        i = (int)0.0; /* generates code to move some value other
                         than 0 into i  ----- BUG ------  */
}
```

Temporary solution:
There is no known work around at this time.

KPR #: 5000191361 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
When $POINTER_SIZE 32$ generates 32 bit arithmetic for 16 bit variables

Problem:
32 bit arithmetic used when 16 bit called for. Example:

```
"C"
"80186"
$POINTER_SIZE 32$
main() {
        int    *p;
        unsigned int  i,j;
        if ( p && (( i - j ) > j )
                        ;
}
```

The left and right sides of the && are both evaluated using 32 bit arithmetic.

Temporary solution:
WORKAROUND
use:

```
     if ( p ==0 && (( i - j ) > j )    /*which causes the right
    hand side to be evaluated using 16 bit arithmetic  */
```

KPR #: 5000214858 Product: 8086/8 C 64818 03.01

Keywords: CODE GENERATOR

One-line description:
Bad code generated when left shift short variable & AND w/ unsigned int

Problem:
Bad source code generated with left shift:

```
"C"
"8086"
$LIST_OBJ$
unsigned short i;unsigned int j;
main()
{
if (j & (1<<i));
       MOV CL,DS:BYTE PTR Dstatic+1H
       MOV AL,#1H
       CBW
       SHL AX,CL   <----- This is incorrect.  The correct code is:  |
D2E0 <---------  SHL AL,CL  <---------------|
       ......
}
```

Temporary solution:
There is no known fix at this time.

KPR #: 5000229245 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Compiler generates MOV SP, BP and LEAVE. This is redundant.

Problem:
The 8086 C Cross Compiler generates redundant code when compiled with the 80286 option. When a Function is exited the compiler generates the the following instructions. MOV SP, BP and LEAVE. MOV SP, BP is implicit to the LEAVE instruction. This minor problem only appears in the 64100A version of the Compiler. This redundancy does not appear when compiled on the HP-UX Version of the Cross Compiler.

Temporary solution:
There is no know work around at this time.

KPR #: 5000278127 Product: 8086/8 C 64818 03.70

One-line description:
Bad code generated on 64000 with "80286" directive.

Problem:
Bad Code generated when "80286" directive is used with C compiler on 64000. The following function generates bad code:

```
"C"
"80286"
$POINTER_SIZE 32$
int  slength(s)
```

KPR #: 5000278127 **CONTINUED**

```
char  *s;
{     int   i;
      while(*s++)
            i++;
}
```

Temporary storage for the compiler overwrites values pushed on the stack as parameters to an Add32 library call.

NOTE: The ENTER command is not used on the 64000, but the offsets look as if it had been. Also, the LEAVE command is used without the ENTER command.

Temporary solution:
No known fix at this time.

KPR #: 5000294199 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300

One-line description:
& address operator generates PUSH DS1 when DS1 not defined.

Problem:
The compiler uses a wrong register(DS1),which is not defined before using it.

```
"C"
"70116"
$EXTENSION  ON$ $WARN OFF$ $SEPARATE CONST OFF$ $POINTER SIZE 32$
$RECURSIVE OFF$ $OPTIMIZE ON$ $INIT ZEROS OFF$ $FAR LIBRALIES ON$
$FAR PROC ON$
extern unsigned int wtest,wtest1;
test()
{
       unsigned int *lptr;
       lptr=&wtest+wtest1;  -----> This generates PUSH DS1
                                   DS1 is not defined any place.
}
```

We can not use Rev 3.7,we decide to use Rev 3.2 again.

KPR #: 5000402214 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300

One-line description:
Wrong code generated for structure in while loop.

Problem:
Compiler generates incorrect codes to pointer operation when it is in 'while' loop.
EXAMPLE

```
      "C"
      "8086"
```


KPR #: 5000402214 **CONTINUED**

```
      struct {
             unsigned char a;
             unsigned char b;
             unsigned char c;
             }ggg[10];
      unsigned char   i,j,k;
      main()
      {
             while (i <= k){
                          j=ggg[i].b;   <--- incorrect codes generated
                          }
      }
```

The above code does indeed generate incorrect code:

```
   while (i<=k) {
        j=ggg[i].b;
   }
         .
         .
         .
        MOV   AL,#+00003H
        MUL   AL            <---- Should MUL with AL and I
        MOV   BX,AX
        MOV   AL,DS:BYTE PTR Dstatic[BX+00001H]
        MOV   DS:BYTE PTR Dstatic+0001FH,AL
         .
         .
         .
```

If ggg[i].b is repeated, the correct code is generated.

KPR #: 5000406348 Product: 8086/8 C 64818 03.70

One-line description:
1006 Fatal error when use structure pointers inside IF statement.

Problem:
The following code generates a 1006 Fatal Error. It seems the "pb->put += n" statement causes the error. This problem was entered under SR# D200027813 as a documentation error in 1985. However, it does not seem to have ever been entered as a software problem that should be fixed.

```
"C"
"8086"

struct buft {
  char  *pbuf;
  char  *get;
  char  *put;
  char  *buflim;
  short  siz;
  short  lowater;
```

KPR #: 5000406348 **CONTINUED**

```
  short  hiwater;
};

main()
{
  int n;
  struct buft pb;

  if ((pb->put += n)  >= pb->buflim)
     pb->put -= pb->siz;
}
```

Temporary solution:
Break up the statement:

```
pb->get += n;
if (pb->get >= pb->buflim)
  pb->put -= pb->siz;
```

KPR #: D200008342 Product: 8086/8 C 64818 00.56

Keywords: CODE GENERATOR

One-line description:
Vars ORGed in seg. 0 in SHORT env. access current DS seg with no warning

Problem:
In the SHORT environment (16-bit pointers) all variables ORGed in segment zero (0) will be accessed as an offset from DS. If DS <> 0 the actual location will not have an address equal to the value selected in the ORG statement in the source. This feature of the compiler permits the address of variables ORGed in segment 0 to be taken in the short environment. However, if the system under design has an I/O port in segment 0, but DS <> 0, there will be problems the cause of which may not be readily apparent. A warning message should be given whenever a variable is ORGed to segment 0, stating that that variable will be located in the current DS segment.

Temporary solution:
If the address of an ORGed variable does not have to be taken (assigned to a pointer, or variable an array, etc.) and the address value of the variable >= 10H, then the variable can be ORGed in a segment other than 0 and the addressed accessed will reflect the value in the ORG statement

---

KPR #: D200038836 Product: 8086/8 C 64818 02.00

One-line description:
Compiler using unacceptable amount of stack space for procedure returns.

Problem:
When procedures are called that return values greater than 16 bits long (32 bit pointers), temporary variables are used to store the result. These temporary variables are created on the stack and are used only for temporary result storage. The current version of the

KPR #: D200038836 **CONTINUED**

8086 C compiler creates a new temporary every time it needs one, even though it will only be used once. The previous version (1.05) generated only as many temporaries as it required and reused them throughout the procedure.

Temporary solution:
No known temporary solution at this time.

KPR #: D200049908 Product: 8086/8 C 64818 03.00

One-line description:
Illegal instruction generated by ASM_FILE

Problem:
An illegal assembly instruction is generated for an assignment statement. The ASM_FILE contains a move of a byte register to a word register.

```
struct {
         char cnt;
         char chain;
         char param[62];
       } parmabuf[20];
stuct  tache{
           struct{
                 struct{
                       genre         :1;
                                     :1;
                       fonction      :4;
                       classe        :2;
                       destination   :3;
                       origine       :3;
                       type          :2;
                       } entete;
                 char sapi;
                 char ces_tei;
                 union{
                       char shparam[2];
                       int  numbuf;
                      } prim[6];
           struct{
                              :8;
                 prm          :6;
                              :1;
                 libcount     :1;
                } event;
           char pr,pv;
          } tache2;
main()
{
int i,k;
tache2.prim[k].sapi = parambuf[j].param[2];
.
.
      MOV  AX,CL     ; illegal assmbly instruction
```

KPR #: D200049908 **CONTINUED**

```
.
;
}
```

Temporary solution:
No known temporary solution.

KPR #: D200063057 Product: 8086/8 C 64818 03.01

One-line description:
Conditional expressions with unsigned mixed operands may fail

Problem:
Conditional expressions with unsigned mixed operands may fail.

Temporary solution:
There is no known work around at this time.

KPR #: D200068080 Product: 8086/8 C 64818 03.01

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"

struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200068700 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Casting ptr. to int as short & incrementing it generates bad code

KPR #: D200068700 **CONTINUED**

Problem:

The following table describes the nature of each compiled file and on the 64100.

test case	"if" used	Ptr size	number of increments; statement separation	increment and gets separate statements	BUG DESRCIPTION
TEST1	yes	32	2 ;	no	Reboots system
TEST2	no	32	2 ;	no	No increment in listing
TEST3	yes	32	2 ,	no	No increment in listing
TEST4	yes	16	2 ;	no	Reboots system
TEST5	no	16	2 ;	no	compiles correct
TEST6	yes	16	2 ,	no	Reboots system
TEST7	yes	32	1	no	No increments in listing
TEST8	yes	16	1	no	Reboots system
TEST9	no	32	1	yes	Error in factor message
TEST10	no	16	1	yes	Error in factor message
TEST11	no	32	1	yes	No increment in listing
TEST12	no	16	1	yes	No increment in listing

The following table describes the nature of each compiled file and on the 64100.

test case	"if" used	Ptr size	number of increments; statement separation	increment and gets separate statements	BUG DESRCIPTION
TEST1	yes	32	2 ;	no	Reboots system
TEST2	no	32	2 ;	no	No increment in listing
TEST3	yes	32	2 ,	no	No increment in listing
TEST4	yes	16	2 ;	no	Reboots system
TEST5	no	16	2 ;	no	compiles correct
TEST6	yes	16	2 ,	no	Reboots system
TEST7	yes	32	1	no	No increments in listing
TEST8	yes	16	1	no	Reboots system
TEST9	no	32	1	yes	Error in factor message
TEST10	no	16	1	yes	Error in factor message
TEST11	no	32	1	yes	No increment in listing
TEST12	no	16	1	yes	No increment in listing

Temporary solution:
There is no known fix at this time.

KPR #: D200069716 Product: 8086/8 C 64818 03.01

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:

KPR #: D200069716 **CONTINUED**

Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200071787 Product: 8086/8 C 64818 03.01

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources

KPR #: D200071787 **CONTINUED**

to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory, remove the DATA pseudo, and reassemble.

KPR #: D200076695 Product: 8086/8 C 64818 03.02

One-line description:
Fields of a structure are dereferenced incorrectly (if fields are big).

Problem:
Structure ponters are not being calculated correctly when relative addressing requires offsets of large sizes. See following code.

```
"C"
"68000"

$FAR$

struct this{
         unsigned short int first[256][256];
         unsigned short int second[256][256];
} one,*bufptr;

unsigned short int *desptr;

main()
{
 bufptr = &one;
 destptr = bufptr->first[0][0];
 destptr = bufptr->second[0][0];   /* Same address assigned. */
 }
```

NOTE: The 8086 line of compilers will generate a pass three error for this code. "Program Counters Disagree. "

Temporary solution:
For the 68000 family of cross compilers you may use the '.' operator instead.

```
"C"
"680XX"

$FAR$

struct  this {
         unsigned short int  first [256][256];
         unsigned short int  second[256][256];
} Structure,*ptrToStruct;
```

KPR #: D200076695 **CONTINUED**

```
unsigned short int *destptr;
main()
{

 destptr = &Structure.second[0][0];
}
```

KPR #: D200079343 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
~, &, | and ^ may not correctly expand shorts in condidionals

Problem:
The complement operator and the bitwise AND, OR, and XOR operators do not correctly extend signed 8-bit quantities. For example, in the following code segment for Z80 "s" is extended as if it were an unsigned quantity before the OR operation is performed. NOTE: no extend is done, and an 8-bit OR performed if $SHORT_ARITH ON$ is in effect.

```
short s;

main()
{
   if (s&0x01)
       LDA s
       MVI H,000H
       MOV L,A
       LXI D,00001H
       CALL Zwor
          .
          .
          .
}
```

KPR #: D200079608 Product: 8086/8 C 64818 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
```


KPR #: D200079608 **CONTINUED**

```
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
       }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081513 Product: 8086/8 C 64818 03.02

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )     /* gives "Illegal type of operand(s) */
          ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;
```

OR

cast the variable to an int:

KPR #: D200081513 **CONTINUED**

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200085738 Product: 8086/8 C 64818 03.70

Keywords: CODE GENERATOR

One-line description:
PASS 2 error when pntr type used to invoke code stored in array.

Problem:
When the typedef statement is used to define pointers to functions, and this pointer type is used in a cast of a variable array to invoke code stored in that array, program execution may transfer to the wrong location. For example, in the following code the simple call to code_array fails while the call and assignment to p works correctly:

```
typedef int(*PFI)();  /* PFI a pointer to int functions */
int code_array[100];  /* array contains code */
PFI p;                /* p a pointer of type PFI */

pfibug()
{
  (*((PFI) code_array))();  /* fails in JSR to code_array */
  (*(p=(PFI)code_array))(); /* assignment and JSR successful */
}
```

Temporary solution:
Set up a dummy variable and perform an assignment to it when doing this type of operation.

KPR #: D200090332 Product: 8086/8 C 64818 03.70

One-line description:
IF statement loads wrong segment for compare statement.

Problem:
Code generated by the IF statement in the following program loads the wrong segment at the CMP level. It seems it is the extra segment being overwritten.

```
"C"
"70108"
$SEPARATE_CONST OFF$
$FAR_LIBRARIES ON$
$RECURSIVE ON$
$AMNESIA ON$
$FAR_PROC ON$
$POINTER_SIZE 32$
$FAR_EXTVARS ON$

struct S1
{
  char *p1;
  int entier;
  char c[10];
} s1;

struct S1 *p0;

main()
{
  toto(p0);
}

toto(pointeur)
struct S1 *pointeur;
{
  if (*pointeur->p1 == pointeur->c[pointeur->entier])  {};
}
```

Temporary solution:
A temporary fix may be to break-up the statements being compared:

```
toto(pointeur)
struct S1 *pointeur;
{
   char  temp1,temp2;
   temp1 = *pointeur->p1;
   temp2 = pointeur->c[pointeur->entier];

   if (temp1 == temp2) {}
}
```

KPR #: D200092080 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300

One-line description:
Long arithmetic expression generates incorrect code.

Problem:
The following expression generates incorrect code:

```
"C"
"8086"

main()
{
  int A, B, C, D, E;

  E = (A - (B/2)) - ((C+D) /2);
}
```

---

generates:

```
      MOV     AX,SS:WORD PTR [BP-00008H]    <-- B
      CWD
      MOV     CX,#+00002H
      IDIV    CX                            <-- B/2
      MOV     DX,SS:WORD PTR [BP-0000AH]
      SUB     DX,AX                         <-- (A-(B/2))
      MOV     AX,SS:WORD PTR [BP-00006H]    <-- C
      ADD     AX,SS:WORD PTR [BP-00004H]    <-- C+D
      MOV     AX,DX               <---- overwrites (C+D)
      CWD
```

Temporary solution:
The temporary fix is to place sections of the long expression into temporary variables, then evaluate:

```
F = A - (B/2);
E = (C+D) /2;
E = F - E;
```

KPR #: 5000138941 Product: 8086/8 PASCAL 64814 03.00

One-line description:
Out of expression storage error generated on code that ran on old ver.

Problem:
Customer has some code that originally compiled on both the 9000 and the 64100. With the latest released version (3.00) he can no longer compile the same file on the 64100. When he tries to compile, he gets a 406:Out of expression tree storage;simply expression.

For example:

```
  VAR X : ARRAY [0..50] OF BOOLEAN;
      A ; BOOLEAN;
  BEGIN
    A := (X[0] OR X[1] OR X[2] ...X[18]);
END;
```

Temporary solution:
Break up expression into simplier pieces.

KPR #: 5000146829 Product: 8086/8 PASCAL 64814 03.00

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory, remove the DATA pseudo, and reassemble.

KPR #: 5000244392 Product: 8086/8 PASCAL 64814 03.02

Keywords: CODE GENERATOR

One-line description:
ERROR 117 generated, but does not indicate variable in error

Problem:

KPR #: 5000244392 **CONTINUED**

Pascal Compiler error 117 "Unsatisfied Forward Reference" does not provide information as to which reference is unsatisfied when the reference is a pointer to an undefined type. The following program gives an example:

```
        "8086"
        PROGRAM ERR117;
        TYPE
            UNSAT = ^UNSAT_TYPE;
        BEGIN
        END.
** ERROR??  ^117
117: Unsatisfied forward reference
```

There is no indication as to which reference is unsatisfied.

Temporary solution:
There is no known work around at this time.

KPR #: 5000272021 Product: 8086/8 PASCAL 64814 03.02

One-line description:
CASE statement produces bad code for complicated expression

Temporary solution:
A temporary fix for the problem is to place the complicated expression used in the CASE statement into a temporary variable. Then, use this temporary variable in the CASE statement instead of the complicated expression.

Signed off 02/03/89 in release A03.90

KPR #: D200006080 Product: 8086/8 PASCAL 64814 00.46

Keywords: CODE GENERATOR

One-line description:
Data structures larger than 64K are not flagged as an error.

Problem:
Variables (data structures) which require more than 64K of contiguous memory are not flagged as an error. Please refer to the example shown below.

```
   $POINTER_SIZE 32$
   VAR
      RAM : ARRAY [0..3FFFFH] OF BYTE;
      I : INTEGER;

   BEGIN
      FOR I := 0 TO 3FFFFH DO
         RAM[I] := 0FFH
   END.
```

In the above example, the first 64K bytes (one segment) of memory are assigned three times.

KPR #: D200006080 **CONTINUED**

Temporary solution:
No known workaround at this time.

KPR #: D200010280 Product: 8086/8 PASCAL 64814 00.60

Keywords: CODE GENERATOR

One-line description:
Byte values may be converted to 16-bit before comparison with byte var.

Problem:
Byte variables may be converted to 16-bit before being compared with another byte variable (constant). Please refer to the following example for further details.

```
PROGRAM TEST;
$EXTENSIONS ON$
VAR A,B : SIGNED_8;

BEGIN               { contents of A:  0H  |          0FFH              }
  CASE A OF         { A->AL, CBW    :  0H  |         0FFFFH            }
    0FFH : B:=0;    {                      | 000FF<>0FFFFH : Case error }
    0    : B:=1     {                ok    |                            }
  END
END.
```

In the example above, 0FFH is converted to a 16-bit value before being compared to the byte variable A. When 0FFH is converted to a 16-bit quantity, the resulting value is 0FFFFH, which will never equal the value of the byte variable A. If the value that is to be compared to A is less than 080H, then conversion to a 16-bit value is not performed.

Temporary solution:
No known workaround at this time.

KPR #: D200014944 Product: 8086/8 PASCAL 64814 01.10

Keywords: RUN-TIME LIBRARY

One-line description:
Failed to detect out-of-bounds case.

Problem:
A run time out-of-bounds error is not indicated for the following program.

```
  PROGRAM TEST;

     CONST
        CON_1 = 32768;

     PROCEDURE PASS_1 (CON_1 : SIGNED_16);
```

KPR #: D200014944 **CONTINUED**

```
        VAR
           LOCAL_1 : SIGNED_16;

        BEGIN
           LOCAL_1 := CON_1;
        END;

     BEGIN
        PASS_1 (CON_1);
     END.
```

Temporary solution:
No known workaround at this time.

KPR #: D200027516 Product: 8086/8 PASCAL 64814 03.02

One-line description:
Nested IFs inside a WITH may generate incorrect code.

Problem:
Incorrect code may be generated by the 8086 pascal compilers in the UNIX environment when nested IF's are used inside a WITH statement.

Temporary solution:
Use simpler, less compilex nesting.

KPR #: D200047779 Product: 8086/8 PASCAL 64814 02.01

One-line description:
80186Generates wrong offset within CONST_data area

Problem:
The statement:

```
    ELA   BX,DS:CONST_data+000183H
```

causes a wrong condition as the CONST_data area goes up to approximately 100H offset.

Temporary solution:
There is no know work around at this time.

KPR #: D200060061 Product: 8086/8 PASCAL 64814 03.00

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
```


KPR #: D200060061 **CONTINUED**

```
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200090597 Product: 8086/8 PASCAL 64814 03.50

Keywords: PROBLEM ON 9000/S300

One-line description:
Assignment of constant into array of 3 elements does not work.

Problem:
When the following code is compiled CONST_data is never defined even though it is referenced in the assembly code.

```
"8086"
$EXTENSIONS ON$
PROGRAM test;
CONST
  SYNC_PATTERN = 20F3FAH;      <---------  32 Bits
TYPE
  UB = UNSIGNED_8;
  UB3 = ARRAY[1..3] OF UB;     <---------  24 Bits
VAR
  SYNC : UB3;
BEGIN
  SYNC := UB3(SYNC_PATTERN);  <----------  LEA  SI,DS:CONST_data
END.
```

Note: The array element is not an even multiple of bytes, therefore it is unclear what the compiler is supposed to do with the constant. The assembly code generated uses CONST_data without defining it.

KPR #: D200090597 **CONTINUED**

Temporary solution:
In this particular situation a fix would be to change the array declaration of [1..3] to [1..4]. That is, to use 32 bits instead of 24. The user must be very careful and make sure the proper code is being generated. The compiler will save these bits in a particular order - which may not be the order the customer had desired! Refering to page 10-12 in the HP-UX Hosted Pascal Cross Compiler - 8086 manual might be helpful. Refer to the section on Functional Type Change.

KPR #: D200093476 Product: 8086/8 PASCAL 64814 03.50

One-line description:
Boolean Index into array generates bad code

Problem:
Detailed Listing for Defect Number LSDqf04464
Text:
boolean index into array generates bad code

```
$EXTENSIONS$

VAR
       bool   :  BOOLEAN;
       int    :  INTEGER;
       arr3 :  ARRAY[BOOLEAN,1..4,BOOLEAN] OF BYTE;
{ ====== END of Declarations for LSDla01274  ====== }

PROCEDURE LSDla01247;

      {Initialize array arr3}
   FOR int:= 1 TO 4 DO
    FOR bool:= TRUE DOWNTO FALSE DO
     BEGIN
      arr3[bool,int,FALSE]:= 2;
      arr3[bool,int,TRUE]:= -2;
     END;
   bool := FALSE;
                { variable::constant::variable }
  IF  arr3[bool,2,bool] = 2  { arr3[FALSE,2,FALSE] = 2}
    THEN
      {OK}
    ELSE
      {  "** ERROR LSDla01274 #16 in file TEST110:Plus_P **."  };

                { constant::constant::variable }
  bool := TRUE;
  IF  arr3[FALSE,4,bool] = -2
    THEN
      {OK}
    ELSE
      {  "** ERROR LSDla01274 #19 in file TEST110:Plus_P **."  };
END;
```

KPR #: D200093484 Product: 8086/8 PASCAL 64814 03.50

One-line description:
Test for set inclusion checks beyond the set boundary.

Problem:
Detailed Listing for Defect Number LSDqf04465
Text:
test for set inclusion checks beyond the set boundary.

```
$EXTENSIONS$

TYPE
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
        {digch :: Set of character; 256 bit set takes 8 bytes }
   digch = SET OF CHAR;

   Array256= ARRAY [0..255] OF BYTE;
VAR
   DIGIT : DIG;
   PADDING: Array256;
   DA : STRING;
{-------------------------Possible FIX & test ------------}
   digitset : digch;
   ch : CHAR;

{ ====== BEGIN Test Procedure for LSDla00270  ====== }
 PROCEDURE LSDla00270;
  VAR index: INTEGER;
  BEGIN
{------*Problems occur due to testing unrelated/unitialized values ---}
{           OUTSIDE of the legal boundaries for the declared SET.      }
{         In this case all the letters have values > '9' and should    }
{         never be tested and should always fail.                      }
{        It appears that this overflow will be detected if $RANGE$ is  }
{        on, but it is entirely ignored.                               }
{  DEMONSTRATE DEFECT by filling the PADDING array                     }
{          with 1's to make sure NO elements erroneously identified!   }


   DIGIT := DIG['0','1','2','3','4','5','6','7','8','9'];
   DA := 'A09z';

   FOR index:= 0 TO 255 DO
       PADDING[index]:= 0FFH;

   IF DA[1] IN DIGIT            { DA[1]='A' is NOT in DIGIT set }
    THEN
      {  "** ERROR LSDla00270 #9 in file TEST111:Plus_P **."  }
  END;
```

KPR #: D200093518 Product: 8086/8 PASCAL 64814 01.90

One-line description:
Error 1006 for complex statement using MOD operator

Problem:
ERROR 1006 WHEN USING TYPE CONVERSION WITH MODULO OPERATION.

THE FOLLOWING SAMPLE PROGRAM WILL PRODUCE AN ERROR 1006 :

```
"PASCAL"
"8086"

$EXTENSIONS$

PROGRAM ERR1006 ;

VAR I : INTEGER ;
    B : BYTE ;
BEGIN
  B := BYTE((SIGNED_16(ADDR(I))) MOD 100H)
END.
```

Temporary solution:
Break this statement into two separate statements as follows.

```
VAR I,temp: INTEGER;
    B:  BYTE;

   BEGIN
      temp := SIGNED_16(ADDR(I));
      B := BYTE (I MOD 100H);
   END.
```

KPR #: 5000238337 Product: 8086/88 PASCAL M 64814-90903 03.00

Keywords: MANUAL

One-line description:
Change manual to say that libraries need to be in same segment

---

KPR #: 5000131029 Product: 8086/88/186/188HLSAM 64332-90902 02.00

One-line description:
Display variable may result in "ERROR:E64".

KPR #: 5000141150 Product: 8086/88/186/188HLSAM 64332-90902 02.00

One-line description:
Data structures too large to display in "display variable" command.

Temporary solution:
Reducing the ascii string size of the variable names
shall cause less space to be used in the 64340 analyzer.
Therefore development can continue with the freed space.

Duplicate Service Requests: 5000141143

KPR #: D200081273 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:
Display Memory and Modify Memory will be incorrect at the
segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the
center of a line.

Modify Memory will be incorrect if done beyond the end of a
segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment
in the middle of the line being displayed.

Modify Memory will be correct if it is not extended through the end
of a segment. For example:

modify memory 0FFFEH to 1,2 will be correct.

modify memory 0FFFEH to 1,2,3 will NOT be correct, because
the third entry is in the next segment.

KPR #: D200081422 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command
files using the users PATH variable for a search path. A defect has
been introduced such that specifying a relative path with a command
file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a
standard shell. Only names not containing any '/' should be searched
for using PATH. All others (especially ./name) should be used relative
to the current directory.

Temporary solution:
Specify command files with full path names if the application is unable
to find your command file.

KPR #: D200082156 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed

KPR #: D200082156 **CONTINUED**

some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083162 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200084947 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

KPR #: D200085969 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

The symbols will not be displayed even if you try to re-execute step

KPR #: D200085969 **CONTINUED**

number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200089920 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Emulator does not work reliably with 64155B memory controller

Problem:
Detailed Listing for Defect Number LSDqf03557
Text:
emulator does not work reliably with 64155B memory controller

Any 8-bit processor which uses a foreground monitor may have strange problems when using the 64155B memory controller. This is due to the MONITOR_CONTROL word being 16 bits wide. The 8-bit processors require two bus cycles to modify this word. Unfortunately, it is possible to read the value between those two bus cycles, resulting in a bogus value being passed up to the host. The likelihood of this happening depends on the timing of the monitor - using a different assembler/linker or changing the monitor code can cause the problem to appear or disappear. A simple example is the 68008DP: running a particular program in target memory, then attempting to modify target memory produces an undefined software breakpoint message.
Detailed Listing for Defect Number LSDqf03557
Text:
emulator does not work reliably with 64155B memory controller

Any 8-bit processor which uses a foreground monitor may have strange problems when using the 64155B memory controller. This is due to the MONITOR_CONTROL word being 16 bits wide. The 8-bit processors require two bus cycles to modify this word. Unfortunately, it is possible to read the value between those two bus cycles, resulting in a bogus value being passed up to the host. The likelihood of this happening depends on the timing of the monitor - using a different assembler/linker or changing the monitor code can cause the problem to appear or disappear. A simple example is the 68008DP: running a particular program in target memory, then attempting to modify target memory produces an undefined software breakpoint message.

KPR #: D200090787 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source

KPR #: D200090787 **CONTINUED**

on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095638 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200095927 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: 5000191767 Product: 8096 ASSEMB 64860 01.03

Keywords: CODE GENERATOR

One-line description:
Linker does not allocate the file at even addresses

Problem:
The 8096 linker does not allocate files at even addresses.
The following example shows this problem:

```
file_a     "8096"                  file_b      "8096"
            L1  DSL  2                          L3  DSL  2
            L2  DSW  1                          L4  DSW  1
            ST  SP,L2                           ST  SP,L4
```

Link above files with load address prog=20H.

FILE/PROG NAME	PROGRAM	DATA
file_a	0020	
next address	002F	
file_b	002F <----	Customer wants to allocate
next address	003E	this file from an even address, automatically from linker.

Temporary solution:
Two possible work-arounds exists. The first is to use an ORG statement in each file to place the file on an even boundry. The second work-around is to place the statement "DSL 0" at the end of each file, thus leaving the file on an even boundry.

KPR #: 5000225078 Product: 8096 ASSEMB 64860 01.03

Keywords: CODE GENERATOR

One-line description:
Using ORG statemnts can generate ERR_LR errors

Problem:
The following program will produce a Legal Range error on line 9,10 and 11.

```
 1 "8096"
 2            ORG        1800H
 3 CPTG1      DSW        2
 4            ORG        1C00H
 5 TEST       DSB        4
 6            PROG
 7 AX         EQU        10H
 8 AL         EQU        20H
 9            ST         AX,CPTG1
10            STB        AL,CPTG1+1
11            LDB        AL,TEST
```

KPR #: 5000225078 **CONTINUED**

Temporary solution:
Move ORG statement to end of the program.

```
 1 "8096"
 2          PROG
 3 AX       EQU       10H
 4 AL       EQU       20H
 5          ST        AX,CPTG1
 6          STB       AL,CPTG1+1
 7          LDB       AL,TEST
 8          ORG       1800H     <-*
 9 CPTG1    DSW       2         <-*
10          ORG       1C00H     <-*
11 TEST     DSB       4         <-*
```

KPR #: 5000275305 Product: 8096 ASSEMB 64860 01.04

One-line description:
Pseudo instruction DCB treats absolute variable as relocatable.

Problem:
The pseudo instruction DCB is not treated correctly by linker. The label which is defined by DSW is used as operand of DCB. After linked, the label is not assigned the proper value. The value is not absolute but relocatable. The following is a example;

```
     "8096"
     AX    DSW  1
           ST   SP,AX
           DCB  AX
```

After the above program is linked, AX remains as relocatable value at DCB line. But the AX in ST SP,AX is assinged the absolute value.

NOTE: Since no emulator exists for this processor, the problem can be verified by looking at the :absolute file. When the program is linked with PROG address set to 80H, the :absolute file shows:

```
                    C301  8000  1800
                                ||
                                ||
          This 00H is not the absolute value.
          Load address of PROG is 80H, so the code should be 80H.
```

Therefore, the correct code must be "C301 8000 1880".

KPR #: D200090928 Product: F9450 EMUL 300 64286S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200075150 Product: F9450 EMULATION 64286 01.04

One-line description:
RS232 Simulated IO will overrun the user's read buffer sometimes.

Problem:
R232 Simulated IO will overrun the user buffer under certain conditions. If the Last Byte Address Pointer coincidentally is pointing to the location of the Read Buffer End Address Pointer, and the 64000 station has received characters input to the RS232 port, when the "Update Read Buffer" (8CH) command is given, the 64000 will write the new bytes to the memory following the Buffer End Address. The 64000 should write the new characters to the Buffer End Address location and then wrap around to the Buffer Begin Addrss location. Instead the new characters continue to be written to ever increasing address locations.

Duplicate Service Requests: 5000194373

KPR #: D200087395 Product: GENERIC ANALYSIS M 64740-90909 01.00

One-line description:
Errors in xtt help screen.

Signed off 01/26/89 in release X00.00

KPR #: D200085530 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Ending value of data stream does not report proper error.

Problem:
When performing a memory modify like the following:

```
>m 0..3=1,2,3,4:junk
```

where the stream of data values is equal to the buffer of memory we which to fill, no error is reported on the last value of 4:junk. The memory does get correctly modified to the values 1,2,3,4 however an error should be reported on the 4:junk value.

KPR #: D200091264 Product: GENERIC EMULATION FW 64700 00.05

One-line description:
Odd byte format records may cause an extra byte written to memory

Problem:
An odd number of bytes contained in an HP format absolute record can cause an extra byte to be written to memory. This problem can be seen on the Z80 emulator and possibly the 186. (not seen on the 68000).

This does not effect the execution of a program which has been loaded. The primary problem seen by the user is when calculating a checksum over the data loaded into memory. The extra byte written with an odd record is random in value.

Temporary solution:
There is no known workaround available.

KPR #: D200085647 Product: GENERIC EMULATION FW 64700-90901 01.00

One-line description:
Improper coverage calculation of overlapping ranges

Problem:
If coverage is used, and multiple ranges are used which overlap, the calculation of coverage sometimes gets an incorrect value. For example, if only location 0fffh has been accessed, the following command is wrong (a cov -r has been done to initialize coverage):

```
cov 0ffe..1000 0fff..1000
percentage of memory accessed: % 40.0
```

The correct result is given by

```
cov 0ffe..1000
percentage of memory accessed: % 33.3
```

Since multiple ranges are allowed, the coverage algorithm should be fixed so that overlaps are correctly computed.

KPR #: D200090134 Product: GENERIC EMULATION FW 64700-90901 00.01

One-line description:
The "stty" command doesn't work correctly for baud rate <= 1200.

Problem:
If you toggle the xon parameter when running at 1200 baud and below, the stty command will return invalid characters.

```
>stty
 stty A 1200 xon
>stty -xon
 [#!,*&^junk characters
```

Since the PC interface calls the stty command upon startup, this problem will make the PC interface fail at startup with a datacomm error at 1200 baud (all lower baud rates are not supported by the PC interface).

Temporary solution:
To get around this problem, just set switch 13 on the emulator's back panel (enable xon). The stty parameter will not be toggled and PC interface will startup successfully.

From the terminal-mode interface, just enter another carriage-return to regain proper communications.

Duplicate Service Requests: D200090142

KPR #: 5000163303 Product: HOST PASCAL 64817 01.04

One-line description:
IOERROR not generated.

Problem:
The following program does not result in an IOERROR # 11 when a string instead of an integer is entered from the keyboard. Using $IOCHECK ON$, the error is detected.

```
VAR I,J : INTEGER;
    F   : TEXT;
BEGIN
RESET(F,'keyboard');
$IOCHECK OFF$
READLN(F,I);
IF IOERROR <> 0 THEN
   BEGIN
     J := IORESULT;
     WIRTELN('ERR');
   END;
END.
```

KPR #: D200014357 Product: HOST PASCAL 64817 01.04

One-line description:
Spurious run-time error doing WRITE(REAL_VAL) after previous I/O error

Problem:
WRITE(REAL_VAL) fails to reset the I/O error indicator. The result is a spurious run-time error if one writes a REAL value immeadiately after a previous I/O error or end-of-file condition. For example,

```
VAR R:REAL;
BEGIN
WHILE NOT EOF DO {Eventually produces end-of-file, an error}
  READLN;
WRITELN(R);      {Write real immeadiately after EOF causes run-time
                  error erroneously.}
```

Temporary solution:
Put a dummy I/O operation before the write of the real. For example, one could write the null string before writing the real. The intervening I/O call resets the error indicator.

```
WHILE NOT EOF DO
  READLN;
WRITELN('',R);  {Writing null string resets error indicator}
```

KPR #: D200015305 Product: HOST PASCAL 64817 01.04

One-line description:
STRWRITE function may produce run time error in specific case.

Problem:
The following HOST Pascal program will produce a run time error based on the STRWRITE function. This is incorrect since only one

KPR #: D200015305 **CONTINUED**

item is being written into the string 's'.

```
program test (input, output);

   var
      s : string[3];
      d : integer;

   begin
      setstrlen (s, 3);
      strwrite (s, 3, dummy, 'c');
   end.
```

Temporary solution:
As a temporary work-around check the value of 'dummy' and reset to STRMAX (s) or less if necessary.

KPR #: D200079483 Product: HOST SOFTWARE 300 64883 01.00

Keywords: TRANSFER

One-line description:
Transfer does not handle extra line-feeds in file.

Problem:
Transfer needs to correctly handle "extra" line-feed characters which may be in the host file. These line-feeds appear in files which have been transferred from the VAX to the 9000 series 300.

KPR #: D200079681 Product: HOST SOFTWARE 300 64883 01.00

Keywords: TRANSFER

One-line description:
Incorrect syntax/usage may not result in warning or error message.

Problem:
High Speed Link (transfer -h) software may not always catch the use of invalid file names or illegal syntax. For example, the command

```
$ transfer -tah file1 file1@1
```

will transfer the file "file1" into "FILE1::source@0". Note that the file is transferred to the wrong cluster. Transfer should (1) copy the file to cluster 1, or (2) flag the transfer statement as syntactically incorrect (the 64000 file name is lower case).

Signed off 02/03/89 in release A01.30

KPR #: D200085076 Product: HOST SOFTWARE 300 64883 01.00

One-line description:
Cluster - Cluster Transfer does not work with filelist

Problem:
Cluster - Cluster transfer via High Speed Link with a filelist does not work. The first file is transfered and a message indicates this is so, however, the second file is not transfered, and the only message is that it is NOT transfered - no other messages. None of the other files in the list are transfered or attempted to transfer. If the second file in the list does not exist, the message is:
ZZZZZZ:userid@n NOT transfered to ZZZZZZ:userid@m

Temporary solution:
Transfer the files one at a time.

KPR #: D200093609 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.

KPR #: D200093609 **CONTINUED**

The problem may also appear in any file list transfer after the 47th file in the list.

KPR #: D200093633 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

KPR #: D200093880 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

Signed off 02/03/89 in release A01.30

KPR #: D200093914 Product: HOST SOFTWARE 300 64883 01.10

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.

KPR #: D200093914 **CONTINUED**

The problem may also appear in any file list transfer after the 47th file in the list.

Signed off 02/03/89 in release A01.30

KPR #: 5000169698 Product: HOST SOFTWARE 500 64880 01.06

Keywords: TRANSFER

One-line description:
Transfer does not correctly parse "FILE:USERID:@HSL".

Problem:
Entering the command:

transfer -hast file FILE:USERID:@1

transfers the file to HSL0 with no comments. The leading colon in front onf @1 should generate a syntax error and produce no transfer instead. The correct syntax is:

transfer -hast file FILE:USERID@1

Signed off 02/03/89 in release A02.50

KPR #: 5000191544 Product: HOST SOFTWARE 500 64880 01.60

Keywords: TRANSFER

One-line description:
Transfer may not move library files.

Problem:
Create a relocatable file on a 64000 workstation using one of the available compilers or assemblers. Transfer this file to the 9000 computer.

Next, create a relocatable file on the 9000. Merge the 64000 created file and the 9000 created file into a library file using the cat(1) command, i.e.

```
$ cat file1.R file2.R > lib.R
```

An attempt at transferring the file 'lib.R' (either RS-232 or HSL) will fail. In the case of the High Speed Link, the error message returned is

```
WARNING: Memory fault
```

Temporary solution:
Create a file list containing the relocatable file names which belong in the library. Transfer ALL relocatable files to the 64000 system using the '-l' option (list option) of transfer.

Create a relocatable library file on the 64000 using the library command.

KPR #: D200036608 Product: HOST SOFTWARE 500 64880 01.20

One-line description:
Transfer to blank userid does not translate file names correctly.

Problem:
Translating files into the blank userid (":") results in incorrect file name translations. When downloading files for emulation, debug, ..., the blank userid should not be used.

Signed off 02/03/89 in release A02.50

KPR #: D200037275 Product: HOST SOFTWARE 500 64880 01.20

One-line description:
xx.L TO xx:link_sym translation wrong for 0 length records (types 3 & 4)

Problem:
When transfer (or translate) translates a host linker symbol file (.L file) which has a 0 length type record 3 or 4, the output file is incorrect.

KPR #: D200043877 Product: HOST SOFTWARE 500 64880 01.20

Keywords: RCMAIN

One-line description:
A session command is req'd before entering the menu in batch jobs.

Problem:
Entering the rcmain menu in a batch job before doing a select command results in hanging remote control. This is only true for select menue commands. Problems can be solved by making sure there is a blank after the menue command, AND doing a remote session command as the first command in the job.

KPR #: D200062539 Product: HOST SOFTWARE 500 64880 01.50

Keywords: TRANSLATE

One-line description:
C.K.1 and C.K.2 both translate to C_K on the 64000.

Problem:
C.K.1 and C.K.2 both translate to the same file name on the 64000 when a file is transfered.

Temporary solution:
None.

KPR #: D200093583 Product: HOST SOFTWARE 500 64880 01.90

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER

KPR #: D200093583 **CONTINUED**

error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

Signed off 02/03/89 in release A02.50

KPR #: D200093617 Product: HOST SOFTWARE 500 64880 01.90

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

Signed off 02/03/89 in release A02.50

KPR #: 1650016618 Product: HOST SOFTWARE VAX 64882 01.60

One-line description:
Transfer fails when downloading relocatable libraries

Problem:
Libraries of relocatables are created by appending the files together. Relocatables uploaded from the HP64000 station can be mixed with relocatables created on the host. Transfer does not handle the mixed source relocatable libraries correctly during download. The result of initiating such a transfer is unpredictable.

Temporary solution:
Each relocatable file could be transfered seperately and combined on the 64000 with the library command.

Signed off 02/03/89 in release A02.50

Duplicate Service Requests: 5000187922 1650025270

KPR #: 5000149724 Product: HOST SOFTWARE VAX 64882 01.60

Keywords: RCMAIN HIGH SPEED LINK

One-line description:
HSL transfer from within RCMAIN does not return control to RCMAIN.

Problem:
An HSL transfer from within RCMAIN, although it complete successfully, does not return control to RCMAIN.

If a 64000 was not selected prior to the transfer, it will additionally return an error message:
"Unrecognized flag (z) option."

Temporary solution:
This problem does not effect any transfer outside of RCMAIN nor RS232 transfers from within.

KPR #: 5000151290 Product: HOST SOFTWARE VAX 64882 01.60

One-line description:
RCMAIN corrupts RCDEVICE.dat file when aborted with Cntl C or Y

Problem:
If a vax terminal hangs up while in rcmain utility and I do a cntl c or a cntl y the rcmain.dat file gets corrupted and an orderly exit from rc main is not accomplished. need to enhance the software to allow for cntl c or cntl y to exit from rcmain if you cannot use the exit command in rc main.
We do need to enhance the software to accept the cntl y or c characters to allow for an orderly exit from the rcmain routine. at present the .da t file does get trashed when cntl c or y is executed
customer needs to be copied on this response...
If a vax terminal hangs up while in rcmain utility and I do a cntl c or

KPR #: 5000151290 **CONTINUED**

a cntl y the rcmain.dat file gets corrupted and an orderly exit from rc main is not accomplished. need to enhance the software to allow for cntl c or cntl y to exit from rcmain if you cannot use the exit command in rc main.

Temporary solution:
NO KNOWN WORK-AROUND

KPR #: 5000180323 Product: HOST SOFTWARE VAX 64882 01.70

Keywords: RCMAIN

One-line description:
RCDEVICE.DAT is not properly maintained.

Problem:
RCMAIN does not update the RCDEVICE.DAT file properly under certain conditions. Example: A small file with only two lines defining two station entries such as:

```
edp1    tty1
edp2    tty3
```

will sometimes not show a busy status when browsed even when a station is selected. The file size has also changed after exiting RCMAIN, and on one occasion placed a B in the busy field.

Temporary solution:
None.

KPR #: 5000239921 Product: HOST SOFTWARE VAX 64882 02.00

Keywords: TRANSFER

One-line description:
Transfer of files over DECnet causes program to crash

Problem:
High Speed Link transfer may fail when file is accessed via DECnet on a remote node. Error messages such as:
Improperly handled exit condition..... are displayed
and the transfer fails to complete.

Temporary solution:
Use DECnet to first copy the file to the node where the high speed link resides, then perform the transfer.

Signed off 02/03/89 in release A02.50

KPR #: D200045096 Product: HOST SOFTWARE VAX 64882 01.20

One-line description:
Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.

Problem:
None of the HP programs react well to the normal VAX terminal control

KPR #: D200045096 **CONTINUED**

commads - CNTRL Z; CNTRL Y; CNTRL C. The programs are not consistant in how they react.

For example if rcmain hangs it is necessary to edit the rcmain file.

The only file to cause real damage was the RCMAIN. I used cntrl Y to exit while connected to the HP. The program left the HP in a busy state that was not cleaned up. A data file had to be edited by hand to correct.

Temporary solution:
None.

KPR #: D200047217 Product: HOST SOFTWARE VAX 64882 01.20

One-line description:
LONG COMMANDS GREATER THAN 1024 CHAR. MALFUNCTION WITH DMF-32 I/O CARD

Problem:
LONG COMMANDS (>1024 CHAR) INPUT TO REMOTE CONTROL CAUSES CORE DUMP OR OR LOSS OF FUNCTIONALITY OF TERMINAL INUSE WHEN 64000 I/O CARD IS DMF-32

KPR #: D200059428 Product: HOST SOFTWARE VAX 64882 01.60

Keywords: RCMAIN

One-line description:
Vax rcdevice file not updated correctly

Problem:
On the VAX, the rcdevice.dat file used by remote control is not being updated correctly when there is no comment at the end of a device entry. It is also affected by the position of the entry in the file. The error is noticed when the entry is the lase entry in the device file.

KPR #: D200059444 Product: HOST SOFTWARE VAX 64882 01.60

Keywords: RCMAIN

One-line description:
VAX remote control dumps when a very long command is entered

Problem:
On the VAX, one of the regression tests for remote control asks you to enter an 11 line command. This causes the remote control session to end in a stack and register dump.

KPR #: D200064055 Product: HOST SOFTWARE VAX 64882 01.70

Keywords: RCMAIN

One-line description:
/DEVICES= does not work with a list of stations.

KPR #: D200064055 **CONTINUED**

Problem:
When RCMAIN is invoked as:
rcmain/dev=(x,y)
where x and y are devices listed in RCDEVICE.DAT file, the program goes directly into interactive mode. When invoked with a single device, the program goes directly into remote control.

Temporary solution:
Use only device lists that consist of one device.

KPR #: D200093591 Product: HOST SOFTWARE VAX 64882 02.40

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

Signed off 02/03/89 in release A02.50

KPR #: D200093625 Product: HOST SOFTWARE VAX 64882 02.40

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-c1[h]

-fh1

-th1

note the -1 option in combination with any option requesting high speed link (-c or -h).

Signed off 02/03/89 in release A02.50

KPR #: 1650033720 Product: HP TEAMWORK 300 64711S004 02.00

One-line description:
Removing models from the index does not delete all its files.

Problem:
When models are removed from the database by selecting them from the model index and selecting "delete" not all files which belong to the model are actually purged from the disc.

Since they can't be accessed from within teamwork anymore they just fill up the disc.

Temporary solution:
Issue a 'dump_twk -d dumpfile -all' command.

'cd $DBPATH/twk_0_files'

'rm dir.*/*'

Return to the dumpfile directory and

'load_twk -d dumpfile -all'

KPR #: D200089342 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
Simultaneous socket connections cause a hang.

Problem:
Simultaneous socket request of the dc_server will cause some requests to be ignored.

Requests are generated by starting HP Teamwork or opening an Index or Diagram.

After a request is "ignored", that session of HP Teamwork can appear to be "hung".

Temporary solution:
None at this time.

KPR #: D200090118 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
DOMAIN -SQRT ERROR generated when Data Flows become tangential to bubble

Problem:
When a process bubble in a data flow diagram is moved to where the data flow is tangential to it, the error:

DOMAIN -SQRT ERROR

is generated from the HP-UX system. In other words, the teamwork window is scrolled up and the error message is generated in the new line. A control L refreshes the screen and no data is lost.

KPR #: D200090118 **CONTINUED**

Temporary solution:
As there is no loss of data, and the screen can be repained, there is no temporary work around.

KPR #: 5000241984 Product: HP TEAMWORK SA M 64710-90903 01.00

One-line description:
Would like where_refer opt to determine which modules call a given mod.

Problem:
CUSTOMER WOULD LIKE A WHERE_REFERENCED OPTION TO DETERMINE WHICH MODULES CALL A GIVEN MODULE. WITH LARGE STRUCTURE CHARTS USING MANY ON/OFF SHEET CONNECTORS IT BECOMES DIFFICULT TO SEE WHO CALLS THE MODULE IN QUESTION.

Temporary solution:
No temporary solution.

KPR #: 5000241976 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
Would like to see load M-SPECS using a command similar to load_dd cmd.

Problem:
CURRENTLY, THE ONLY WAY TO IMPORT M-SPECS TO THE TEAMWORK DATABASE IS USING THE PULL-DOWN MENU IN THE BODY OF THE M-SPEC EDITOR. THIS TYPE OF ENHANCEMENT WOULD BE GREAT FOR CUSTOMERS WITH LARGE PROJECT TEAMS.

Temporary solution:
Load from the pull down menu.

KPR #: D200077636 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
PRINT OBJECTS from the PI doesn't work correctly.

Problem:
If one of the DFD's is open and you return to the PI window, select any number of objects (as long as it included the DFD that is open), PRINT OBJECTS (from the PI window), the follow error message is printed:

The DFD Context-Diagram;0 was not printed for the following reason:
object locked.

(The Context-Diagram was the DFD that was open in this case.)

The DFD was not active. It was partly obscured by the PI window.
It should be 'readable' for printing.

Temporary solution:
None at this time.

KPR #: D200077891 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
Spline is too large for binder.

Problem:
Temporary solution:
no temporary solution.

KPR #: 1650004499 Product: HP-UX 68000/8/10 A M 64845-90905 01.30

One-line description:
Assembler flagging LR error for correct offset when using PC+IND+OFFSET.

Problem:
The following program shows a problem with PC-relative addressing with displacement. The displacement is taken as the low-order 8 bits of the label instead of relative to the current PC.

```
         ORG     0F8H
         MOVE    LABEL[PC,D6],D6
         ORG     102H
LABEL    DC.W    0FFFFH
```

This results in an error message:

LR - Legal Range, Address or displacement out of range of the instruction's addressing capabilitities.

Temporary solution:
No temporary solution.

KPR #: D200045880 Product: HP-UX 68000/8/10 A M 64845-90905 01.30

One-line description:
Wrong offset calculated when using PC+index reg+ offset mode of addr.

Problem:
When using the PC relative with offset and index register mode of addressing the assembler may caluculate the wrong address. The error will be made if the offset symbol is at an absolute location greater than FFH.
"68000"

```
     ORG     010H
     MOVE    #0,D0
     JMP     TABLE[PC,D0]
     ORG     100H
TABLE        DS.W     10
```

Temporary solution:
No temporary solution.

KPR #: D200081836 Product: HP-UX 68000/8/10 A M 64845-90905 01.04

Keywords: MANUAL

One-line description:
Cannot substitute Macro parameter at beginning of variable.

Problem:
The following macro parameter substitution will not work:
"68000"

```
 LABELAA
AALABEL
```


KPR #: D200081836 **CONTINUED**

```
PRO10   MACRO     &P1
        BRA.W     LABEL&P1 ---------> O.K.
        BRA.W     &P1LABEL ---------> DOES NOT WORK, BUT W/ MOTOROLA
        MEND                          ASSEMBLER IT WILL  WORK

        PRO10     AA
```

The manual needs to be changed in section 6, the section dealing with Macros. On pages 6-4, 6-5 we need to explain, that the kind of macro substitution above, will not work. The problem is, the assembler parses the ENTIRE value, &P1LABEL, and cannont find a parameter

---

KPR #: D200064386 Product: HP-UX 68000/8/10 C M 64819-90903 01.40

One-line description:
Byte parameters are pushed onto the stack incorrectly.

Problem:
When passing a byte parameter it is not pushed onto the stack as the manual specifies it will be. The Pascal and C manual specify that a byte parameter will be pushed in the upper byte of the word which is pushed on the stack. The C compiler does a Move.W and pushes the char in the lower byte. The pascal compiler does the push correctly.

```
"C"
"68000"

char called_func();

calling_func() {

  char  passed_parm;
  passed_parm = 'b';

  called_func(passed_parm);
}

char  called_func(parm)
char  parm;
{
  char local_var;
  local_var = parm;
}
```

KPR #: 5000240937 Product: HP-UX 8051 ASSM M 64855-90903 01.40

Keywords: MANUAL

One-line description:
Change 8051 manual page 8-4

Problem:
The 8051 assembler/linker reference manual has paragraph with errors relating to the CSEG directive. The errors are typographical, but can lead to confusion:
Page 8-4 of the HP-UX hosted manual talks about the CSEG directive with a paragraph:

The code segment counters can be charged (changed) with the DS, DW, and DW (DB) pseudos, and with each instruction encoded. Each unit in the program relocatable counter represents one byte in the code address space within the range of 0 to 64. (0 to 64K)

(corrections are indicated in parenthesis)

Temporary solution:
Be aware of these changes when using the 8051.

KPR #: 1650008128 Product: HP-UX 8085 C M 64826-90902 01.50

One-line description:
New and dispose have inconsistient parameters

Problem:
If you call DISPOSE as the manual states on page 4-10 a run time error is flagged.

```
DISPOSE (&pointer, sizeof (*pointer));  /* as in manual. */

DISPOSE (pointer, sizeof (*pointer));   /* this works.   */
```

This would be acceptable, but, NEW is called with the address of a pointer as the manual states. It seems that NEW and DISPOSE should be called in the same manner.

Temporary solution:
Call DISPOSE with a pointer rather than its address.

```
DISPOSE (pointer, sizeof (*pointer));
```

KPR #: D200079574 Product: HP-UX 8086/88 ASSM M 64853-90905 02.20

Keywords: CODE GENERATOR MANUAL

One-line description:
.

Problem:
THE FOLLOWING PROGRAM PRODUCES AN error ET.

"70108"

```
        GLB     HPTEST
        PROG
        EXTERN  SYMBOL
HPTEST  PROC    FAR
        ASSUME  DS0:DATA,PS:PROG
        MOV     AW,SEG SYMBOL
        MOV     DS0,AW
        MOV     AW,SYMBOL
        INC     AW
        MOV     SYMBOL,AW
        RET
```

Temporary solution:
There is no known solution at this time.

KPR #: 5000211359 Product: HP-UX 8086/88 C M 64818-90903 03.02

One-line description:
Additional info about the $SEPARATE_CONST$ directive works, pg. 2-3.

Problem:
This SR consists of 2 complaints. The first requests that the $SEPARATE_CONST ON/OFF$ option default to off, instad of on. This will not be implemented at this time, because of the installed base using our compilers.

The second is a request to change the manual, making it easier to find information on "preparing the program for prom programming". This request has been turned over to the manual writers, and will be addressed soon.

KPR #: 5000188813 Product: HP-UX 8086/88 PAS M 64814-90904 01.01

One-line description:
DOC. FOR THE PASCAL LIB. ERROR HANDLING ROUTINES NEEDS IMPROVEMENT.

Problem:
Documentation for the Pascal Library error handling routines needs improvement. For Example: The manual documents that a routine "CaseError" is called if an unexpected CASE value is encountered. If the user does not supply his own "CaseError" routine, he/she can eventaully determine that our CaseError library routine calls a routine called "Abort", which in turn calls "PASCAL_ERROR". If HP's monitor is linked, PASCAL_ERROR is in the monitor program. If the monitor is not linked, we provide a PASCAL_ERROR routine. This routine never returns control to the calling routine. Neither Abort nor PASCAL_ERROR are documented in the manual.
Please improve the documentation in this area.

Temporary solution:
No known temporary solution at this time.

KPR #: D200079517 Product: HP-UX OP SYS M 64801-90903 01.00

One-line description:
Meas system unuseable if WINDEX exited without ending measurement.

Problem:
If you run a 64000-UX measurement system
feature under WINDEX, and you exit WINDEX
without first doing an "end" or "end release_system"
within the feature, the measurement system in
question becomes hung in the "in-use" state
(even though noone is using it). It cannot be
released without rebooting the HP-UX system.

Temporary solution:
Do an "end" or "end release_system" within the
measurement system before exiting WINDEX.

KPR #: D200090431 Product: HP-UX OP SYS M 64801-90903 01.00

One-line description:
ftio command for hp-ux 6.01 does not function as documented.

Problem:
ftio command for hp-ux 6.01 does not function as documented.
IE. ftio -ocx / complains about not being able to locate something
and eventually terminates abnormally.

This problem has been fixed for hp-ux 6.2.
Also, this SR was entered under the wrong product code.

Temporary solution:
There is no workaround.

KPR #: 5000182824 Product: HP-UX SYSTEM INST M 64880-90901 01.02

One-line description:
DOC SHOULD INCLUDE LIST OF SUPPORTED CARDS FOR RS232 XFER.

Problem:
64000 II rs232 transfer function and the available cards.
Rs232 transfer from the 9000/300 is only supported through the 98628 rs
232 card. It is not supported on the human interface card and there is
no documentation that lets you know that. Two requests.

1. Documentation should include list of supported cards for rs232 transf
er right next to the suggestion for using rs232 transfer for file transf
ering from the 64000 to the 64000 II and near all explanations of rs232
transfer. teh suggestion to transfer all files from old ssytem to new u
sing rs232 is in the instalation and configuration manual.

2. Make the human interface card a supported card. Most people get it b
ecause it seems to be the most versital. it seems to be a waist to have
a card and not be able to use it fully.

Temporary solution:
No temporary solution.

KPR #: 5000269381 Product: HP-UX SYSTEM INST M 64880-90901 01.00

One-line description:
Manual needs to be more explicit about /dev/ttyXX where XX is numeric

---

KPR #: D200068429 Product: HP-UX SYSTEM INST M 64880-90901 01.02

Keywords: HIGH SPEED LINK

One-line description:
Fails to transfer first passworded file, but doesn't notify the user.

Problem:
When transfering a file list to a passworded HP64000 userid,
transfer will ask you for the password. If you misstype the password,
which happens alot because of UPPER/lower case differences, transfer
will display an error message, tell you that the first file is not
transfered, and prompt you again for the password. If you then type
the password correctly, transfer will go ahead and transfer the
remaining files, BUT NOT THE FIRST ONE.

Temporary solution:
None at this time.

KPR #: D200077933 Product: INVERSE ASSEMB 64856 01.01

One-line description:
Can loop forever when a source file contains macros.

Problem:
It is not possible to use macros in the Inverse Assembler source definition.

Temporary solution:
Until this problem is fixed, either avoid using macros, or simply expand all macros that already exist.

KPR #: D200093088 Product: NETWORK TRANSFER 300 64887S004 01.00

One-line description:
The transferII utility does not work using nft as the transport

Problem:
When using hostcopy or get64 and nft as the transport an errror message of hostcopy:untranslateable NFT response, more information: usage: [-p -s -d -L -r -P -F -B -A] fromnode#user#file tonode#user#file abortin g.

KPR #: D200094490 Product: NSC800 EMULATION 64292 01.03

One-line description:
Incorrect memory display between two odd addresses

Problem:
When a display memory command is issued which displays memory between two odd addresses, the last byte is sometimes displayed incorrectly.

KPR #: 1650058925 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Problem with Macro code generation.

Temporary solution:
There are several work arounds:
- Use a subroutine
- Use the following code:

```
     M2    MACRO
           .GOTO   FOO
     FOO   .NOP
     L&&&&  NOP
```

It's hard for us to see the exact use of the MACRO sent, so we could probably supply a better workaround with a more specific example.

KPR #: 2700005769 Product: OPERATING SYSTEM 64100 01.39

Keywords: DC600

One-line description:
DC600 backup hangs up when it encounters a defective tape.

Problem:
Customer attempted DC600 backup from keyboard of master. Backup never completed. Error message appeared very briefly claiming a CRC check failure. Then system attempts to restart backup procedure. Situation continues ad nauseum.

Bill Furch has complete details and customer's tape.

Temporary solution:
no temporary solution

KPR #: D200015297 Product: OPERATING SYSTEM 64100 02.00

One-line description:
CDC FLOPPY DRIVE DOESN'T FORMAT CORRECTLY IN A COMMAND FILE.

Problem:
CDC DRIVES DON'T FORMAT CORRECTLY WHEN INITIATED FROM A COMMAND FILE.

Temporary solution:
no temporary solution.

KPR #: D200041178 Product: OPERATING SYSTEM 64100 02.02

One-line description:
Nested macro calls cause incorrect macro expansion.

Problem:
The following code will assemble with no errors on the 64000, but the macro expansion is incorrect. When a macro calls another macro, the expansion includes the name of the macro being called on a separate line, then includes the body of the macro itself.

KPR #: D200041178 **CONTINUED**

```
"processor name"
DUMMY       MACRO
            NOP
            MEND
MAKR01      MACRO
            DUMMY
            MEND
VAR1        EQU      40H
MAKR02      MACRO
            MAKR01
NULL        .SET      0
            DB       VAR1
            MEND
            MAKR01
            MAKR02

The call to MAKR01 will expand as follows:
+           DUMMY
+           NOP

The call to MAKR02 will expand as follows:
+           MAKR01
+           DUMMY
+           NOP
+           DB      VAR1
```

Temporary solution:
No temporary solution at this time.

KPR #: D200069989 Product: OPERATING SYSTEM 64100 02.06

Keywords: DC600

One-line description:
store to DC600 causes 64000 to reboot.

Problem:
Store to dc600 causes 64000 to reboot

KPR #: D200084897 Product: OPERATING SYSTEM 64100 02.07

One-line description:
Recover cmd on 64000(PISCES I) will recover all types on disc's > 150Mb

Problem:
The recover command fails on large disc drives (greater than 150 Mbytes) if no file type is given. This syntax is normally used to recover all possible types of a purged file.

This failure shows in a station hang.

See the Lab Text for more information.
Detailed Listing for Defect Number LSDqf01747
Text:

KPR #: D200084897 **CONTINUED**

recover cmd on 64000(piscesI) will recovering all types on disc's >150 Mb

Large disc drives (greater than 150 Mbytes) will show a defect in the 64000 operating system when a user trys to recover all possible types of a purged file.

Failure syntax:

recover X

Working syntax:

recover X:source

Defect apparently is in the file manager, and is related to discs with 32 sectors per page as formatted for large disc drives. The failure shows in a station hang. The station will hang AFTER successfully compl eting
any recovery. The work around, is to specify any file(s) with specific types.

Temporary solution:
The temporary solution to this problem is to specify each file type individually when recovering purged files.

KPR #: D200086694 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Macro use of a label is missing from xref.

Problem:
Another difference between 64000 and 64000-UX assembler found.
The label "LABEL" is missing from the cross reference.

```
      1   "processor"
       2   RMB    MACRO &P1
       3   VALUE .SET   &P1
       4          AND    VALUE
       5          MEND
       6
       7   LABEL EQU    0          <- DEFINITION
       8          RMB    LABEL      <- REFERENCE
The X-REF table of this program is
      LINE# SYMBOL     TYPE      REFERENCES
          7 LABEL      A                 <- This field should be 8.
      ***   VALUE      U                 <-                      4.
```

Temporary solution:
No temporary solution.

KPR #: 5000240952 Product: PROM PROGRAMMER 300 64501S004 01.30

One-line description:
PROM programmer has problems in UX envr programming 32 bit system.

Problem:
File expansion to 32 bits causes byte to be dropped.

KPR #: 5000231571 Product: ROM EMULATION 64272 01.04

One-line description:
store command generates 16-bit width absolute file only

Problem:
The store memory command in the 64272 ROM emulator creates an absolute file specifying a bus width of 16 bits, regardless of the emulation configuration. This makes programming EPROMs for systems with 8 bit wide data buses difficult.
The emulator should create the file with the bus width (8 or 16) specified in the emulation configuration.

Temporary solution:
No workaround at this time.

KPR #: D200093906 Product: RS-232 TRANSFER 300 64885 01.30

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

Signed off 02/03/89 in release A01.50

KPR #: D200093930 Product: RS-232 TRANSFER 300 64885 01.30

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

Signed off 02/03/89 in release A01.50

KPR #: D200065219 Product: RS-232 TRANSFER 500 64884 01.10

Keywords: TRANSFER

One-line description:
Transfer hangs after bad options message is displayed.

Problem:
Transfer hangs after bad options message is displayed

KPR #: D200093898 Product: RS-232 TRANSFER 500 64884 01.40

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

Signed off 02/03/89 in release A01.50

KPR #: D200093922 Product: RS-232 TRANSFER 500 64884 01.40

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

Signed off 02/03/89 in release A01.50

KPR #: 5000194951 Product: RS-232 TRANSFER VAX 64886 01.10

Keywords: TRANSFER

One-line description:
Inaccurate specification in HELP for TRANSFER command

Problem:
RS232 transfer on VAX does not work as specified in the VMS help pages. The command syntax shown for a filelist transfer is:

```
    TRANSFER /OPTION(S) <filelist> [<HP64000_file_spec>]
```

which implies that the 64000 filespec is optional. But, if that is omitted, an error message:

```
     ERROR BAD PARAMETERS - NO DESTINATION SPEC
```

is given.

This second parameter should be optional.

Signed off 02/03/89 in release A01.70

KPR #: 5000211375 Product: SOFTKEY EDITOR 64790-90901 01.00

One-line description:
The find command does not work correctly, cannot find string includ '$'.

Problem:
In sk editor mode and when I use command file as follows,the find command does not work correctly.

```
   find "$PAGE$"  all   ----> this change    find "$"  all
```

This means we can not find the string including '$' when using command file.

Temporary solution:
No temporary solution.

KPR #: D200089896 Product: SOFTKEY EDITOR 64790-90901 01.00

One-line description:
AND '$' NEEDS TO BE ESCAPED ON COMMAND LINE TO PREVENT SHELL EXPANSION

Problem:
Text:
any '$' needs to be escaped on command line to prevent shell expansion

If the user wishes to have a command that includes a '$' anywhere in the command, then the user must put a backslash before that '$'. Otherwise, the softkey package does shell variable expansion, which will probably cause the '$' and any text following it to be replaced with either a null string, or whatever the text matches in the shell.

ie. echo $hello ---> echo \$hello

KPR #: 1650061580 Product: SOFTKEY EDITOR 300 64790S004 02.00

One-line description:
When 4 retrieves are done, the sk editor jumps to shell.

Problem:
The following command sequence forces sk back to the shell:

Make an empty file with only 3069 line feeds (insert line, repeat 3068).
Edit the file, copy one line
retrieve
retrieve
retrieve
retrieve <<< return to shell
$
Retrieve 4 after 3069 lines doesn't cause any problems , just this sequence

Also, the following error message is printed:
"Malloc error, unable to continue. Use SK preserve"

Temporary solution:
This problem seems to occur only under these exact conditions. Therefore, the chance of having this problem is minimal. It can, however, be avoided by using "retrieve 4" instead of 4 sequential retrieve commands.

KPR #: 5000401349 Product: SOFTKEY EDITOR 300 64790S004 02.10

One-line description:
Status line does not change after file is written for the save command.

Problem:
The SK editor does not update the status line after it completes storing a file caused by a save command. Example:

- if you issue a "save command, the status line will read "writing /users/joe/file" until another key is pressed.

This does not effect the operation of the editor itself.

KPR #: 5000401372 Product: SOFTKEY EDITOR 300 64790S004 02.10

One-line description:
sk editor replace command does not work properly with anystring (*).

Problem:
The sk editor does not properly handle replace commands involving anystring (*) and a limiting range.
Example:
The current line contains the letters aa in columns one and two and the range is set to one. The command "replace ^* ^ with ^*!^" should change the string to "aa!", instead it changes the line to "aa!!....".

KPR #: D200090241 Product: SOFTKEY EDITOR 300 64790S004 02.10

One-line description:
When retrieving enough lines to get file exactly 1024 in size; core dump

Problem:
Detailed Listing for Defect Number LSDqf03643
Text:
when retrieving enough lines to get file exactly 1024 in size, core dump

A core dump occurs in the following situation:

enter sk without any filename.
enter a single line in INSERT mode
Get out of INSERT mode.
retrieve the line 1024 times.
sk goes into infinite loop or core dumps.

There may also be problems with merging files with exactly 1024 lines.

KPR #: D200075028 Product: STATE ANALYZER 64620 00.71

One-line description:
Source referencing will not work with non-zero segments (8086, etc)

KPR #: D200091538 Product: STATE ANALYZER 300 64620S004 01.10

One-line description:
File names <8 chars in link_sym will cause translate problems

Problem:
File names in the linker symbol file which have less than 8 characters cause a problem in the translation routines of translate(1) and transfer(1). If this occurs, the user will see several disc transfer error reports from transfer(1) because the malloc(3) buffer space gets trashed in the translation.

Temporary solution:
Be careful to use file names which have at least 8 characers.

KPR #: 5000122374 Product: SW PERF ANALYZER 64310 01.11

One-line description:
"show curr_meas" after measurement change crashes station.

Temporary solution:
Do not show the current measurement after you have changed the measurement setup. This is not very inconvenient, since the user is about to take a new measurement anyway; and presumably knows the results of his last measurement.

KPR #: D200080176 Product: SW PERF ANALYZER 300 64310S004 01.20

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200081026 Product: SW PERF ANALYZER 300 64310S004 01.20

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200082347 Product: SW PERF ANALYZER 300 64310S004 01.20

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can

KPR #: D200082347 **CONTINUED**

release the processes by
cat < ptyxx
This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082370 Product: TIMING ANALYZER 300 64610S004 01.60

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by
cat < ptyxx
This causes the pending output to be flushed, and the processes will die naturally.

KPR #: 5000089359 Product: TIMING/STATE 64610 01.00

One-line description:
label cannot be deleted in trigger specification

Temporary solution:
Just change the trigger specification to "trigger on anything"
and then delete the label.
****
In order to remove a label from a trigger in this set of circumstances, punch in "trigger on anything" before "trigger received". This will clear the trigger of any labels, and the timing analyzer will then receive the trigger correctly.
****

Duplicate Service Requests: 5000122770

KPR #: D200090522 Product: TMS320C25 EMUL FW 64787 00.01

One-line description:
Data words at address 6 & 7 can apparantly be displayed and modified

Problem:
Detailed Listing for Defect Number LSDqf03720

Text:
Data words at address 6 & 7 can apparently be displayed & modified

Temporary solution:
There is no known workaround.

KPR #: D200035261 Product: UPROG 64276 01.00

One-line description:
IN UP_CNTL,"LIST TRACEDATA" SHOWS "AND" EVEN IF NO "ABSOLUTE_IS"

Problem:
In uprog_control context, if list tracedata <LABEL> relative_to <MAP> entered, the softkeys will display and as a valid option (as in and segments or and symbols) even if no asmb_db has been specified with t absolute_is command.

The and segments and and symbols options are only valid if a asmb_db has been specified. Therefore, the and should be suppressed if no asmb_db has been specified.

Temporary solution:
Workaround:
Ignore and option, end command with <RETURN> after specifying map.

KPR #: D200035287 Product: UPROG 64276 01.00

One-line description:
IN UP_CNTL, NO ERRMSG ISSUED IF "RUN UNTIL W/JAM ATTEMPTED W/O JAM LABEL

Problem:
In uprog_control context:
If (1) run until issued without any jam at start;
and (2) on_break action is to jam (rather than stop clocks);
and (3) no default jam label is in effect
then error message No default jam label in effect should be issued because instrument has no way of determining width of jam desired.

Circumstances causing this defect are very rare because a default jam label is always defined unless the user enters format specification context, deletes the current default jam label, and enters more than one alternative jam labels.

Temporary solution:
Workaround:

Define a default jam label by issuing a run command with some sort of ja start.

KPR #: 5000251322 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Expand Directive not working on 64000.

Problem:
The expand directive is not working in the 64000.

"processor"

```
FOXTROT     MACRO
            LD      B,H
            MEND

            EXPAND
            FOXTROT
            END
```

No expansion is done.

Temporary solution:
Specify expand on the command line rather than in your source.

KPR #: D200068924 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Duplicate Symbols in Symbols Declarations not flagged as an error.

Problem:
Duplicate symbols are not flagged as errors in the UDA definition source.

Example:

```
SYMBOLS = NUMBERS
  ZERO = 0
  ONE  = 1
END

SYMBOLS = TWO_SYM
  ZERO  = 0         <<<Should be flagged as
  ONE   = 1         <<<duplicate symbols
END
```

Temporary solution:
Do not enter duplicate symbol names. Each symbol must have a unique name.

KPR #: D200068932 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Duplicate SYMBOLS Definitions are not flagged as an error

Problem:
Duplicate symbol type names not flagged as an error. Very misleading to user who thinks code assembled correctly.

KPR #: D200068932 **CONTINUED**

Example:

```
SYMBOLS = DSEL_SYM
   INT = 0
   EXT = 1
END

SYMBOLS = DSEL_SYM      <<<Duplicate symbol type
   INT = 0                 declaration should be
   EXT = 1                 flagged as an error
END
```

Temporary solution:
Do not define duplicate symbol types.

KPR #: D200068940 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Bad table code generated when more than 25 SYMBOLS definitions

Problem:
Only 25 user-defined symbol types allowed. Any more than that are not flagged as an error. Instead, bad code is generated. The last valid SYMBOLS declaration can be detected in the UDA :listing file as follows.

```
00FF  source_line#a   SYMBOLS = NAME1   <<<Last valid Symbols
                                           declaration has 00FF

010F  source_line#b   SYMBOLS = NAME2   <<<First invalid Symbols
                                           declaration has 010F
```

Temporary solution:
Do not define more than 25 User defined symbols types.

KPR #: D200079376 Product: USER DEF ASSEMB 64851 00.70

Keywords: CODE GENERATOR

One-line description:
High order bits stripped from source characters in Pisces I

Problem:
High order bits stripped from source characters in Pisces I.

Temporary solution:
There is no known fix at this time.

KPR #: D200089409 Product: USER DEF ASSEMB 64851 00.70

One-line description:
REPT will only take arguments range 1 thru 32767

KPR #: D200089433 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Page size is different between PI and Hosted assemblers

KPR #: D200089458 Product: USER DEF ASSEMB 64851 00.70

One-line description:
line number only 16-bits in size... This is too small for long files.

KPR #: D200092619 Product: USER DEF ASSEMB 64851 00.70

One-line description:
COPY :asmb_sym to display behaves like disc_image on.

Problem:
Please contact Caren Johnson x5714 for the supporting software:

After assembling the given UDA code, the two given sample programs display the problem.
The output from

```
   copy file:asmb_sym to display
```

uses half disc_image and half normal output.

Example after assembling sample code, and using "copy file:asmb_sym to display" command:

```
    Record # 1   size = 121
    Asmb_sym record:
    8053  594D  5F50  4F52  5431  0000 .........
    5F50  4F52  5430  ...........................
    5433  0000  ..................................   <-- DISC
    ..........................................           IMAGE
    ..........................................


    Record # 2   size = 102
    Asmb_sym record:
    SYM_PMGB      0000H Absolute      SYM_PMGA    0000H Absolute
    SYM_BSB2      0000H Absolute      SYM_IPS     0000H Absolute
    SYM_SIO       0000H Absolute      SYM_BSB1    0000H Absolute
    ..........................................................
    ..........................................................
                          ^
                          |
                          |
                       NORMAL
```

Temporary solution:

KPR #: D200092619 **CONTINUED**

For some reason if the two statements:

```
SYM_PORT10      SET     PORT10
SYM_PORT11      SET     PORT11
```

are added to the sample programs the problem does not occur.

KPR #: D200094078 Product: USER DEF ASSEMB 64851 00.70

Keywords: CODE GENERATOR

One-line description:
Error Message ", errors= " will appear on listing.

Problem:
The message ",errors= 0" appears on listing.

KPR #: 5000294181 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
Undefined Error placed on all macro usage, if just one label undefined

Problem:
The given code does indeed flag P1, P2, and P3 as undefined even though P2 is the only label that is undefined. The SR was submitted for Hotsite Epic #1487. The current version in QA however seems to have fixed this problem while only flagging P2. This pre-released version has been sent to the customer.

KPR #: 5000409102 Product: USER DEF ASSEMB 300 64851S004 02.20

One-line description:
UDA produces "core dump" when external is used.

Problem:
The customer is complaing that when externals are used which were never declared as global, a "core dump" occurs. However, I found that a "core dump" occurs even when the external is declared as global.

The supporting source files were sent to Dave Ritchie (two test files, assembler source, and linker source).

KPR #: 5000417790 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
ORG 10000H will change address of next line, but not those following.

Problem:
Example:

```
"Z8001"
        ORG   10000H
SYMA    NOP                 ‹10000›
SYMB    NOP                 ‹00002›
SYMC    NOP                 ‹00004›
```

The ORG will set SYMA at absolute address 10000H
but, SYMB will be placed at 00002H and SYMC at 00004H.
If the ORG is to address 100H, the correct address are assigned (100H, 102H, 104H).

KPR #: 5000419440 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
Linker ERROR messages go only to standard error.

Problem:
Assemble the following modules and link:

```
--------------CALL.S-----------------------
"8051"
      EXTERNAL  SUBR
```

KPR #: 5000419440 **CONTINUED**

```
      EXTERNAL  SUBR1

      ACALL     SUBR
      END
```

----------CALL1.S----------

```
"8051"
      GLB  SUBR
SUBR  NOP
      NOP
      END
```

----------CALL.K----------

```
object files     call.R
load addresses   0,0,0
object files     call1.R
load addresses   1000H,0,0
```

The above will write the "WARNING: Pass 2 Undefined Symbol" to standared Error and standard Output. However, the "ERROR: Address out of range" message will go only to standard Error.

Temporary solution:
To redirect both standard Error and standard Output to the same file:

```
Bourne Shell:   lnk -ox call.K › call.map 2›&1
C Shell :       lnk -ox call.K ›& call.map
```

KPR #: D200087569 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, destroying a pointer to the phase error occurance chain in the MS-DOS version.

Temporary solution:
Do not use forward referencs.

KPR #: D200094565 Product: USER DEF ASSEMB 300 64851S004 02.20

One-line description:
Tabs converted to spaces in assembly code.

Problem:
Tabs used within quotes generate incorrect code - the tabs are expanded into spaces.

KPR #: D200094565 **CONTINUED**

```
                        "80188"

0000   202074776F             DB      "            two tabs"
0005   2074616273
000A   202074776F             DB      '            two tabs'
000F   2074616273
                        END
```

The Tabs were obviously converted into spaces (20). Earlier revisions of the series 500 assembler did not convert these tabs into spaces, but rather generated the correct 09 instead of 20. The reason this is a problem is the customer would like to compare the series 300 relocatables with the series 500 relocatables - but the tabs and spaces are causing differences.

Temporary solution:
We are going to provide the customer a checksum routine for the various hosts.

KPR #: D200065391 Product: USER DEF ASSEMB 500 64851S001 01.50

One-line description:
Conditional assembly for INCLUDE files causes error.

Problem:
The following program has a conditional which causes one of two files to be included. However, if the file that is not included by the conditional is missing, an error is generated when the program is assembled:

```
"processor name"
        DATA
X       RMB             0
        PROG
        IF              X = 0
        INCLUDE         julie1
        ELSE
        INCLUDE         julie2
                        ^ IO - invalid operand error if julie 2 is
                               missing
        ENDIF
```

Temporary solution:
No known workaround at this time.

KPR #: D200087544 Product: USER DEF ASSEMB 500 64851S001 02.10

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, which trashed a pointer to the phase error occurance chain in the MS-DOS version.

KPR #: D200093781 Product: USER DEF ASSEMB 500 64851S001 02.20

Keywords: CODE GENERATOR

One-line description:
Problem with parameter passing in macros

Problem:
The following program causes the errors shown:

```
 "Z80"
 AAA:   MACRO   &A, &B

   AAA  abc, def  ghi
+  ;abc
+  ;def  ghi       The ghi is a comment here, not part of the parameter


 "Z80"
 AAA:   MACRO   &A,&B,&C
```


KPR #: D200093781 **CONTINUED**

```
        ;&A
        ;&B
        ;&C

        AAA         {,+,}
        ;{
        ;=
        ;=,}        This is only a problem with the "}". Other
                    special characters do not cause this problem.
```

Temporary solution:
There is no known work around at this time.

KPR #: D200095075 Product: USER DEF ASSEMB 500 64851S001 02.20

One-line description:
Not all Parameters in linker config. table read.

Problem:
LINKER ON HOSTED ASSEMBLERS -

With the addition of the additional address prompt for the A5 register on the 68000 family the number of valid constants in the linker configuration table was increased; however, the existing tables were not updated to increase the number of valid constants and place the value of zero (0) in the constant field to indicate that they did not need the additional prompt. This resulted in a 'smiley face' prompt on some PC linkers following the COMN prompt in the addresses field.

.fix

The hosted linkers only allocate enough room for the number of valid constants specified, and, only read that many constants. The constant table can be a maximum of 32 words long (and room must be left for 32 constants). The current hosted linker procedure for reading the constant table is this:

- Read in number of valid constants 'n' (Usually 16)
- Allocate enough room for the constants
- Read 'n' constants

Since the additional prompt is specified in constant 17, the results are unpredictable. The 17th constant contains the string length of the prompt, if it is zero, no prompt is printed. The contents of the 17th constant are supject to the whims of the O.S. allocation routines and runtime system.

To remedy this situation, I am implementing the following procedure for reading the constant tables:

- Allocate space for maximum number of constants (32)

KPR #: D200095075 **CONTINUED**

- o Zero all constants
- o Read in number of valid constants 'n'
- o Read 'n' constants (Constants n+1 through 32 will be zero)

KPR #: D200091272 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
Can not assemble a file on a different disk...(ie: 'A:' from C:)

Problem:
Detailed Listing for Defect Number LSDqf03929
Text:
Can't assemble file on different disk (eg: 'a:') from c:

Found a problem with assembler on DOS where I could not assemble a file on a different disk than the current disk because assembler builds wrong path name for source file.

Example:
current directory is c:\junk and we want to assemble file 'fred' on drive a:

```
C:\JUNK> asm a:fred
```

Assembler cannot open object file because it builds a path for the source file name of: 'C:\JUNK\A:FRED'. This is wrong.

This bug was fixed in the file pathgen.c and will be resolved in the next SUDS release (Jul 88).

This report is for informational purposes only. Please enter in STARS and resolve with code change. Thanks.

KPR #: D200091314 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
Assembler crashes when directory path name is too long

KPR #: D200065409 Product: USER DEF ASSEMB VAX 64851S003 01.50

One-line description:
Conditional assembly for INCLUDE files causes error.

Problem:
The following program has a conditional which causes one of two files to be included. However, if the file that is not included by the conditional is missing, an error is generated when the program is assembled:

```
"processor name"
        DATA
X       RMB             0
        PROG
        IF              X = 0
        INCLUDE         julie1
        ELSE
        INCLUDE         julie2
                        ^ I0 - invalid operand error if julie 2 is
                               missing
        ENDIF
```

Temporary solution:
No known workaround at this time.

KPR #: D200087551 Product: USER DEF ASSEMB VAX 64851S003 02.10

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, which destroyed a pointer to the phase error occurance chain in the MS-DOS version.

Temporary solution:
Do not use forward references.

KPR #: D200094656 Product: USER DEF EMUL 300 64274S004 01.20

One-line description:
64000-UX "configude" executes utils using $PATH, not $HP64000

KPR #: D200095760 Product: USER DEF EMUL 300 64274S004 01.20

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200096057 Product: USER DEF EMUL 300 64274S004 01.20

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: D200096206 Product: USER DEF EMUL 300 64274S004 01.20

One-line description:
DEFAULT_STATUS can not be set to a combination of 0's and X's.

Problem:
The default status field of the UDE configuration file will treat a pattern consisting of 0's and don't cares as all don't cares. Only default status fields with at least one bit specified as 1 will be correctly recognized.

KPR #: 5000153981 Product: USER DEFIN ASM M 64851-90904 01.00

One-line description:
DE must be defined before being referenced.

Problem:
Assembler manual has apparently ambiguous error definition.

Specifically

DE - Indicated symbol must be defined prior to it being referenced.
Symbol may be defined later in program sequence.

The 64000 and 500 seem to agree with the later statement. The series 300 thinks the first one is true.

Temporary solution:
No temporary solution.

KPR #: D200079558 Product: USER DEFIN ASM M 64851-90904 00.70

One-line description:
64000 station resets when linking if SKELETON command used improperly

Problem:
When the SKELETON command is used improperly in the linker, the 64000 station behaves erratically. Symptoms range from resetting of the station to garbage characters displayed when linking the target code.

Temporary solution:
The SKELETON command should only be used in the linker definition source when a GEN_CODE <RELOC_FMT>,BOTH is used in the assembler definition source. The keyword here is BOTH. If VALUE is used, the SKELETON command in the linker should not be used. A note to this effect will be added to the next revision of the manuals.

KPR #: 5000267468 Product: USER INTERFACE M 64808-90901 01.00

One-line description:
Need to add Note saying that 64100 Terminal Mode is not supported.

KPR #: 5000291427 Product: USER INTERFACE M 64808-90901 01.00

One-line description:
PMON doesn't allow a file to begin with a numeric value.

Problem:
When in PMON on a hpux 9000 s/300, and you try to access a file (editor or emulation), you get an error cannot access file. This occurs if the file begins with a numeric character. If you try to use the same editor on a similar named file while in unix (outside of PMON), you can access the file ok. This was a limitation in the old 64100 system, but is not congruent with the unix naming convention. You can get around this problem if you place an * for the first character followed by a numeric character.

Temporary solution:
You can get around this by placing a '*' for the first character followed by the numeric character.

KPR #: 5000413161 Product: USER INTERFACE M 64808-90901 01.00

One-line description:
Two points that need to be indicated on pg 3-5 of 64808-90901 manual.

KPR #: D200090613 Product: USER INTERFACE 300 64808S004 02.10

One-line description:
A command file containing these three characters in that order #'! fails

Problem:
Detailed Listing for Defect Number LSDqf03870
Text:
command file containing #'! (3 characters, in that order) fails

A command file containing the characters

#'!

will fail, leaving the command line garbled; subsequent commands in the command file are not executed. The line may have other characters in it, for example

#comment - this isn't my idea!

which will produce the same results.
The defect was noticed while running regression tests for ptui z80.

Temporary solution:
Be careful not to use that particular combination of characters in a comment line.

KPR #: D200092502 Product: UTILITIES PKG 500 64888S001 01.20

One-line description:
A HP-UX directory can be destroyed by transfer (1)

Problem:
Detailed Listing for Defect Number LSDqf04212
Text:
a HP-UX directory can be destroyed by transfer(1)

attempting to transfer a file from a 64000 cluster to a HP-UX directory name can destroy the HP-UX directory.

ie:

```
transfer -fhas WELCOME::source hpux_directory
can destroy hpux_directory.
```

Detailed Listing for Defect Number LSDqf04212

KPR #: 5000220764 Product: VMS 68000/8/10 ASM M 64845-90906 01.60

Keywords: MANUAL

One-line description:
Manual explains linker options incorrectly.

Problem:
In all "Using the linker" chapters for VAX hosted manuals the *options definitions are both incorrect and misleading. The /map option states that a load map listing is produced. This is done by default when /output is specified. In fact you cannot specify /map without /output so rather than explaining /map we should explain that /nomap can be used to supress a linker map when /output is specified. Secondly, .LIS is not the default extension as stated in the /output definition. filename.MAP is the default extension for the linker. filename.LIS is used by the assembler and compilers.

* Linker options are typically described on page 3-3 of the Using the linker chapter.

Temporary solution:
No temporary solution.

KPR #: 5000220772 Product: VMS 68000/8/10 ASM M 64845-90906 01.60

Keywords: MANUAL

One-line description:
Manual states incorrectly that EXT is a pseudo op.

Problem:
Antoinette Burkett sc:

On page 5-11 of the 68000 cross assembler/linker manual for the vax it states that the either the EXT or EXTERNAL will work as the operator for the external pseudo op, however only EXTERNAL works. EXT generates an error message.

Temporary solution:
Use EXTERNAL in replace of EXT. EXT is an assembler instruction for the 68000 family.

KPR #: D200046268 Product: VMS 68000/8/10 ASM M 64845-90906 01.30

One-line description:
LR error flagged for correct offset using PC+INDEX+OFFSET mode of addr.

Problem:
The following program shows a problem with PC-relative addressing with displacement. The displacement is taken as the low-order 8 bits of the label instead of relative to the current PC.

KPR #: D200046268 **CONTINUED**

```
        ORG     0F8H
        MOVE    LABEL[PC,D6],D6
        ORG     102H
LABEL   DC.W    0FFFFH
```

This results in an error message:

LR - Legal Range, Address or displacement out of range of the instruction's addressing capabilitities.

Temporary solution:
Temporary solution:

"68000"

```
        ORG     0FFH
        MOVE    TABLE-($+2)[PC,D0],D1
TABLE   DS      1
```

Duplicate Service Requests: 5000116046 D200045898 5000160754 5000163576 5000270629

KPR #: 5000238543 Product: VMS FILE FORMATS M 64882-90903 01.02

One-line description:
VAX file format manual doesn't give clear explantion of VAX file types.

KPR #: D200055202 Product: VMS SYSTEM INSTAL M 64882-90904 01.03

One-line description:
Need setting for rear panel of old HP 64000 and 64110(with jumper jacks)

Problem:
Apendix B should also include the setting for the rear panel of an old 64100, one with jumper packs. Also an old 64110, one with jumper packs.

Temporary solution:
This information will be added to future revision of the manual (it has never been documented even in the older manuls).

KPR #: 5000222489 Product: VMS USERS GUIDE M 64882-90902 01.60

One-line description:
Page 3-9 states vt52 emulation using 64100 but does not perform functs.

Temporary solution:
Page 3-9, add Note: VT52 terminal emulation with EDT Does Not apply for VMS 4.0 and higher.

KPR #: D200045492 Product: VMS USERS GUIDE M 64882-90902 01.01

One-line description:
Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.

Problem:
None of the HP programs react well to the normal VAX terminal control commands - CNTRL Z; CNTRL Y; CNTRL C. The programs are not consistant in how they react.

For example if rcmain hangs it is necessary to edit the rcmain file.

The only file to cause real damage was the RCMAIN. I used cntrl Y to exit while connected to the HP. The program left the HP in a busy state that was not cleaned up. A data file had to be edited by hand to correct.

Temporary solution:
No temporary solution.

KPR #: 2700005918 Product: Z8 ASSEMB 64850 00.00

One-line description:
Assembler not generating error message when attempt to load label.

Problem:
When an attempt to load a label, which was previously defined as a constant using the EQU pseudo op, is made no error message is generated.

Temporary solution:
No temporary solution at this time.

KPR #: D200091645 Product: Z8 ASSEMB 64850 00.01

One-line description:
Assembler generates Phase Error of forward referenced EQU

Problem:
The assembler generates a Phase Error on forward referenced EQU instruction. The following code is an example program that will produce the phase error:

"Z8"

```
        INC  FRED
FRED    EQU  R15              ;R15 is a register symbol
```

A phase error should not be the only error produced.

Temporary solution:
There is no workaround available.

KPR #: 1650069773 Product: Z80 FUI DOS 64753S006 01.01

One-line description:
The terminal window escape sequence is not available on German keyboards

Problem:
"CONTROL BACKSLASH" as an escape from the terminal window in the user interface is not reachable via a german keyboard !
Try to supply an alternate exit.

KPR #: 1650069765 Product: Z80 ASSEMB 64842 01.90

One-line description:
"/" character does not work as delimiter for lnk options

Problem:
The command syntax lnk /o filename.K does not work on the PC.
The command syntax lnk -o filename.K does work but is not mentioned in the manual. The "/" cannot be used as a delimiter for the lnk command options.

KPR #: 5000139535 Product: Z80 ASSEMB 64842 01.12

One-line description:
Using HEX psuedo is causing bad address calculations.

Problem:
"Z80"

```
TEST1           HEX     16,1A,0E,16
TEST2           HEX     16,1A,0E,00
TEST3           NOP
                NOP
                JP      TEST3
DEMO            DEFW    TEST1
                DEFW    TEST2
                DEFW    TEST3           ;ADDR IS CALCULATED AS 006H
                                        ;WHEN IT SHOULD BE 0008H
```

Temporary solution:
Use the DEFW psuedo instead of the HEX psuedo.

KPR #: 5000152819 Product: Z80 ASSEMB 64842 01.11

One-line description:
Revision number on output listing is incorrect.

Problem:
The revision number printed on the output listing file is incorrect. It is always 1.10 for the Z80 assembler. The correct revision for this product on the 64000 is 1.10; on the 9000 series 500 it is 1.30; on the 9000 series 300 it is 1.00. The revision does not appear on an output listing produced on the VAX.

KPR #: 5000239939 Product: Z80 ASSEMB 64842 01.12

One-line description:
Xref lists symbols which are under False conditional assembly blocks.

Problem:
Using "IF true" "IF failure" instruction on the 64100A system, the Z-80 assembler outputs x-references table from failure routine.

64100AF Rev2.07 64842AF Rev1.12

KPR #: 5000239939 **CONTINUED**

For example

```
                1 "Z80"                 |  11         IF        FALSE
          <0001> 2 TRUE     EQU     1   |  12 DATA    EQU       1
          <0000> 3 FALSE    EQU     0   |  13         LD        A,DATA
                 4 ;                    |  14         IFEND
                 5         ORG     0    |  15 ;
                 6         IF      TRUE |  16         END
          <0000> 7 DATA     EQU     0   |
0000 3E00        8         LD      A,DATA| LINE# SYMBOL TYPE REFERENCES
                 9         IFEND        |  12  DATA     A  8,13
                10 ;                    |   3  FALSE    A  11
                                        |   2  TRUE     A  6
```

The symbol "DATA" is determined at LINE 7 and used only LINE 8.

Temporary solution:
No temporary solution at this time.

Duplicate Service Requests: 5000242495

KPR #: 5000264986 Product: Z80 ASSEMB 64842 01.10

One-line description:
Complex macro interaction causing invalid errors.

Problem:
A sample program with a complex macro calling scheme is causing the assembler to generate invalid errors.

Temporary solution:
No temporary solution at this time.

KPR #: D200086686 Product: Z80 ASSEMB 64842 01.12

One-line description:
Difference between 64000 and host in XREF when no symbols.

Problem:
The cross reference tables differ between the 9000 host and pisces I. If there are no symbols the 9000 will generate a header, but, the 64000 will not.

Temporary solution:
No temporary solution at this time.

KPR #: 1650047167 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
HPIB 64120 I/O AND POWER FAILED WHEN MODIFYING TARGET MEMORY

Problem:
VERIFIED THE PROBLEM AS DESCRIBED EXCEPT THE HP-IB 64120 I/O AND POWER FAILURE MESSAGE CAN OCCUR REGARDLESS OF ADDRESS RANGE MODIFIED IN TARGET MEMORY

KPR #: 1650047340 Product: Z80 EMULATION 300 64252S004 00.00

One-line description:
EMULATION SOFTWARE STATUS DOES NOT RECOGNIZE THE "HALT" INSTRUCTION

KPR #: D200069542 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200080655 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080952 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081489 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
PC contents lost over continuation if in break state

Problem:
8085 (64203) and old z80 (64252) emulators do not save the PC value over continuation. This only applies if the emulator is in the break state when ending out of the emulator. On continuation the PC is set to zero. Thus the first run command will start executing at zero if no address is specified, and a registers display will list the Next PC as zero.

KPR #: D200081901 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082230 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083246 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085332 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
IMPROPER IDENTIFICATION OF THE SECOND Z80 CONTRL CARD IF TWO Z80 PRESENT

KPR #: D200086033 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

1. display trace absolute symbols on
2. end ; end locks the emulation session
3. <system name> <module name> ; continues the emulation session
4. display trace

KPR #: D200086033 **CONTINUED**

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200090860 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200095711 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200096008 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

KPR #: 1650041624 Product: Z80 PASCAL 64812 00.00

Keywords: CODE GENERATOR

One-line description:
$ORG directive can cause incorrect code to be generated.

Problem:
Z80 Pascal compiler generates wrong code when option $ORG $ is switched on.

```
"PASCAL"
"BZ80"
$ASM_FILE$
PROGRAM walesch;
TYPE
  AUF1 = RECORD
    NUMBER  : SIGNED_16;
    DATA_NO : SIGNED_16;
  END;
  AUF = ARRAY[1..100] OF AUF1;
VAR
  $ORG 1000H$
  AUF_INDEX :AUF;
  $END_ORG$
BEGIN
  AUF_INDEX[1].NUMBER := 0;  {This generates wrong code}
  AUF_INDEX[1].DATA_NO :=0;  {LD HL,01000H missing}
END
.
```

Temporary solution:
There is no know work around at this time.

KPR #: D200069906 Product: Z80/NSC800 C 64824 01.03

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200081554 Product: Z80/NSC800 C 64824 01.04

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
        ;
}
```

KPR #: D200081554 **CONTINUED**

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

Use if(x != 0.0) ;

OR

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200090175 Product: Z80/NSC800 C 64824 02.10

One-line description:
Indirect comparison of parameter bytes may fail

Problem:
Comparisons of indirectly accessed byte parameters may fail.
The following code repeatedly compares *right to *right, which is always true, causing the for loop to cycle forever.

```
"C"
"Z80"

func(left,right)
char *left, *right;
{
   for (;*left == *right; left++, right++)
   {
   }
}
```

Temporary solution:
Use $SHORT_ARITH ON$ or $AMNESIA ON$ compiler directives.

KPR #: D200090217 Product: Z80/NSC800 C 64824 02.10

One-line description:
Certain set operations with explicit type changes may fail.

Problem:
The following code performs arithmetic multiplication and addition, rather than set intersection (AND) and inclusion (OR) operations in the assignments to Byte1.

```
"BZ80"
$EXTENSIONS$
PROGRAM TEST;

TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;

VAR
    Byte1 : BYTE;
    I : SIGNED_16;

PROCEDURE ERR_PROC;
BEGIN
   Byte1 := BYTE(SET_OF_BITS(ADDR(I)) *
                 SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);

   Byte1 := BYTE(SET_OF_BITS(ADDR(I)) +
                 SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);
END;
```

Temporary solution:
Reverse the order of the operands, as in:

```
   Byte1 := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] *
                 SET_OF_BITS(ADDR(I)));

   Byte1 := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] +
                 SET_OF_BITS(ADDR(I)));
```

KPR #: 5000170191 Product: Z80/NSC800 P M 64823-90901 01.00

One-line description:
Documentation and examples for Z80 I/O port

Problem:
Customer would manual to provide documentation that would make writing assembly language routines that interface with compiler routines easier. A very common example is the documentration that would be needed for the customer to write his/her own routine that would use the IO port of the Z80. Our compilers do not supply code generation or library routines for that purpose.

Temporary solution:
No temporary solution.

KPR #: D200059600 Product: Z80/NSC800PASCAL 64823 01.02

Keywords: PASS 1

One-line description:
$Range ON$ causes incorrect code to be generated for a test operation.

Problem:
The following program when compiled with the $RANGE ON$ option wil cause incorrect code to be generated.

```
"B8085" | "BZ80"
$EXTENSIONS$
$RANGE ON$

PROGRAM  BOOLREAL;

VAR  A,B,C    :   REAL;
     L        :   BOOLEAN;

BEGIN
     A := 10.0;
     B := 15.0;
     C := 12.0;


     L := (C < (B+.5)) AND ((C + .5) > A);
END.
```

The two intermediate results "(C < (B +.5))" and "((C+.5) >A)" are anded together and this result is compared with the value two. Thus the case is never true. With RANGE OFF correct code is generated.

Temporary solution:
It is necessary to turn $RANGE OFF$ to obtain correct code. Simply breaking up the expression will not work.

KPR #: D200060186 Product: Z80/NSC800PASCAL 64823 01.02

Keywords: PASS 3

One-line description:
Incorrect data offsets in listing file.

Problem:
I am expanding this to all pascal compilers. The C compilers list the correct offset. $FAR ON$ only applies to the 68000 cross compiler. The other compilers exhibit the defect w/o any options on.

```
"processor name"
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
```

KPR #: D200060186 **CONTINUED**

```
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200090209 Product: Z80/NSC800PASCAL 64823 01.90

One-line description:
Certain set operations with explicit type changes may fail.

Problem:
The following code performs arithmetic multiplication and addition, rather than set intersection (AND) and inclusion (OR) operations in the assignments to Byte1.

```
"BZ80"
$EXTENSIONS$
PROGRAM TEST;

TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;

VAR
    Byte1 : BYTE;
    I : SIGNED_16;

PROCEDURE ERR_PROC;
BEGIN
   Byte1 := BYTE(SET_OF_BITS(ADDR(I)) *
                 SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);

   Byte1 := BYTE(SET_OF_BITS(ADDR(I)) +
                 SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);
END;
```


KPR #: D200090209 **CONTINUED**

Temporary solution:
Reverse the order of the operands, as in:

```
   Byte1 := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] *
                 SET_OF_BITS(ADDR(I)));

   Byte1 := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] +
                 SET_OF_BITS(ADDR(I)));
```

KPR #: D200092593 Product: Z80/NSC800PASCAL 64823 01.90

One-line description:
Incorrect code in complex parameter assignments

Problem:
Detailed Listing for Defect Number LSDqf04287
Text:
  Incorrect code in complex parameter assignments

Assignment of values, accessed in complex ways, to complex parameters may fail. The following code illustrates:

```
TYPE
ADDRESS=RECORD
   CASE SIGNED_16 OF
      1 : (i          : SIGNED_32);
      2 : (lo,hi      : UNSIGNED_16):
      3 : (b0,b1,b2,b3 : BYTE);
      4 : (P          : ^ADDRESS);
   END;
MAIL = ADDRESS;
MAILBOX_QUEUE=RECORD
   mail : ARRAY[0..32] OF ADDRESS;
   outp : SIGNED_16;
END;

VAR
   MBQ : ARRAY[1..32] OF MAILBOX_QUEUE;

PROCEDURE ERR_PROC ( Queue   : SIGNED_16;
                     Message : MAIL      );
BEGIN
   WITH MBQ[Queue] DO
   BEGIN
      Message.lo := mail[outp].lo;
      Message.hi := mail[outp].hi; (* This assignment fails *)
   END;
END;
```

Temporary solution:
Turn on $AMNESIA$ aroung the offending statement.

KPR #: D200093823 Product: Z80/NSC800PASCAL 64823 02.00

One-line description:
Type casting the ADDR function to SET causes 1006 error

Problem:
One-liner:
Type casting the ADDR function to SET for masking may cause error.

Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

generated 1006 error on VAX for Z80 and 8085

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

HERE is an example:

```
"PASCAL"
"Z80"
PROGRAM Error;
$EXTENSIONS$
TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;
VAR
    S : SET_OF_BITS;
    Byte1,Byte2: BYTE;
    I : SIGNED_16;
PROCEDURE BadADDRsetMASK;
 BEGIN
  Byte1:=BYTE(
  (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
  END;
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
TempADDR := ADDR(variable);
Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```


KPR #: 1650006544 Product: Z8000 C 64820 00.01

One-line description:
Code generated for unsigned multiply is the same as for signed multiply.

Problem:
Code generated for an unsigned multiplication is the same as for signed multiplication. The following code is an example:

```
"C"
"processor name"
unsigned u1,u2,u3;
int i1,i2,i3;
main()
{
  u3 = u1*u2;          (*LD     R13,Dstatic        *)
                       (*MULT   RR12,Dstatic+00002H*)
                       (*LD     Dstatic+00004H,R13 *)

  i3 = i1*i2;          (*LD     R11,Dstatic+00006H *)
                       (*MULT   RR10,Dstatic+00008H*)
                       (*LD     Dstatic+0000AH,R11 *)
}
```

Temporary solution:
No known temporary solution.

KPR #: 5000160671 Product: Z8000 C 64820 01.06

One-line description:
Logical AND produces a multiply operation.

Problem:

```
"C"
"Z8001"
$SEPARATE ON$
test()
{
            SUB   R15,#00010H
unsigned short *px, *py, *(*psub)();
unsigned long x;
   px = (unsigned long) py & 0x7F00FFFFL;
            LDL   RR12,RR14[#00004H]
            LDL   RR10,#07F00FFFFH
            AND   R12,R10
            AND   R13,R11
            LDL   RR14[#00000H],RR12
/* Multiplication when (and) is called for */
   px = (unsigned long) &x & 0x7F00FFFFL;
            LDA   RR6,RR14[#0000CH]
            MULTL RQ4,#07F00FFFFH
            LDL   RR14[#00000H],RR6
/* Indirect subroutine call */
   px = (*psub)(py,0);
            PUSH  @RR14,#00000H
            LDL   RR12,RR14[#00006H]
            PUSHL @RR14,RR12
```

KPR #: 5000160671 **CONTINUED**

```
            LDL   RR12,RR14[#0000EH]
            CALL  @RR12
            INC   R15,#6
            LDL   RR10,RR2
/* Next line appears to have wrong offset for RR14 */
            LDL   RR14[#0000CH],RR10
            INC   R15,#16
   Rtest
```

Two code generations problems:
1) Logical "and" operation produces a multiply
2) After calling a function indirectly that returns a pointer
   the register offset is incorrect for referencing variables

Temporary solution:
No temporary solution at this time.

KPR #: 5000181545 Product: Z8000 C 64820 01.04

One-line description:
$OPTIMIZE$ compiler directive works differently for signed and unsigned.

Problem:
The $OPTIMIZE$ compiler directive optimizes operations on unsigned types like char, but doesn't optimize operations on signed variables like ints. When this directive is on, the compiler should not reload a register if it has already been loaded with a previous value. The compiler should assume that the value is correct. When this directive is off, then register contents should be forgotten, and reloaded. In the following code example, with $OPTIMIZE ON$, the compiler optimizes the code if the variables are unsigned, but doesn't if they are signed:

```
"C"
"processor name"
$OPTIMIZE ON$
main()
{
  char  *a,*b;     (*if these are declared as int, then no optimization
                        is done*)
  while ((*a=*b) != '\0')
   {
    a++;
    b++;
   }
}
```

Temporary solution:
No known temporary solution at this time.

KPR #: 5000246983 Product: Z8000 C 64820 01.06

One-line description:
Local parms not accessed properly when func called via pointer.

Problem:
In the following program the code generated for the access of func2's local variables is incorrect. This happens only if func2 is called via a pointer and is defined after main.

```
"C"
"Z8001"


int *func2();

main()
{

int i;
int (*func)();


func = func2;
i = (*func)(i);

}

int *func2(p1)
int p1;

{
int j,k,l,m;

      j =1;
      k = 1;
      l = 1;
      m = 1;

/* All of the above references to the local variables will cause
   bad code to be generated.  For example, the reference to j is
   RR14[0004] rather than RR14[0000].                  */


Temporary solution:
Define the function and its body prior to making calls to it
via a pointer.

func2()
{

.
.
.
}

main()
```

KPR #: 5000246983 **CONTINUED**

```
{

int (*func)();
int i;

   func=func2;

  i = (*func)(i);

}
```

KPR #: 5000280958 Product: Z8000 C 64820 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
Compiler does not create an 'array too large' error when size > 32k.

Problem:
Z8001 C compiler generates 1113 error. When the array area is more than 32k bytes, compiler generates 1113 error. The C compiler on 64100 does not generate that error.

EXAMPLE:

```
"C"
"Z8001"
  char   array[0x8000];
  main()
  {
  }
```

NOTE: according to SR#D200078873 the actual error may exist in the 64100 compiler. The Z8001 catches the 'array too large' error - the 64100 compiler allows this error to pass unnoticed.

Temporary solution:
There is no workaround available.

KPR #: D200061762 Product: Z8000 C 64820 01.04

One-line description:
Inconsistient error message when linking ASM.R files versus COMP.R files

Problem:
When linking files with DATA addresses defined by more than 16 bits the assembler output causes link errors. If the program below is assembled the linker will flag a legal range error, but, will still generate an absolute file. Flagging the legal range error seems reasonable as the immediate mode of addressing only allows 16 bits for its value (remember the data are must be put in at 10000H or greater for this error to occur).

```
"Z8002"
        DATA
LABEL   RMB         1
```


KPR #: D200061762 **CONTINUED**

```
        PROG
START   PUSH        @R15,#LABEL     ;ONLY 16 BITS ARE ALLOCATED/ALLOWED
                                    ;FOR THE IMMEDIATE VALUE.
```

Now, if the below file is compiled it will generate a similar PUSH instruction is generated. If this file is linked with the data area at 10000H or greater NO error is given. Furthermore, if you turn ASM_FILE on and assemble and link the ASMZ8002 file the legal range error is flagged, but, identical absolutes are generated. In summary, it seems the LR error is appropiate because the immediate mode of addressing only allows 16 bits for its value. If for some reason the LR is inappropriate then the linker should be changed so it is consistient.

```
"C"
"Z8002"

$ASM_FILE ON$
$SEPARATE ON$

float  table[10];

main() {

float  *i;

*i = table[1] + table[2];

}
```

Temporary solution:
No temporary solution at this time.

KPR #: D200064808 Product: Z8000 C 64820 01.05

One-line description:
Superfluous register load in switch statement on the 64000

Problem:
In certain situations, the Z8000 C compiler generates different code on the 64000 than it does on the hosts. The C constructs that cause this to occur are switch statements where the switch value is a four byte quantity accessed via the structure pointer operator. This code difference manifests itself as a superfluous register load that appears to have no adverse effects on the execution of the program. The extra register load occurs only on code compiled on the 64000, not on the hosts.

The following code demonstrates this problem:

```
"C"
"Z8001"
```

KPR #: D200064808 **CONTINUED**

```
struct GROUP {
    char data1;
    long data2;
    char data3;
    int  data4;
    long data5;
    } group_struct = {'A',0x42L,'C',0x44,0x45L};

struct GROUP *grp_ptr = {&group_struct};

main()
{
  switch(grp_ptr->data2)    (*extra code is LDL RR12,RR0 instruction*)
   {
     case 0x42 : break;
     default : break;
    }
}
```

Temporary solution:
No known temporary solution at this time.

KPR #: D200068155 Product: Z8000 C 64820 01.05

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"

struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200069781 Product: Z8000 C 64820 01.05

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200078873 Product: Z8000 C 64820 01.06

One-line description:
Oversized data segment not being flagged as an error.

Problem:
Using the z8001 C compiler the compiler does not always flag too large of data area. See the example code below.

```
"C"
"Z8001"

char   array[60000];  /* should cause pass II error "Data
                         segment too large. */
```

KPR #: D200078873 **CONTINUED**

```
main()
{}
```

If you have two arrays of whose sum total size is greater
than 32K then the appropriate error message is generated.
On the 9000 series of hosts the above program causes
a pass III error (program counters disagree.) On the
64100 the file incorrectly reports no errors.

Temporary solution:
No temporary solution.

KPR #: D200079616 Product: Z8000 C 64820 01.06

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This
instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
```


KPR #: D200079616 **CONTINUED**

```
      }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function
memory test. This will cause slightly more code to
be generated.

KPR #: D200081521 Product: Z8000 C 64820 01.06

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself
as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
        ;
}
```

Customer feels that this variable should be evaluated to see if it
is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;
```

OR

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200085381 Product: Z8000 C 64820 01.06

One-line description:
Function calls via pointers with parameters mess up subsequent calls.

Problem:
Calls to functions made via a dereferenced pointer where parameters are passed will cause problems in accessing objects on the stack in subsequent functions. The following code illustrates.

```
"C"
"Z8002"

func1()
{
   int (*TEST)();

   (TEST)(11);
}

int func2(parm)
int parm;
{
   return(parm);    /* parm not correctly accessed */
}
```

Temporary solution:
There is no clean solution. You can avoid indirect calls when parameters are involved. Or make sure that the indirect call is the last thing in the source file.

KPR #: 5000123497 Product: Z8000 PASCAL 64816 01.09

One-line description:
Jump table may generate code which accesses wrong data space.

Problem:
The following program will generate a jump table which uses a LD R3,<src> instruction to execute. This instruction causes the status lines to indicate a RAM access when in fact the jump table is in the PROG (ROM) area.

```
"Z8002"
$EXTENSIONS ON$
$SEPARATE ON$

PROGRAM   TEST;

VAR   V    :    INTEGER;

BEGIN
   V  := 10;

   CASE   V   OF

          1:     V:=2;
          2:     V:=3;
          3:     V:=4;
          4:     V:=5;
          5:     V:=6;
          7:     V:=8;
   END;

END.
```

Temporary solution:
No known temporary solution at this time.

KPR #: 5000134916 Product: Z8000 PASCAL 64816 01.10

Keywords: PASS 3

One-line description:
Calling func. twice in statement causes return value to be overwritten

Problem:
In the following program the user defined function "SQR" is called twice in one statement. The result of the first call is overwritten by the results of the second call because RH3 is not saved.

```
"Z8001"
$EXTENSIONS ON$
PROGRAM FUNCTION_CALL;

VAR
```

KPR #: 5000134916 **CONTINUED**

```
$EXTVAR ON$
   RESULT          :    REAL;
   NUMBER1,NUMBER2:     REAL;

$EXTVAR OFF$

FUNCTION SQR( NUM : REAL) : REAL;  EXTERNAL;

PROCEDURE  TESTFUNC;
BEGIN
   RESULT := SQR (NUMBER1) + SQR (NUMBER2);
END.
```

Temporary solution:
Break up the statement into two separate statements with the first call storing the result of "SQR" in a temporary variable.

```
"Z8001"
$EXTENSIONS ON$

PROGRAM FUNCTION_CALL;

VAR
$EXTVAR ON$
RESULT            :REAL;
NUMBER1,NUMBER2   :REAL;
TEMP              :REAL;
$EXTVAR OFF$

FUNCTION  SQR( NUM: REAL ) : REAL; EXTERNAL;

PROCEDURE TESTFUNC;
BEGIN
  TEMP := SQR(NUMBER1);
  RESULT := TEMP + SQR(NUMBER2);
END.
```

KPR #: 5000150151 Product: Z8000 PASCAL 64816 01.04

One-line description:
"Downto" used in a for statement generates incorrect code.

Problem:
Using "downto" in a for statement generates incorrect code. The loop will only be executed once, because the jump condition at the end of the loop jumps on no carry instead of on carry.
The following is an example:

```
"processor name"
PROGRAM TEST;
TYPE
    DAYS = (SUN,MON,TUES,WED,THURS,FRI,SAT);
VAR
    DAY_COUNT:ARRAY[DAYS] of SIGNED_16;
    DAY:DAYS;
```


KPR #: 5000150151 **CONTINUED**

```
BEGIN
    FOR DAY:= SAT DOWNTO SUN DO
        DAY_COUNT[DAY] := 17;
```

```
Code generated is:
LDRB    RL7,DTEST+0000EH
SUBB    RH7,RH7
SLL     R7,#1
LD      DTEST[R7],#00011H
LDRB    RL7,DTEST+0000EH
ADDB    RL7,#0FFH
LDRB    DTEST+0000EH,RL7
JR      UGE,TEST00_0
```

```
END.
```

The JR UGE,TEST00_0 instruction jumps on no carry instead of on carry, and the loop only executes once.

Temporary solution:
No known temporary solution at this time.

KPR #: D200060145 Product: Z8000 PASCAL 64816 01.10

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:

KPR #: D200060145 **CONTINUED**

If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200061721 Product: Z8000 PASCAL 64816 01.10

One-line description:
Inconsistient error message when linking ASM.R files versus COMP.R files

Problem:
When linking files with DATA addresses defined by more than 16 bits the assembler output causes link errors. If the program below is assembled the linker will flag a legal range error, but, will still generate an absolute file. Flagging the legal range error seems reasonable as the immediate mode of addressing only allows 16 bits for its value (remember the data are must be put in at 10000H or greater for this error to occur).

```
"Z8002"
        DATA
LABEL   RMB         1
        PROG
START   PUSH        @R15,#LABEL     ;ONLY 16 BITS ARE ALLOCATED/ALLOWED
                                    ;FOR THE IMMEDIATE VALUE.
```

Now, if the below file is compiled it will generate a similar PUSH instruction is generated. If this file is linked with the data area at 10000H or greater NO error is given. Furthermore, if you turn ASM_FILE on and assemble and link the ASMZ8002 file the legal range error is flagged, but, identical absolutes are generated. In summary, it seems the LR error is appropiate because the immediate mode of addressing only allows 16 bits for its value. If for some reason the LR is inappropriate then the linker should be changed so it is consistient.

```
"C"
"Z8002"

$ASM_FILE ON$
$SEPARATE ON$

float  table[10];

main() {

float  *i;

*i = table[1] + table[2];

}
```

Temporary solution:
No temporary solution at this time.

KPR #: D200073015 Product: Z8000 PASCAL 64816 01.11

One-line description:
BA address mode may attempt to use RRO Illegally as source

Temporary solution:
None.

KPR #: D200085282 Product: Z8000 PASCAL 64816 01.12

One-line description:
$RANGE$ & type conversion of UNSIGNED_32 var may cause error 1006.

Problem:
A pass 2 error #1006 on the HP-UX and VMS compilers, or compilation hanging in pass 2 on the HP 64000, may occur under the following conditions:

1) The $RANGE ON$ compiler directive is in effect.
2) An arithmetic operation between two UNSIGNED_32 vars is type converted to SIGNED_32 and used as an array index.
3) The $RANGE OFF$ is then put into effect.
4) There is an array reference inside a conditional.

The following code illustrates:

```
"Z8002"
$EXTENSIONS ON$
PROGRAM ERROR;

PROCEDURE TEST;
VAR
   P,R : UNSIGNED_32;
   ARR : ARRAY [0..5] OF BYTE;
BEGIN
$RANGE ON$
   ARR[SIGNED_32(P+R)] := 95;
$RANGE OFF$
   IF ARR[5] <> 95 THEN ;       (* error 1006 or pass 2 hangs *)
END;
```

KPR #: D200093658 Product: Z8000 PASCAL 64816 01.90

One-line description:
Large sets may produce invalid results for elements outside set range

Problem:
Detailed Listing for Defect Number LSDqf04513
Text:
  Test for set inclusion checks outside set boundaries.
.....
One-liner:
 Large sets may produce invalid results for elements outside set range

The set inclusion operation may test undefined bit when the element

KPR #: D200093658 **CONTINUED**

being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,        }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the the full 256 bit set implementation.

Instead of defining the large set as:

```
DIG = SET OF '0'..'9';
```

It could be rewritten:

```
digch = SET OF CHAR;
```

With the sets now using a full 256 bits, all bits will be set and tested properly.

```
TYPE
       {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,        }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```


KPR #: D200093666 Product: Z8000 PASCAL 64816 01.90

One-line description:
Type casting the ADDR function for masking may cause error #1006

Problem:
One-liner:
Type casting the ADDR function to SET for masking may cause error.

Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
    Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

will geneate incorrect code.

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

HERE is an example:

```
"PASCAL"
"Z8001"
PROGRAM Error;
$EXTENSIONS$
TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;
VAR
    S : SET_OF_BITS;
    Byte1,Byte2: BYTE;
    I : SIGNED_16;
PROCEDURE BadADDRsetMASK;
 BEGIN
  Byte1:=BYTE(
  (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
  END;
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
    Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
    TempADDR := ADDR(variable);
    Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: D200093674 Product: Z8000 PASCAL 64816 01.90

One-line description:
Boolean Index into array generates bad code

Problem:
Text:
Boolean Index into array generates bad code

```
$EXTENSIONS$

VAR
       bool   :  BOOLEAN;
       int    :  INTEGER;
       arr3 :  ARRAY[BOOLEAN,1..4,BOOLEAN] OF BYTE;

PROCEDURE LSD1a01247;

      {Initialize array arr3}
   FOR int:= 1 TO 4 DO
    FOR bool:= TRUE DOWNTO FALSE DO
     BEGIN
      arr3[bool,int,FALSE]:= 2;
      arr3[bool,int,TRUE]:= -2;
     END;
   bool := FALSE;
                { variable::constant::variable }
  IF  arr3[bool,2,bool] = 2  { arr3[FALSE,2,FALSE] = 2}
    THEN
       {OK}
    ELSE
       {  "** ERROR LSD1a01274 #16 in file TEST110:Plus_P **."  };

                { constant::constant::variable }
  bool := TRUE;
  IF  arr3[FALSE,4,bool] = -2
    THEN
       {OK}
    ELSE
       {  "** ERROR LSD1a01274 #19 in file TEST110:Plus_P **."  };
END;
```

Temporary solution:
Don't use Boolean array index.

KPR #: 5000131573 Product: Z8001/2 EMUL M 64980-90923 01.00

One-line description:
Need more info on sharing user system calls & monitor interaction.

Problem:
Customer feels that manual should include more information on the sharing of user system calls with the emulation monitor's system calls. Include a short section which explains how this can be done, including an example of user code which would handle all system calls and fall through to the monitor if appropiate. The customer was able to do this himself but felt it should be explained in the manual.

Also, the chapter "EMULATION CONFIGURATION" subsection "Monitor Interaction" talks of the transparency of the Break system call instruction, including the jamming of the PSA information. Customer felt that this info should be more detailed.

Temporary solution:
See problem text.

KPR #: 5000258616 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
CANNOT ACCESS COMPILER GENERATED SYMBOLS IN HP64000-UX EMUL ENVIRONMENT

Problem:
Compiler generated symbols are specific to the HP64000 software products. While in the HP64000-UX emulation environment, users can access all symbol information in the user program, but can not access these compiler generated symbols for displaying, tracing and modifying purposes. An enhancement investigation may be possible if needs are warranted.

KPR #: D200095729 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Emulation core dumps when run in a small window.

Problem:
When emulation is run in a small window, it end releases before the status line comes up and generates a core file. It should leave the emulator locked and display a message, "Display size is too small".

KPR #: D200096016 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
"Copy to Read-Only files", fails to deliver an error message to screen

Problem:
If an attempt is made to use the copy command to write to read only file the command fails silently. The error message "permission denied" never shows up.

Part Number: 5958-6018
PRINTED IN U.S.A. FEBRUARY 1989
E0289

HEWLETT PACKARD