

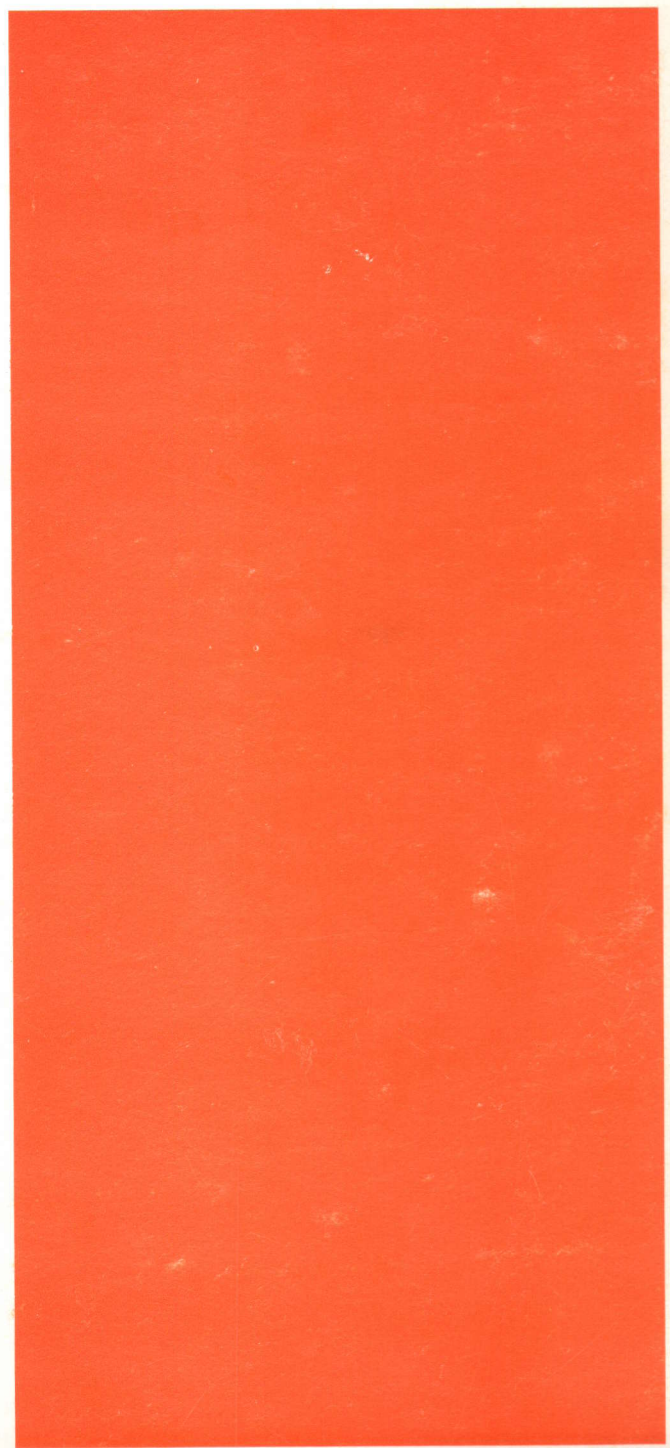
Honeywell

TIME-SHARING SYSTEM
TERMINAL/BATCH
INTERFACE FACILITY

SERIES 600/6000

GCOS

SOFTWARE



Honeywell

TIME-SHARING SYSTEM TERMINAL/BATCH INTERFACE FACILITY ADDENDUM B

SERIES 600/6000

GCOS

SOFTWARE

SUBJECT:

Additions and Changes to Series 600/6000 Time-Sharing System
Terminal/Batch Interface Facility.

SPECIAL INSTRUCTIONS:

This is the second addendum to BR99, Revision 1, dated January 1972.
The previous Addendum A still applies.

Insert the attached pages into the manual according to the collating
instructions on the back of this cover. Change bars in the margins
indicate technical additions and changes; asterisks indicate deletions.
These changes will be incorporated into the next revision of the
manual.

NOTE: Insert this cover after the manual cover to indicate the
updating of the document with Addendum B.

SOFTWARE SUPPORTED:

Series 600 Software Release 7.0
Series 6000 Software Release E

DATE:

January 1973

ORDER NUMBER:

BR99B, Rev. 1

8875
7.5C1173
Printed in U.S.A.

COLLATING INSTRUCTIONS

To update this manual, remove old pages and insert new pages as follows:

<u>Remove</u>	<u>Insert</u>
2-3 through 2-6	2-3, 2-4 2-5, blank 2-5.1, 2-6
2-9, 2-10 2-10.1, blank 2-13, 2-14 4-1 through 4-4	2-9, 2-10 2-10.1, blank 2-13, 2-14 4-1 through 4-4

GCOS is a coded system designed to extend the power of Series 600/6000 in the areas of program preparation and maintenance, data control, operations control, and utility functions. It is supported by comprehensive documentation and training; periodic program maintenance and, where feasible, improvements are furnished for the current version of the system, provided it is not modified by the user.

Honeywell

TIME-SHARING SYSTEM TERMINAL/BATCH INTERFACE FACILITY

SERIES 600/6000

GCOS

SUBJECT:

Description and Use of Time-Sharing Subsystems and Batch-Type Programs.

SPECIAL INSTRUCTIONS:

This manual, Order Number BR99, Rev. 1, supersedes BR99, Rev. 0 (formerly CPB-1642A), dated March 1970; Technical Information Bulletin 1642A-1, dated May 1970; Technical Information Bulletin 1642A-2, dated September 1970; and Addendum No. 3, dated January 1971. The new order number is assigned to be consistent with the overall Honeywell publications numbering system. Software Releases were formerly called Systems Development Letters or SDL's. Technical additions and changes from the previous edition are indicated by change bars in the margins; deletions are indicated by asterisks.

SOFTWARE SUPPORTED:

Series 600 Software Release 6.0
Series 6000 Software Release D

INCLUDES UPDATE PAGES ISSUED AS ADDENDUM A, JUNE 1972.

DATE:

January 1972

ORDER NUMBER:

BR99, Rev. 1 (Formerly CPB-1642)

PREFACE

This manual describes a group of special purpose subsystems and batch-type programs that are available to the users of a Honeywell Series 6000 Time-Sharing System. Included in this manual are subsystem usage procedures, terminal operating procedures, and definitions of Time-Sharing System error messages.

The time-sharing functions described in this manual are also available on the Honeywell Series 600 Information Processing System.

Series 600/6000 GCOS is a coded system designed to extend the power of Series 600/6000 in the areas of program preparation and maintenance, data control, operations control, and utility functions. It is supported by comprehensive documentation and training; periodic program maintenance and, where feasible, improvements are furnished for the current version of the system, provided it is not modified by the user.

FUNCTIONAL LISTING OF PUBLICATIONS
for
SERIES 600 SYSTEM

FUNCTION	APPLICABLE REFERENCE MANUAL		ORDER NO.
	TITLE	FORMER PUB. NO.	
	Series 600:		
Hardware reference:			
Series 600	System Manual	371	BM78
DATANET 355	DATANET 355 Systems Manual	1645	BS03
Operating system:			
Basic Operating System	Comprehensive Operating Supervisor (GCOS)	1518	BR43
Control Card Formats	Control Cards	1688	BS19
System initialization:			
GCOS Startup	System Operating Techniques	DA10	DA10
Communications System	GRTS/355 Startup Procedures	1715	BJ70
Storage Subsystem Startup	DSS180 Disk Storage Subsystem Startup Procedures	DA11	DA11
Data management:			
File system	GCOS File System	1513	BR38
Integrated Data Store (I-D-S)	Integrated Data Store	1565	BR69
File Processing	Indexed Sequential Processing	DA37	DA37
Program maintenance:			
Object Program	Source Object Editor	1723	BJ71
System Editing	System Library Editor	1687	BS18
Test system:			
Peripheral on-line testing	GCOS On-Line Peripheral Test System (OPTS-600)	1573	BR76
Language processors:			
Macro Assembly Language	Programming Reference Manual	1004	BN86
COBOL Language	COBOL Reference Manual	1652	BS08
COBOL Usage	COBOL User's Guide	1653	BS09
ALGOL Language	ALGOL	1657	BS11
JOVIAL Language	JOVIAL	1650	BS06
FORTRAN Language	FORTRAN	1686	BJ67
FORTRAN IV Language	FORTRAN IV	1006	BN88
DATANET 355	DATANET 355 Macro-Assembly Program	1660	BB98
Generators:			
Sorting	Sort/Merge	1005	BN87
Merging	Sort/Merge	1005	BN87
Simulators:			
DATANET 355 Simulation	DATANET 355 Simulator	1663	BW23
Remote terminal system:			
DATANET 355	GRTS/355 Programming Reference	1664	BJ66
DATANET 30	GRTS/30 Programming Reference	1558	BR68

FUNCTION	APPLICABLE REFERENCE MANUAL		ORDER NO.
	TITLE	FORMER PUB. NO.	
	Series 600:		
Service and utility routines:			
File I/O	File and Record Control	1003	BN85
Loader	General Loader	1008	BN90
Utility programs	Utility	1422	BQ66
Conversion	Bulk Media Conversion	1096	BP30
System Accounting	GCOS Accounting Summary		
	Edit Program	1651	BS07
FORTTRAN	FORTTRAN IV Subroutine		
	Libraries	1620	BR95
Controller loader	Relocatable Loader	DA12	DA12
Time-sharing systems:			
Operating System	GCOS Time-Sharing System		
	General Information	1643	BS01
System Programming	GCOS Time-Sharing		
	Terminal/Batch	1642	BR99
System Programming	GCOS Time-Sharing System -		
	System Programmer's		
	Reference	1514	BR39
BASIC Language	Time-Sharing BASIC	1510	BR36
FORTTRAN Language	Time-Sharing FORTRAN	1566	BR70
Text Editing	Time-Sharing Text		
	Editor	1515	BR40
Handbooks:			
Console Messages	GCOS Typewriter Messages	1477	BR09
Index	Comprehensive Index	1499	BR28
Pocket guides:			
Time-Sharing Programming	GCOS Time-Sharing System	1661	BS12
Macro Assembly Language	GCOS GMAP	1673	BS16
COBOL Language	COBOL	1689	BJ68
Control Card Formats	GCOS Control Cards and Abort		
	Codes	1691	BJ69
Software maintenance (SMD):			
Table definitions	Introduction and System		
	Tables	1488	BR17
Startup program	Startup (INIT)	1489	BR18
Input system	Input System	1490	BR19
Peripheral allocation	Dispatcher/Peripheral		
	Allocation	1491	BR20
Core allocation/rollcall	Rollcall, Core Allocation and		
	Operator Interface	1492	BR21
Fault processing	Fault Processing	1493	BR22
Channel modules	I/O Supervision (IOS)	1494	BR23
Error processing	Exception Processing	1495	BR24
Output system	Termination and SYSOUT	1496	BR25
File system modules	File System	1497	BR26
Utility programs	GCOS Utility Routines	1498	BR27
Time-sharing system	Time-Sharing Executive	1501	BR29

FUNCTIONAL LISTING OF PUBLICATIONS
for
SERIES 6000 SYSTEM

FUNCTION	APPLICABLE REFERENCE MANUAL		ORDER NO.
	TITLE	FORMER PUB. NO.	
	Series 6000:		
Hardware reference:			
Series 6000	Summary Description	DA48	DA48
DATANET 355	DATANET 355 Systems Manual	1645	BS03
Operating system:			
Basic Operating System	Comprehensive Operating Supervisor (GCOS)	1518	BR43
Control Card Formats	Control Cards	1688	BS19
System initialization:			
GCOS Startup	System Startup and Operation	DA06	DA06
Communications System	GRTS/355 Startup Procedures	1715	BJ70
Storage Subsystem Startup	DSS180 Disk Storage Subsystem Startup Procedures	DA11	DA11
Data management:			
File System	GCOS File System	1513	BR38
Integrated Data Store (I-D-S)	Integrated Data Store	1565	BR69
File Processing	Indexed Sequential Processor	DA37	DA37
Program maintenance:			
Object Program	Source Object Editor	1723	BJ71
System Editing	System Library Editor	1687	BS18
Test system:			
On-Line Peripheral Testing	GCOS On-Line Peripheral Test System (OPTS-600)	1573	BR76
Language processors:			
Macro Assembly Language	Programming Reference Manual	1004	BN86
COBOL Language	COBOL Reference Manual	1652	BS08
COBOL Usage	COBOL User's Guide	1653	BS09
ALGOL Language	ALGOL	1657	BS11
JOVIAL Language	JOVIAL	1650	BS06
FORTRAN Language	FORTRAN	1686	BJ67
DATANET 355	DATANET 355 Macro-Assembly Program	1660	BB98
Generators:			
Sorting	Sort/Merge	1005	BN87
Merging	Sort/Merge	1005	BN87
Simulators:			
DATANET 355 Simulation	DATANET 355 Simulator	1663	BW23

FUNCTION	APPLICABLE REFERENCE MANUAL		
	TITLE	FORMER PUB. NO.	ORDER NO.
	Series 6000:		
Service and utility routines:			
File I/O	File and Record Control	1003	BN85
Loader	General Loader	1008	BN90
Utility Programs	Utility	1422	BQ66
Conversion	Bulk Media Conversion	1096	BP30
System Accounting	GCOS Accounting Summary		
	Edit Program	1651	BS07
FORTRAN	FORTRAN IV Subroutine		
	Libraries	1620	BR95
Controller Loader	Relocatable Loader	DA12	DA12
Time-sharing systems:			
Operating System	GCOS Time-Sharing System		
	General Information	1643	BS01
System Programming	GCOS Time-Sharing		
	Terminal/Batch	1642	BR99 ✓
System Programming	GCOS Time-Sharing System -		
	System Programmer's		
	Reference	1514	BR39
BASIC Language	Time-Sharing BASIC	1510	BR36
FORTRAN Language	FORTRAN	1686	BJ67
Text Editing	Time-Sharing Text Editor	1515	BR40
Remote terminal system:			
DATANET 355	GRTS/355 Programming		
	Reference	1664	BJ66
DATANET 30	GRTS/30 Programming		
	Reference	1558	BR68
Handbooks:			
Console Messages	GCOS Typewriter Messages	1477	BR09
Index	Comprehensive Index	1499	BR28
Pocket guides:			
Time-Sharing Programming	GCOS Time-Sharing System	1661	BS12
Macro Assembly Language	GCOS GMAP	1673	BS16
COBOL Language	COBOL	1689	BJ68
Control Card Formats	GCOS Control Cards and Abort		
	Codes	1691	BJ69

TABLE OF CONTENTS

		Page
Section I	Terminal/Batch Interface Description.	1-1
Section II	CARDIN Subsystem.	2-1
	CARDIN Function.	2-1
	CARDIN Operation	2-1
	CARDIN Detailed Operation.	2-3
	CARDIN Command Language	2-3
	RUN or PRINT Question/Answer Sequence	2-7
	ASCBCD Question/Answer Sequence	2-12
	BCDASC Question/Answer Sequence	2-14
	BPUNCH or BPRINT Question/Answer Sequence	2-15
	First Line Reformatting Information in the File	2-15
	\$ SELECTA Card Option	2-16
Section III	SCAN Subsystem.	3-1
	SCAN Function.	3-1
	SCAN Operation Description	3-2
	SCAN Verbs	3-3
Section IV	FDUMP Subsystem	4-1
	FDUMP Function	4-1
	FDUMP Operational Description.	4-1
	FDUMP Error Indication and Replies.	4-3
	FDUMP Copy Function Details	4-4
Section V	Conversational Debug.	5-1
	RBUG General Description	5-1
	RBUG Instructions.	5-2
Section VI	Conversational File and Record Input/Output	6-1
	Control Cards for File and Record Control I/O	6-1
	Applicable I/O Calls	6-2
	Special File and Record Control I/O Considerations	6-3
Section VII	JOUT Subsystem.	7-1
	JOUT Function.	7-1
	JOUT Operational Description	7-2
Appendix A	Interface Usage Examples.	A-1
Index.		X-1



SECTION I

TERMINAL/BATCH INTERFACE DESCRIPTION

The Terminal/Batch Interface Facility of the Honeywell Series 6000 Time-Sharing System¹ consists of an interrelated set of time-sharing subsystems and batch programming facilities. These combine to provide the time-sharing terminal user with full batch programming capabilities. This facility allows the user to submit a batch program, debug the program, and inspect the program output; all from a remote terminal. This interface also facilitates direct, interactive communication between the user's batch program and his (or another) terminal.

The elements making up this facility fall into three categories:

1. The CARDIN time-sharing subsystem, which allows the actual submission of batch jobs to the System Input program and integrates the other elements into a functional whole.
2. Supporting subsystems, SCAN, JOUT, and FDUMP, which allow batch output scanning, batch output manipulation, and file dumping and modification (in octal), respectively. (The latter is primarily a debugging tool.)
3. Two batch dimension features, Conversational Debug Routine (RBUG) and Conversational-I/O extensions to the File and Record Control logical record processing routines, are for inclusion in the batch program itself. These allow direct access connection between the program and the remote terminal, where desired.

¹All references to Series 6000 also apply to Series 600 unless otherwise stated.



SECTION II
CARDIN SUBSYSTEM

CARDIN FUNCTION

CARDIN is the component subsystem of the Series 6000 Time-Sharing System, which makes available to the time-sharing user the wider facilities of batch processing. It permits a remote terminal user to perform such functions as developing and checking out a batch program written in macro assembler (GMAP) language. CARDIN also provides a convenient means of maintaining and initiating application programs such as APT, Linear Programming, and data reduction programs. Almost any other situation where a time-sharing terminal user needs access to batch facilities can be accommodated by CARDIN.

Essentially, CARDIN allows the user to create at his terminal a card image input file, submit it for batch processing, and receive notice of the result (successful termination, abort, etc.). The user then may have the job output printed at the central site or a remote batch facility, or inspect the output file conversationally by selective listing at the terminal. (Refer to SCAN Subsystem, Section III.)

The TALK option of CARDIN allows a terminal user to talk to a program running in the central computer system. This option places the user's terminal in direct access, conversational connection with any program that was submitted through CARDIN and that contains remote terminal input/output. The TALK option facilitates the use of the Conversational Debug (RBUG) routine by allowing interactive, execution time debugging from the user's terminal.

Permission from the master user must exist in the Time-Sharing System before a specific user can employ CARDIN. If master user's permission has been granted for use of TALK disposition, use of all CARDIN functions is implied.

CARDIN OPERATION

As in Time-Sharing BASIC, the file-building facility provided by CARDIN requires line numbers. BASIC-like editing of keyboard input and updating of the current file is performed. The line numbers can be either stripped automatically from each line or moved to the card image sequence number field at the time the job input file is passed to the batch world.

In converting the user's time-sharing source ASCII file to a BCD card image job input file, one line of terminal input represents one card. The user must include in his source file all cards that would normally be required in the batch environment except the \$ SNUMB, \$ USERID, and ***EOF cards. A \$ USERID card is generated by CARDIN when required for a permanent file, so that the user need not reveal his log-on password. This card does not show on a LIST request for the file. However, the user may supply his own \$ USERID card if he wishes, and the System Input program uses this rather than the one supplied by CARDIN.

Tab characters may be used when building the file in order to avoid keying in the blanks required by a fixed-field language such as GMAP. The tab character can be assigned a sequence of character-position values--for example, tab,8,16,32,73. Each successive occurrence of the tab character within a line causes the next unused value in the sequence to be assigned to it. More than one tab character may be specified, each with its own sequence of tab settings.

ASCII-to-BCD conversion and reformatting of a file is obtained at the following times:

1. When the user gives a RUN command, causing conversion onto a scratch file and passage of the file to the System Input program for processing.
2. When the user gives an ASCBCD command, causing conversion onto a permanent file only.

In neither case is the original ASCII file itself affected.

The information needed by CARDIN for converting and reformatting a file may be supplied by the user in either of two ways:

- The tab characters, settings, and line number disposition may be supplied in response to questions asked by CARDIN.
- To minimize the user/subsystem interface, the same information may be supplied on the first line of the file itself.

The user may also convert a system format Binary Coded Decimal (BCD) file to a time-sharing ASCII file by means of the BCDASC command.

The general operation of the subsystem is as follows:

1. The terminal user builds his file under CARDIN or calls for an OLD file previously prepared under CARDIN or other time-sharing subsystem. If line numbered, the file can then be corrected, updated, resequenced, etc., using standard time-sharing commands. The user can request that his current file be run, using RUN or RUN filedescr.
2. When the user requests a run of his current file, he is given the SNUMB sequence number of his batch run. He is then asked a series of questions concerning the format of his file and the desired disposition of the job, if this information did not appear on the first line of the file.
3. When the question/answer sequence is completed, the user's time-sharing source file (which is in ASCII) is converted to BCD and put in the desired card format. It is then passed to the System Input program for processing.
4. Following a batch job termination message, the user may inspect his output by giving the command SCAN filename, where filename is the name of a permanent file upon which the batch-job output has been placed. This placement is achieved by appropriate use of the \$ PRMFL control card.

The output scanning facilities of the SCAN subsystem are available through use of the SCAN command.

5. Instead of having the file converted and processed, the user can have it converted and then printed out at the terminal in the requested job input format by means of the PRINT command. This facilitates visual inspection before actually passing the file to the System Input program (with a subsequent RUN command). The PRINT command initiates a question/answer sequence similar to that for the RUN command.

CARDIN DETAILED OPERATION

CARDIN Command Language

The CARDIN commands which follow are not all unique to CARDIN, but some have special meaning.

!

*

*

- RUN

1. RUN

Convert and pass the current file to batch processing, following the question/answer sequence that is initiated by this command if the file does not contain first line reformatting information. The SNUMB of the job is printed.

2. RUN filedescr

Convert and pass the file saved under filedescr to the batch world for processing, following the question/answer sequence that is initiated by this command if the file does not contain first line reformatting information. The current file, if one exists, is not disturbed. Filedescr is simply the file name for a quick access permanent file; for other permanent files it is the file description. The SNUMB of the job is printed.

- PRINT

1. PRINT

Reformat the contents of the current file and print it at the terminal, following the question/answer sequence that is initiated by the command if the file does not contain first line reformatting information.

2. PRINT filedescr 1; filedescr 2;...;filedescr n

Reformat the contents of the file saved under filedescr and print it at the terminal, following the question/answer sequence that is initiated by this command if the file does not contain first line reformatting information. The current file, if one exists, is not disturbed. (See definition of filedescr under RUN command.)

- JDAC

JDAC name

Allows time-sharing terminal user to establish connection with a GRTS (General Remote Terminal Supervisor) Direct Access (DAC) program, where name is the program's remote inquiry name already supplied to GCOS via a MME GEROUT. If name is not supplied with the JDA request, the system returns a NAME? query.

When the direct access program terminates, the system returns to the command entry (build mode) level within CARDIN.

- JSTS

JSTS snumb (Job Status)

Causes the current batch processing status of the job specified by snumb to be printed at the terminal in plain text.

With an OUTPUT WAITING or an OUTPUT COMPLETE message, JSTS will attempt to give termination status. For example:

```
*JSTS 0003T
0003T OUTPUT COMPLETE
IF LAST JOB SUBMITTED, STATUS WAS:
F0 - MEMORY ADDRESS FAULT
```

This termination status information is available only for the most recent job submitted by the user during the current log-on session. If the job referred to by snumb is not the last job submitted, the termination status does not apply to it, but to the last job submitted.

- JABT

JABT snumb (Job Abort)

Causes the batch job specified by snumb to be aborted, with an abort code assigned. Only jobs containing a valid \$ USERID control card for the requesting user will be aborted.

- ASCBCD

ASCBCD ascfil; bcdfil (ASCII-to-BCD)

```
ASCBCD
INPUT FILES ascfil;bcdfil
```

If the response to "INPUT FILES" is null (carriage return), control is returned to the level from which the command was issued. The ASCII time-sharing file specified by ascfil is converted to a standard system format BCD file on the permanent file specified by bcdfil. This follows the question/answer sequence that is initiated by this command if the ASCII file does not contain first line reformatting information. Both ascfil and bcdfil may be simply a file name or a full file description, as required. (See definition of filedesc under RUN command.) The ascfil field may specify the current file by an asterisk.



- ASCASC filedescr 1; filedescr 2

This command, issued under Series 6000 FORTRAN and other time-sharing language systems, causes the translation of a time-sharing format ASCII file to a standard system format ASCII file or vice versa. In both translations, file 1 is converted to the format required in file 2. If file 1 is in time-sharing ASCII format (logical record type 5), the file is read and converted to the word-oriented standard system ASCII format for file 2. File 2 may then be used as input data for the language system. If file 1 is in standard system ASCII format (logical record type 6), the file is read and converted to the character-oriented time-sharing ASCII format for file 2. File 2 may then be listed at a terminal.

- BCDASC

BCDASC bcdfil; ascfil (BCD-to-ASCII)

The standard system format BCD permanent file specified by bcdfil is converted to an ASCII time-sharing file on the predefined permanent file specified by ascfil. This follows the question/answer sequence that is initiated by this command. Both bcdfil and ascfil may be simply a file name or a full file description, as required. (See definition of filedescr under RUN Command.) The ascfil field may specify the current file by an asterisk.

- BPUNCH and BPRINT

BPUNCH ascfil (Batch Punch)

BPRINT ascfil (Batch Print)

The contents of the ASCII time-sharing file specified by ascfil is converted to BCD and is punched or printed, respectively, at the central computer site. This follows the question/answer sequence that is initiated by these commands, if the file does not contain first line reformatting information. These commands allow the user to create card backup for his time-sharing files or to list long files on a high speed printer. The ascfil field may be simply a file name or a full file description, as required. (See definition of filedescr under RUN Command.) The ascfil field may specify the current file by an asterisk.

Since a batch dimension bulk media conversion (BMC) job is implied by these commands, batch \$ IDENT card information is requested.

- FDUMP

Permits inspection and maintenance of permanent files at remote terminals. Refer to the description of the FDUMP subsystem in Section IV.

- SCAN

Allows the scanning of batch output stored on a permanent file. Refer to the description of the SCAN subsystem in Section III.

- JOUT

JOUT snumb

Permits manipulating from a time-sharing terminal the output of certain types of batch jobs. Refer to the description of the JOUT subsystem in Section VII.

RUN or PRINT Question/Answer Sequence

If the file to be converted does not contain first line reformatting information immediately following the RUN or PRINT command, a fixed sequence of questions is asked by the system concerning the format and desired disposition of the file.

Response to CARD FORMAT,.....?

The user responds to the first part of the question, CARD FORMAT, DISPOSITION? with ASIS, STRIP, MOVE, or NORM. (The short form defined in parentheses may also be used.)

ASIS (A)

The user wants the file passed to batch just as it exists. In this case, each line of terminal input is converted to BCD and put in card format. The content of the first character position of the terminal line goes in the first column of the card, unless it is a tab character. Use of ASIS implies either that the file does not contain initial line numbers or that the user desires them to appear in this batch input file.

STRIP (S)

This specifies that the user has employed line numbers when preparing his file and wants the line number stripped from each line before converting to card format. When this option is used, the first nonnumeric character of a terminal line, if not a tab, goes in column 1 of the card. (Line numbers in the source file are unaffected.)

MOVE (M)

This option states that the user has employed line numbers when preparing his file and wants the line numbers moved to the sequence field (columns 73-80) of the card. When this option is used, the first nonnumeric character of a terminal line, if not a tab, goes in column 1 of the card.

NORM (N)

This option implies the MOVE option and specifies that the normal tab character (the colon) and tab settings (8,16,32,73) have been employed in building the file. Following a response of NORM, the TAB CHARACTER AND SETTINGS? query is omitted.

The NORM format may also be used to define a character other than the colon to be used as a tab and may permit multicard lines by means of a character used to indicate the end of a card. These options are indicated in a parentheses enclosed field following NORM. Either character may be omitted, as shown in the examples below.

Examples:

NORM(;/,/)

The semicolon is the tab character and end-of-card is indicated by /. That is, a line such as the following has been employed in the file:

70;LDA;A;/STA;B;/TRA;C

NORM(,;)

The colon is the tab character by default, and the semicolon is the end-of-card character.

NORM(;)

The semicolon is the tab character; no end-of-card character is given.

Note: A line number is from one to eight numeric characters immediately followed by a nonnumeric character (including blank), the numerics being the first nonblank characters in the line:

Ø...Ønnnnnnnnxccc.....c

Where: Ø...Ø - Optional initial blank.

nnnnnnnn - Numeric characters.

X - Nonnumeric character.

cc.....c - Any characters.

For STRIP, MOVE, and NORM, if the user wants a numeric in column 1 of the card image and line numbers exist in the source file, a pound sign (#) character immediately following the line number causes the character following it to go into column 1.

For example, 220#123 would result, upon a MOVE conversion, in the card numbered 220 having 123 in columns 1, 2, and 3. If the user desires that a pound sign go into column 1 of the card image, two pound signs following the line number will achieve that effect. For example, 220## would result, upon a MOVE conversion, in the card numbered 220 having a pound sign in column 1.

Response to DISPOSITION?

Having entered a response to the CARD FORMAT portion of this question, the user answers the DISPOSITION portion with one of the following entries (note that each is preceded by a comma).

...,WAIT	-- Wait for job termination.
...,TALK	-- Enter conversational mode. (Use may be restricted by master user. See discussion of TALK.)
...,URGC(xx)	-- Assign specified initial urgency to this job.
...,JOUT	-- Save all implied files for examination by JOUT subsystem.
...,ROUT(xx)	-- Direct output to station xx. Only one ROUT entry is permitted.

If, instead of entering a reply to DISPOSITION, the user follows his card format entry with a carriage return, the job is initiated, and the system returns to build mode for further CARDIN command entries.

WAIT (W)

This response indicates that the user desires to wait at this point for a status indication sent upon completion of the job in the batch environment. Normal termination is indicated by the message

NORMAL TERMINATION

Any abnormal termination is indicated by an appropriate plain text message. (See abort/delete messages in Comprehensive Operating Supervisor reference manual.)

Following the termination message, the system returns to build mode for further CARDIN command entries.

TALK (T)

This response implies that the batch job includes execution of a program containing conversational (direct access) input/output -- either user-implemented direct access I/O or, for example, the RBUG routine (see Section V). Following the actual submission of the job by CARDIN, the user's terminal is placed in direct access connection with the submitted program (by SNUMB). When the job terminates, the user receives a normal or abnormal termination message and returns to build mode, as with WAIT.

Permission from the master user must exist in the Time-Sharing System before a specific user can employ the TALK disposition. (Granting of permission to use TALK implies permission to use all CARDIN functions.)

URGC(xx) (U)

This response may be used only when the RUN command has been previously entered. The response indicates that the user wants to assign initial urgency xx to the job. The assigned urgency must not be greater than the maximum allowed, or the system will reply with the message ILLEGAL URGENCY after the next carriage return and follow it with the CARD FORMAT, DISPOSITION? question.

If xx is not specified, maximum allowable urgency is automatically assigned.

This disposition response may be specified by itself or along with JOUT and/or WAIT or TALK.

JOUT (J)

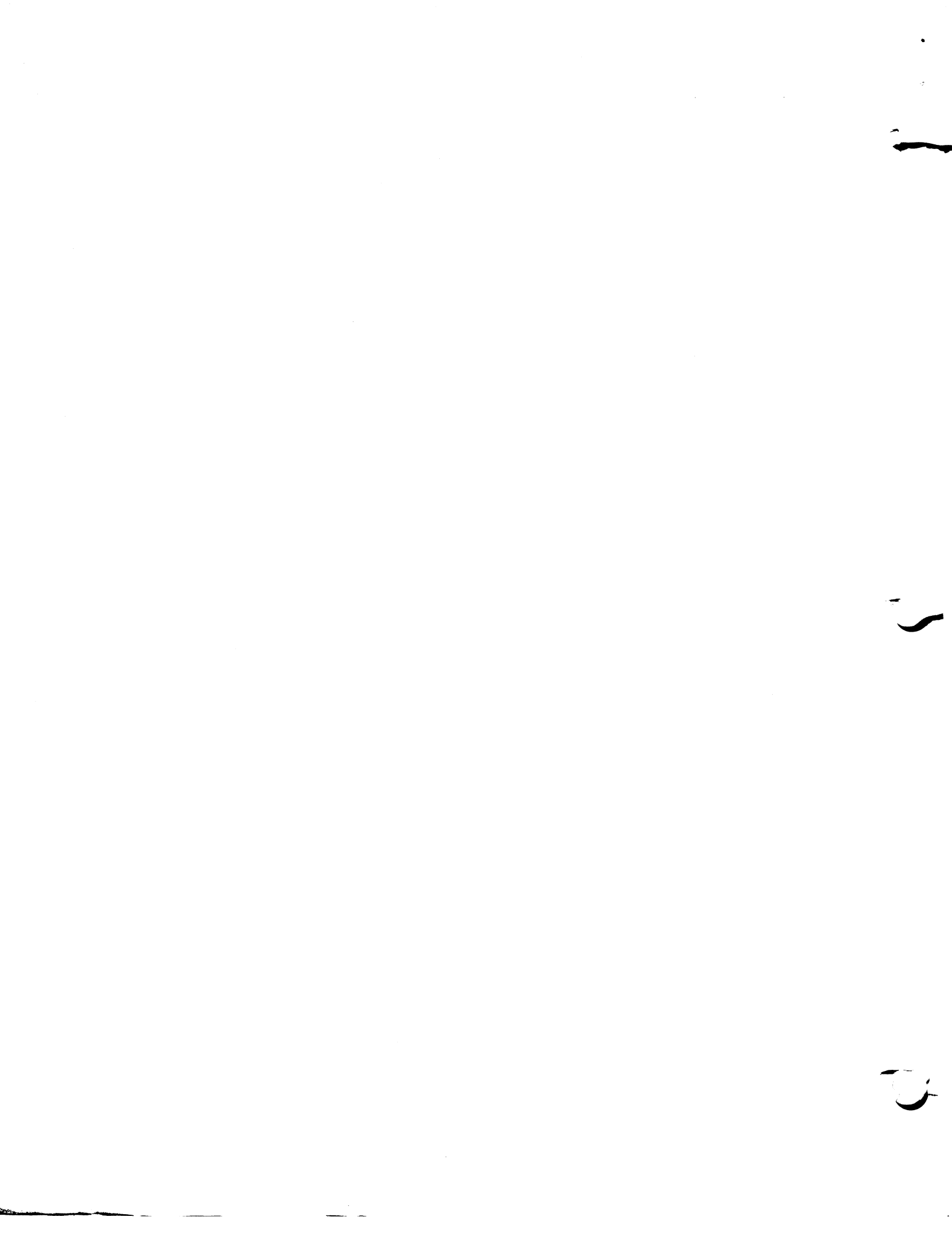
This response may be used only when the RUN command has been previously entered. The response requests that all implied files be saved so that they may be examined using the JOUT subsystem.

*

This disposition response may be specified by itself or along with URGC and/or WAIT or TALK.

ROUT(xx) (R)

This response may be used only when the RUN command has been previously entered. The response indicates that the user wishes the implied files generated by the program execution to be directed to the specified two character remote station. This response to DISPOSITION may be specified by itself or along with any other disposition except JOUT. Only one ROUT entry permitted.



Following are examples of responses to the CARD FORMAT, DISPOSITION? question. Note these points:

1. A response must be entered for CARD FORMAT, and this response must be the first item.
2. The responses may be represented by their initial letters.
3. The DISPOSITION responses may be in any order.
4. The responses are separated by commas, with no embedded blanks.

Examples:

MOVE	
ROUT(BC)	
MOVE,TALK	
R(AD)	(same as as ROUT(AD))
M,T	(same as MOVE,TALK)
NORM(;/),W	(W = WAIT)
N(;/),URGC(35),J	(N = NORM, J = JOUT)
N,U(35),J,T	(same as NORM, URG(35),JOUT,TALK)
A,J	(same as ASIS,JOUT)
S,U,JOUT	(same as STRIP,URGC,JOUT)
A,R(DD)	(same as ASIS,ROUT(DD))

Response to TAB CHARACTERS AND SETTINGS?

This question is asked only if NORM has not been specified in response to CARD FORMAT, DISPOSITION?

When the file is being prepared, the user can employ tab characters to format his input instead of the required blanks. Any character except the pound sign (#) can be used to indicate a tab, or the regular tab key can be used. Delimiters used for tab should be unique. When responding to the above question, the user may specify a tab character and up to 20 tab settings. All fields are separated by commas. For example, if the user employed an ampersand (&) as a tab character and desired successive tab values of 8,16, and 40 in his card-image input file, he would respond with &,8,16,40.

If the user employed the standard character and settings (:,8,16,32,73), he may respond simply with NORM or N.

If the user did not use any tabs in preparing his input, he would simply respond to the above question with a carriage return.

The user may employ multiple tab characters (tab entries separated by a semicolon) in his file--one for language statements and another for control cards, for example.

&,8,16;+,12,32,40

where the tab characters are & and +.

A maximum of seven different tab characters and a total of 35 individual tab settings can be specified within the limits of one 80-character input line of the file.

ASCBCD Question/Answer Sequence

The questions LABELS? and, conditionally, TAB CHARACTERS AND SETTINGS? are asked when the ASCBCD command is given, if the file ascfil does not contain first line reformatting information. The responses are for the most part identical with those given in the RUN and PRINT question/answer sequence. Where the responses are identical, the explanation of them is not repeated, as the meaning is the same.

Response to LABELS?

- MOVE
- STRIP
- ASIS
- NORM

- abcde(i,j)₁; abcde(i,j)₂; ...; abcde(i,j)_n

This option implies that the line numbers in the file are to be moved to the BCD label field (columns 73-80) and that the alphanumeric prefix abcde_i is to be inserted -- left justified -- in the label field as specified by the line numbers (i,j)_i;

abcde_i represents any alphanumeric prefix, to a maximum of five characters

i_i represents the initial line number

j_i represents the final line number to which abcde_i is to be prefixed

The (i,j) field is mandatory; the conventions on its use are as follows:

(i,j) = (i,j)

(,j) = (0,j)

(i,) = (i,99...9)

Multiple sets of prefix specifications may be given, semicolon separated, as shown in the general form above. If an interval of line numbers is found in the file that has not been specified by the user, only the line number will appear in the label field.

The prefix will overlay leading significant digits of the line number, if the line number becomes too large to be accommodated.

Response to TAB CHARACTERS AND SETTINGS?

If the response to LABELS? was not NORM, the question TAB CHARACTERS AND SETTINGS? is asked. The possible responses are:

- carriage return (null response) -- Indicates that no tab characters are employed in the file.
- t,s₁,s₂,...,s_n -- Indicates a single set of tab character (t) and settings (s_n) have been used, as in the RUN or PRINT responses. The value of s_n must be less than 81.

- tab-set₁;tab-set₂;...;tab-set_n -- Indicates that multiple sets of tab characters and settings have been used in the file -- as in the RUN or PRINT responses--where tab-set has the same form as the single-set specification shown above.

A maximum of seven different tab characters and a total of 35 individual tab settings can be specified within the limits of one 80-character input line of the file.

BCDASC Question/Answer Sequence

The questions LINE NUMBERS? and TAB CHARACTER AND SETTINGS? are asked when the BCDASC command is given. The first question requests information concerning what line numbers, if any, are desired in the converted ASCII file (ascfil). The second question requests information concerning a single tab character.

The responses to LINE NUMBERS are:

- carriage return (null response) -- Do not create line numbers for the ASCII file.
- MOVE -- Use the sequence-number digits of the label field in the BCD file as line numbers for the ASCII file.
- AUTO -- Create line numbers for the ASCII file, starting with 0010 and incrementing by 10.
- AUTO n,m -- Create line numbers for the ASCII file, starting with n and incrementing by m. If either n or m is null (AUTO n, or AUTO ,m), the missing variable will have the standard value shown under AUTO, above.
- ASIS,n -- Convert from BCD to ASCII n columns of the BCD record.

Response to TAB CHARACTER AND SETTINGS?

The response is similar to the response to the same question under ASCBCD. However, only one tab character and its settings may be entered. Blanks in the ASCII file will be eliminated and the tab character inserted as directed in the response. Thus, the created ASCII file will appear as though it was entered from a terminal with the use of tab characters.

BPUNCH or BPRINT Question/Answer Sequence

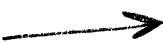
Both the BPUNCH and BPRINT commands ask the same series of questions:

\$ IDENT?

LABELS?

TAB CHARACTERS AND SETTINGS?

The responses to the first two questions are the same as those for the ASCBCD command. The response to \$ IDENT is identical in form and content to the variable field information that would be required on the \$ IDENT control card for a comparable batch job. (A batch BMC job is actually initiated).



First Line Reformatting Information in the File

To minimize the subsystem/user interface involved in the use of the RUN, PRINT, and ASCBCD commands, the required reformatting and disposition information may be contained within the file to be converted as its first line and identified by the special marker characters ##. If this information exists in the file submitted, the normal question/answer sequence is bypassed.

The general format of the first line reformatting information is as follows:

line-number##first-response second-response

Where:

- line-number - Optional line number.
- first-response - An appropriate full response to CARD FORMAT, DISPOSITION? in the case of RUN; to CARD FORMAT? in the case of PRINT; or to LABELS? in the case of ASCBCD.
- second-response - An appropriate response to TAB CHARACTER AND SETTINGS? -- if required -- in the case of all three commands, separated from the preceding first-response by one blank.

The form of the information represented by first-response and second-response is precisely that which would be required by the question/answer sequence of the respective commands.

An example of first line control information for the RUN command is as follows:

```
10##MOVE, WAIT ;, 8, 16, 32
```

Note that the semicolon following the single blank specifies the tab character and is not a separator.

Another example:

```
10##NORM
```

Note that the second-response portion is null in this example because of the use of NORM as the first-response.

\$ SELECTA Card Option

With \$ SELECTA cards the user may merge two or more time-sharing input files into one batch file. The select function is initiated when a \$ SELECTA card is detected in the input stream. The format is:

```
1      8      16
-----
$      SELECTA  filedescr
```

Where: filedesc applies to the file selected for merging with the current file.

Upon encountering this card, CARDIN stops processing input from the current file and begins processing the selected file. When an EOF is encountered on the selected file, processing reverts to the previous file. The \$ SELECTA card is deleted from the input stream by CARDIN and does not become part of the batch input.

Restrictions on the \$ SELECTA card option are as follows:

- Nesting of a select activity within another select activity is permitted to a depth of 10.
- The contents of the selected file must be in time-sharing format.
- No test will be made for first line reformatting information on selected files.
- A \$ ENDJOB card detected in a selected file will be bypassed.

The following terminal printout is an example of the use of the CARDIN subsystem, utilizing the LIST and PRINT control commands and the SCAN subsystem. The text enclosed within brackets is not part of the printout but has been added to explain salient features.

SYSTEM ? CARDIN
OLD OR NEW-OLD PASSRC
READY

*LIST ←----- List the input file.

010\$;IDENT;VXA,JANEDOE
015\$;USERID;JANEDOE\$PASSWORD
020\$;NOLIB
030\$;OPTION;NOSETU,SAVE/PX,NOGO
040\$;LOWLOAD
050\$;ENTRY;BEGIN
060\$;GMAP;NDECK
070;TTL;SUBSYSTEM TO LIST AVAILABLE FILE TABLE
080;LODM;.G3TSS
090;.SSDRL
095;BSS;36;REQUIRED RESERVE SPACE FOR TS
100;SYMDEF;BEGIN
110BEGIN;DRL;PASAFT;READ THE AFT
120;ZERO;LOCA,0
130;LXL2;LOCA;GET FILE COUNT
140;TZE;END;NO FILES...
150;EAX1;LOCA+1
160AGAIN;LDA1;0,1;PUT NAME IN LINE
170;STAQ;NAME
180;LDAQ;TAL1;REFRESH THE TALLIES.
190;STAQ;TEMP
200;DRL;KOUT;WRITE THE FILENAME.
210;ZERO;TEMP,CHAR
220;EAX1;2,1;INCREMENT LOCA POINTER
230;EAX2;-1,2;DECREMENT COUNT
240;TNZ;AGAIN;GET NEXT NAME
250END;DRL;RETURN
260;REM
270LOCA;BSS;25;AFT BUFFER
280TAL1;TALLY;TEMP+1,1;OUTPUT TALLY
290;TALLYB;NAME,8
300CHAR;OCT;015012177177;CR,LF,NULL,NULL
310NAME;ASCII;2,
320;END

330\$;PRMFL;H*,R/W,R,JANEDOE/PASFIL ←

Place program in system loadable format on permanent file PASFIL.

340\$;PRMFL;P*,R/W,L,JANEDOE/PASLST ←

Place system output file on permanent file PASLST for use by SCAN.

350\$;ENDJOB

READY

* ←----- Indicates BUILD mode.

*PRINT----- Print the file in its extended format.
 CARD FORMAT ?
 MOVE
 TAB CHARACTER AND SETTINGS?
 ;,8,16,32

08/02/68 12.92

```

010 $ IDENT VXA,JANEDOE
015 $ USERID JANEDOE$PASSWORD
020 $ NOLIB
030 $ OPTION NOSETU,SAVE/PX,NOGO
040 $ LOWLOAD
050 $ ENTRY BEGIN
060 $ GMAP NDECK
070 TTL SUBSYSTEM TO LIST AVAILABLE FILE TABLE
080 LODM .G3TSS
090 .SSDRL
095 BSS 36 REQUIRED RESERVE SPACE FOR TS
100 SYMDEF BEGIN
110 BEGIN DRL PASAFT READ THE AFT
120 ZERO LOCA,0
130 LXL2 LOCA GET FILE COUNT
140 TZE END NO FILES...
150 EAX1 LOCA+1
160 AGAIN LDA1 0,1 PUT NAME IN LINE
170 STAQ NAME
180 LDAQ TAL1 REFRESH THE TALLIES.
190 STAQ TEMP
200 DRL KOUT WRITE THE FILENAME.
210 ZERO TEMP,CHAR
220 EAX1 2,1 INCREMENT LOCA POINTER
230 EAX2 -1,2 DECREMENT COUNT
240 TNZ AGAIN GET NEXT NAME
250 END DRL RETURN
260 REM
270 LOCA OBSS 50 AFT BUFFER
280 TAL1 ETALLY TEMP+1,1 OUTPUT TALLY
290 TALLYB NAME,8
300 CHAR OCT 015012177177 CR,LF,NULL,NULL
310 NAME ASCII 2,
320 END
330 $ PRMFL H*,R/W,R,JANEDOE/PASFIL
340 $ PRMFL P*,R/W,L,JANEDOE/PASLST
350 $ ENDJOB

```

*

```

*RUN<----- Prepare to send input file to batch.
      SNUMB # 0122T<----- SNUMB Assigned to this job.
CARD FORMAT,DISPOSITION ?
M,W<----- CARDIN options MOVE and WAIT
TAB CHARACTER AND SETTINGS?
; ,8,16,32<----- As used in input file.
      NORMAL TERMINATION.----- Job termination from batch.
*SCAN PASLST<----- Call SCAN subsystem.
FORM?GMAP<----- Identify job type
006 ERRORS
EDIT?Y
?FLAG<----- List assembly errors.
M 2 LODM .G3TSS 00000080 #0013
O 000051 000000 0000 11 000 11 AGAIN LDA 0,1
  PUT NAME IN LINE 0000016
O #0023
A 000052 000117 7570 00 010 12 STAQ NAME 00000170 #0024
U 000054 000000 7570 00 000 14 STAQ TEMP 00000190 #0026
U 000056 000000 000116 001 16 ZERO TEMP,CHAR 00000210 #0028
U 000114 000000 0001 00 000 23 TALL TALLY TEMP+1,1
  OUTPUT TALLY 000002
80 #0035
TEMP #0061

```

```

?DONE<----- Leave. At this point the assembly
                errors should be corrected and the
                job resubmitted.

```

```

To place the object program (H*)
on file PASFIL, a $ EXECUTE card
would be added to the deck setup
of the source program.

```

SECTION III
SCAN SUBSYSTEM

SCAN FUNCTION

The SCAN subsystem permits scanning output from the following types of batch jobs at a Time-Sharing System terminal:

- Those submitted via CARDIN.
- Those submitted via remote batch facility of the General Remote Terminal Supervisor (GRTS).
- Those submitted at the central site.

The output to be scanned must have been saved on a permanent file.

The output file to be scanned will fall into one or more of the following format categories:

GMAP assembly listing (GMAP)

FORTRAN compilation listing (FORT)

COBOL compilation listing (COBOL)

General Loader output: load map, etc. (LOAD)

User-generated output (USER) -- that is, not in any standard format

Special command verbs and other facilities are provided for operating on these forms of output.

Essentially, the SCAN subsystem allows perusal, by means of highly selective listing, of any type of batch output, with special features that facilitate efficient inspection of commonly desired items of information -- for example, flagged lines in a GMAP assembly.

SCAN OPERATION DESCRIPTION

The SCAN question/answer sequence is as follows:

Question: SYSTEM?

Response: SCAN

Question: FILE?

Response: filedescr or filedescr;n

where filedescr is the file description and n is the number of segments to be scanned in sequence when the file is segmented by end-of-file marks.

Question: FORM?

Response: One of the following, indicating the format of the file:

GMAP - subsystem initially responds with the number of assembly errors detected.

FORT - subsystem initially responds with the number of compilation errors detected.

COBOL - subsystem initially responds with the number of compilation errors detected.

LOAD - subsystem initially responds with the number of errors detected by the loader.

USER - subsystem initially responds with the question, CODE? The user's answer to this question is from one to five characters to be used by the subsystem as a line code in searching the user's output text, or the answer may be a null response (carriage return only). The matching line code in the file is assumed to be found left-justified in character positions 2-6 of each line; the FIND and PRINT verbs ignore all lines in user format output that do not contain the current line code (if line code is given). The initial line code given may be replaced by means of the CODE verb (see the following section). The null response is equivalent to ignore line codes, if any.

Line codes can serve to mark lines of the same type or category in the output text. Use of this feature implies that the user has prepared his output with this feature in mind. It is not an essential part of the system, however; and SCAN may be used with output files not having line codes. But if the user has control of his output format, line codes can facilitate the use of SCAN.

A matching mask is created automatically, so that a hierarchy of line codes can be established. For example, given lines with E1, E2, E1.5 as line codes, a code E will find all lines, code E1 will find lines E1 and E1.5, etc.

Question: EDIT?

- Response: (1) Y or YES - for multiple blank suppression.
- (2) N or NO - for printing of multiple blanks as is.
- (3) A null response is assumed to be a NO response.

Question: ?

Response: a SCAN verb

Note the four level structure in the question/answer sequence:

1. SYSTEM?
2. FILE?
3. Format (FORM?)
4. Verbs(?)

The transition from a lower level to the next higher is made by responding to one of these questions with a null line (carriage return only). For example, if a user is scanning a file created by a FORTRAN compile and execute, he may wish to use the formats FORTRAN and LOAD alternately. When he has scanned the FORTRAN listing and wishes to proceed to the load map, or vice versa, he responds to the initial question mark (verb level) with a null line; SCAN then asks for a new format (FORM?).

SCAN VERBS

The SCAN verbs are given below, with their arguments. Only the first four characters are significant; that is, PRINT can be abbreviated as PRIN, and BATCH can be extended to BATCHJOB. A blank must separate the verb from its argument.

First letter abbreviations of the following SCAN verbs are also acceptable: PRINT, FIND, SPACE, BACK, ERROR, UNDE, and CODE.

FIND

FIND /literal string/;n

/ represents any desired delimiter chosen by the user. The literal string is a pattern of characters to be searched for; n represents the nth occurrence.

The FIND verb positions an implied pointer to the nth line containing the literal string (beginning with the line currently pointed to). If n is not given, 1 is assumed. If no literal string is given, all lines are assumed to match; for example, FIND; 2 is equivalent to SPACE 1 (see SPACE verb).

The FIND verb also accepts all standard Text Editor argument forms.

PRINT

PRINT n

n is the number of lines to be printed.

PRINT causes a printout at the terminal of the next n lines, beginning with the current value of the pointer. If n is not specified, only one line will be listed. Lines will be listed along with their automatically generated line numbers for future reference in LINE verb commands. Line codes are effective, if in force.

Note that the PRINT verb does not move the pointer, so that a subsequent FIND verb will begin its search with the first line that was listed.

If n consists of the literal ALL, all lines with a matching code will be printed, from the current line to end-of-file.

The appearance of only a number in the listing of a line indicates that the corresponding record is a slew record.

The PRINT verb also accepts all standard Text Editor argument forms.

LIST

List n

LIST is synonymous with PRINT in all respects.

Batch not available if SCAN entered at the JOUT.
BATCH — use DIRECT ONL at Function level.

The system asks STATION CODE? The user replies ab or simply a carriage return, where ab is the station code of a remote-batch terminal.

The system then asks, \$ IDENT?, to which the user replies with the variable field of his batch \$ IDENT card.

The BATCH verb initiates a Bulk Media Conversion (BMC) job which will transfer the entire contents of the file to remote printer ab. If the station-code reply is null, the output will be printed at the central site.

SPACE

SPACE n

Spaces the pointer ahead n lines. If n is not specified, the pointer will advance one line. An attempt to position the file beyond its end will result in the file being positioned at its beginning, with a warning message to the terminal (EOF). The SPACE verb operates independently of line codes.

BACK

BACK n

Spaces the pointer back n lines. If n is not given, the file will be rewound (pointer moved to line 1). The BACK verb operates independently of line codes.

LINE

LINE n

As each line is listed, an automatically generated line number will be typed with it. The LINE verb repositions the pointer to the specified line number, n. (The line number used need not have been printed prior to being referred to.) Line codes are not effective.

The above are general purpose verbs. The following special verbs are useful for manipulating specific formats:

ERROR

ERROR n

Requests a list of the next n error printouts of the form corresponding to the output format in question. (For example, if the format is GMAP, flagged instructions. If USER, .FXEM messages will be listed.) The absence of n implies all such messages.

UNDEFINED

UNDE

This command (no argument) is used while scanning GMAP assemblies to list all undefined symbols.

FLAG

FLAG x

Lists all lines of a GMAP assembly having the error flag specified by x (A, U, M, O, etc.). The absence of a specific error tag implies that the user wishes a list of all flagged instructions.

LOAD MAP

LOAD

Prints out an abbreviated load map. Only primary SYMDEFs are listed, and library routines are omitted.

CODE

CODE abcde

Employed with the user format to change the line code. The argument abcde is a one- to five-character code from the Series 600/6000 BCD character set. A null argument turns off line codes. That is, all line codes are accepted until the CODE verb is used to resume with a valid line code.

EDIT

Returns the subsystem to EDIT?

DONE

Returns the subsystem to SYSTEM?

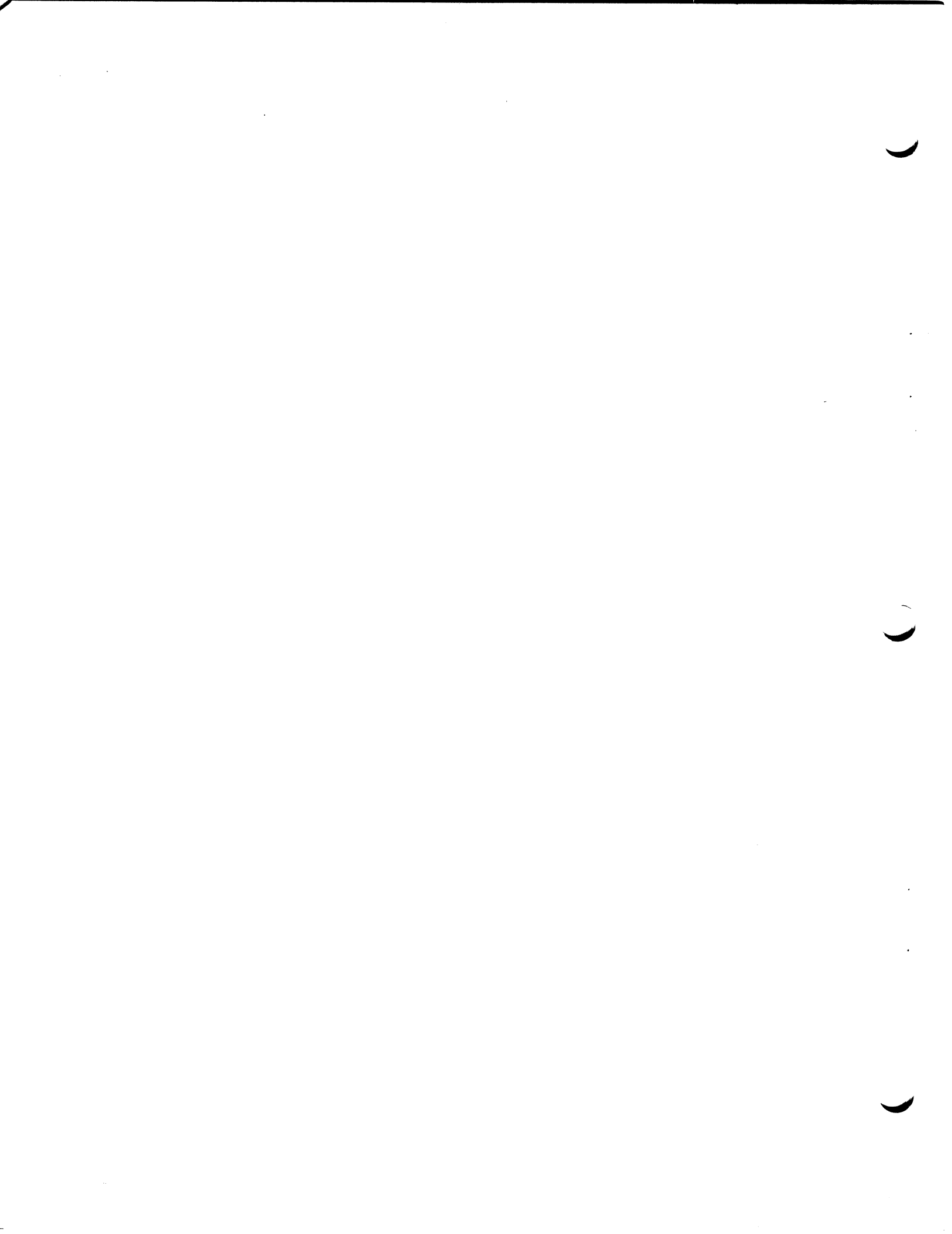
BYE

Terminates user's current session with the Time-Sharing System.

REM

REM text

Produces the text as remarks line on printer log; otherwise ignored by SCAN.



SECTION IV
FDUMP SUBSYSTEM

FDUMP FUNCTION

The FDUMP subsystem (file dump and correction) is a remote terminal file inspection and maintenance facility for permanent files in the GCOS file system. These files reside on a shared-file type of device such as a disk storage unit. The files may be generated in either batch, remote/batch, or time-sharing environments.

With the FDUMP subsystem, the user at a remote terminal can manipulate the content of any permanent file (to which he has access) as follows:

- Read into core storage any given 320-word block of the file.
- Scan the block content, by dumping all or nonzero portions of the block at his terminal, using the snapshot function.
- Patch selected locations in the block, using the patch function.
- Rewrite the corrected block back onto the file.
- Copy the file.

This facility is provided primarily for those situations where the user needs or wishes to inspect the actual binary content of a file. It is, therefore, mainly a debugging tool. The file is dumped and patches are specified in octal form. Word locations within a block are also specified in octal, beginning with zero.

FDUMP OPERATIONAL DESCRIPTION

The user selects FDUMP in response to the subsystem selection question SYSTEM? or as a command under the CARDIN subsystem.

The first-level question asked after the subsystem is called is:

FILE NAME?

The permissible responses are:

carriage return - return to the SYSTEM? level.

filedescr - Specified file is accessed, if possible, and treated as a linked or random file, depending on how it is defined.

filedescr;L - Specified file is accessed, if possible, and treated as a linked file.

filedescr;R - Specified file is accessed, if possible, and treated as a random file.

Caution: filedescr must not be the same name as another even if the duplicate filedescr is in another catalog when the FUNCTION is copy (C).

The second-level question is:

BLOCK TO BE READ -

The permissible responses are:

carriage return - return to the FILE NAME? level.

n - the block specified by the block serial number n will be read into an internal buffer. The Copy function (see following function description) requires a dummy response of 1.

If the block serial number is outside the limits of the file, the error message BSN OUTSIDE FILE LIMITS is given, and BLOCK TO BE READ will be repeated. If the block serial number is within the current file size but the implied block was not written on the file, it will be read but will contain garbage data not pertaining to that file.

The third level (and final) question is:

FUNCTION? - (this question is repeated upon return from any of the FDUMP functions.)

The permissible responses are:

carriage return - return to BLOCK TO BE READ.

Sloc - Snap the specified (octal) location.

Sloc-loc - Snap the field specified by (octal) location-location (from-to).

Sloc,n - Snap n words starting with the specified (octal) location.

Ploc data - Patch the specified (octal) location with the specified (octal) data.

W - Write the corrected block back into the permanent file.

C filedescr - Copy the complete file onto another file specified by filedescr. (Refer to FDUMP Copy Function Details paragraph.)

D - Done. Return to SYSTEM? or to CARDIN.

FDUMP Error Indication and Replies

1. When the named file cannot be accessed, FDUMP replies

CANNOT ACCESS FILE filename

and returns control to the calling level (SYSTEM? or CARDIN).

2. When the block serial number given is either zero or is a number larger than the possible number of blocks in the file, the error message is:

BSN OUTSIDE FILE LIMITS

BLOCK TO BE READ is then repeated.

For linked files, block size is assumed to be 320 words; the first block serial number is 1. Random files are positioned by multiples of 40 words DSU200 or 64 words (MDS200 drum and mass storage devices), beginning with block 0. However, they are read in blocks of 320. Therefore, one read makes available eight contiguous blocks of 40 words or five contiguous blocks of 64 words.

3. When the system receives a bad hardware status, FDUMP replies:

51 FILE filename -- I/O STATUS xx

FUNCTION?

A partial block may have been Read and may be correctable by use of the S, P, and W functions. If none of the block appears to have been Read, answer with a carriage return to repeat the BLOCK TO BE READ question. Also, verify the block serial number that was specified.

4. When parameters are incorrect in form for the S, P, or W functions, FDUMP will reply

INVALID INPUT-RETYPE

FDUMP Copy Function Details

*

The copy is a physical copy; that is, it does not stop at logical end-of-file but continues to the file length defined in the file system as current size.

If the copy file is smaller than the size defined for the file to be copied, FDUMP grows the copy file to the necessary size.

Note that filedescr, specifying the copy file, may be simply a file name or may be a catalog/file string, but must not have the same filename even if duplicate filename is under a separate catalog.

At completion of the copy, the user will be returned to the FILE NAME? level.

SECTION V
CONVERSATIONAL DEBUG

RBUG GENERAL DESCRIPTION

The Conversational Debug Routine (RBUG) is a dynamic debugging tool for batch programs initiated through the CARDIN subsystem. The user of RBUG should be familiar with the normal DEBUG routine and its associated control cards as described in the General Loader reference manual.

When the control card \$ USE RBUG are used, the normal DEBUG routines are superseded by RBUG; control and information are then obtained in exactly the same manner as in the standard routine. By the use of the BREAKPOINT pseudo-variable on the General Loader DEBUG control card, the user may gain control at specified points in his program. At those points, the user may:

1. Obtain snapshots of locations and registers.
2. Replace the contents of locations and registers.
3. Insert further breakpoint locations or delete breakpoints he has placed earlier (breakpoints specified in the control card or FORTRAN statement may not be deleted).
4. Continue the run at the interrupt point or another point; or stop the run, either by normal termination or abort.

Because of the way the loader builds DEBUG tables, variables are printed for each debug location in reverse order. Therefore, if a breakpoint is desired, the BREAKPOINT pseudo-variable should be the first in the list, so that a typeout of the requested variables occur before the breakpoint is executed. When a point is reached in the execution of the program for which conditions specified by the control card or DEBUG statement are met, output (if any) is typed at the user's terminal. In addition, if there has been a breakpoint specified, execution of the program is suspended and a question mark printed, asking for instructions. After each instruction has been acted upon, a question mark is typed, requesting further instructions until the user asks to run, terminate, or abort.

RBUG instructions, listed in the following section, are all composed of a single letter command followed in most cases by an argument. The argument follows the standard DEBUG convention that all variables are considered to be single dimension arrays, with the first element being element number 1. Therefore the general form of an argument is:

Name (i, j, k)

where the subscripts have the meanings from, to, and increments of. For example A(3,9,2) has the meaning from A(3) to A(9) in steps of 2 -- that is, A(3), A(5), A(7), A(9).

The name is that of a variable within the subroutine at which the program was interrupted or of a variable in common. If the name contains only numerics between 0-7, it is considered an octal location. Therefore 9(3) would be interpreted as the location of statement 9 plus 2, but 7(9) would be interpreted as octal location 17 (7 plus 8 = decimal 15 = octal 17). In order to enable reference to FORTRAN statements not containing an 8 or 9, the pound sign (#) is treated as an invisible alphabetic. For example, statement 23 can be referred to as #23, 23#, or 2#3.

Subscripts, where applicable, are optional.

RBUG INSTRUCTIONS

The following RBUG instructions are available to the user:

- A (ASCII) prints out in ASCII format the contents of the requested locations.
- B (Breakpoint) inserts a dynamic breakpoint at the desired location. When this point in the program is later reached, the user is notified, and has then the same instruction capability as he would have at a normal breakpoint. However, if the breakpoint location is specified octally, no symbol table is available; and therefore at this breakpoint only octal addresses may be used.
- C (Complex) prints out the contents of pairs of cells in complex format. Subscripts refer to pairs, not to individual locations.
- D (Double precision) prints out a double precision, floating point representation of the word pairs referenced. Subscripts refer to word pairs, as above.

- E (Erase breakpoint) eliminates a previously inserted dynamic breakpoint.
- F (Floating) prints out a floating point representation of the locations referenced.
- H (Hollerith) prints out a Hollerith (BCI) representation of the locations referenced.
- I (Integer) prints out a decimal representation of the locations referenced.
- L (Logical) prints out a logical representation (T or F) of the locations referenced.
- M (Modify) is used to change the contents of a register. The form of the instruction is Mr xxxx, where r represents the register to be modified, and xxxx is an octal representation of the new contents - right-justified and zero filled in the register. The permissible values of r are A, Q, E, I, or X0 through X7.
- O (Octal) prints out the contents of the referenced locations in octal notation.
- P (Patch) replaces the contents of the given location with a new value. General form of the instruction is Pname(i) xxxxxxxx. The value xxx... is right-justified and zero-filled.
- Q (Quit) - no argument. Causes the program being monitored to abort with reason code X1.
- R (Run) causes the program to resume at the instruction which was replaced with the breakpoint. If an argument is given (single location), the program is given control at that point.
- S (Starting address) is a single argument instruction. The location given as the argument is added to all subsequent locations given in octal. For example, having given SABC as an instruction, 023 prints out (in octal) cell 23(8) relative to location ABC. Locations specified symbolically are not affected by the starting address. The address may be changed at will by the user and may be nullified by a simple S (equivalent to S0, start at zero).

- T (Terminate) causes the program to terminate normally, by calling .FEXIT.
- W (Where) is used with a single argument. It prints out in octal the effective address of that argument.
- X (Display registers) is used without argument. It yields a panel dump of all registers. It may also be used selectively with the following forms: XA, XQ, XE, XI, or X0 through X7.

The user establishes direct access connection with his program either by the TALK option of CARDIN or by logging off from time-sharing and calling a GRTS phone number. He then requests to be connected with his SNUMB. For example, when submitting from CARDIN, the user is notified that he has been assigned SNUMB #0042T. He then answers BYE to the system question, calls GRTS, and requests \$*\$DAC0042T.

Appendix A contains a sample program in which use of RBUG is illustrated.

SECTION VI

CONVERSATIONAL FILE AND RECORD INPUT/OUTPUT

The File and Record Control routines include a conversational, or direct access, input/output capability, using the File and Record Control OPEN, GET, PUT, and CLOSE calls, and a direct access file control card, \$ DAC. The \$ DAC control card allows the user to specify one or more of his program files as conversational, to be connected directly to a remote terminal device.

An additional control card, \$ USE .RTYP, must be used in order to activate the required File and Record Control routines (.GOPNR, .GCLSR, .GGETR, .GPUTR).

Appendix A contains a sample program in which use of File and Record Control conversational routines are illustrated.

CONTROL CARDS FOR FILE AND RECORD CONTROL I/O

- \$ DAC

The Direct Access (DAC) file control card has the following format:

```
$ DAC fc (,d)
```

Where: fc - A two-character alphanumeric file code identifying a direct access file referred to in the program (or FCB).

d - An optional one-character logical unit designator. If the ,d field is omitted, blank is assumed.

The logical unit designator is provided primarily for future system implementations. This field normally is omitted; if the associated terminal device is to be connected via the TALK option in CARDIN -- the standard case -- this field must be omitted (or d must be blank).

- \$ USE .RTYP

The \$ USE .RTYP control card must appear in the job control deck to direct loading of the correct File and Record Control routines. If this card does not appear, any reference to a file defined as direct access causes a program abort with a code of NT.

APPLICABLE I/O CALLS

The following File and Record Control calls are applicable to a direct access, or conversational, file:

- OPEN -- Connect a remote terminal device and open file.
- GET -- Read logical record from a remote terminal.
- PUT -- Write logical record to a remote terminal.
- CLOSE -- Close file and Disconnect a remote terminal device.

The calling sequences are standard, as described in the File and Record Control reference manual. The usage of the calls is also standard except as noted in the following functional descriptions.

- OPEN

This function attempts to connect a remote terminal device identifying itself by snumbd, where snumb is the sequence number of the calling program (job), and d is the logical unit designator (normally blank). If no such terminal requests a direct access connection within a reasonable time (on the order of several minutes), the FCB is marked file-not-present; and control is returned to the program. If a connection is made, the station code (line ID) of the terminal is placed in LOCSYM -4 (upper); and the file is marked present and open.

- GET

The terminal is sent the input-request message:

fc?

where fc is the current file code.

When input is received from the terminal, it is translated to BCD and stored as a standard system record in the user's buffer. If the record type is fixed, short input lines are blank filled to correspond to the record size specified in the FCB. Variable length records are blank filled to a minimum of 14 words.

A null input line (carriage return only) results in an end-of-file return, as will also a break or disconnect. On an end-of-file return, the user may determine the reason from the first status return word (LOCSYM -3, bits 6-11); the code is:

- 0 = null line
- 1 = break signal
- 2 = terminal disconnect

The carriage return null input does not set the end-of-file bit, thus communication with the terminal may continue.

- PUT

The output in the user's buffer is translated from BCI to ASCII and sent to the terminal. The use of this call carries the same restriction as is imposed on the WTREC call -- data must be present prior to the call. (This restriction does not normally apply to PUT for nonconversational files, where it can be used to reserve buffer space without moving data.)

A break signal or a disconnect during the call will cause the end-of-file status to be posted in the FCB, with status code 1 or 2. (See GET, above.) (End-of-file during a PUT on nonconversational files normally causes an abort.)

- CLOSE

The connected terminal is notified that the file is being closed, and the corresponding file is marked closed.

SPECIAL FILE AND RECORD CONTROL I/O CONSIDERATIONS

All required ASCII-to-BCD and BCD-to-ASCII translations are performed by the GET and PUT subroutines. The user's output buffer must therefore contain BCD data; his input buffer will contain BCD data following the GET call.

During a PUT call, a break or attention signal (normal means of interrupt from a terminal) or a user-initiated disconnect results simply in the posting of an end-of-file status in the FCB, rather than a program abort. It is the user's responsibility to check periodically for such occurrences (code 1 or 2 in LOCSYM -3, bits 6-11).

The optional logical unit designator in the \$ DAC control card allows multiple terminals to be connected simultaneously to one program (one terminal device per file). The terminals requesting connection are distinguished by snumbd₁, snumbd₂, ..., snumbd_n. A terminal connected via the TALK option of CARDIN implicitly carries a logical unit designator of blank (040 octal); the corresponding \$ DAC control card d field must be null. Any terminals to be connected to nonblank designated program files must be connected through the normal GRTS procedure: (1) dialing up a GRTS line and (2) a direct access connection request, \$*\$ DAC snumbd.

One terminal device may be connected to more than one program file (though the converse is not true). In this case the several \$ DAC file cards specify or imply the same logical unit designator (normally blank).

SECTION VII
JOUT SUBSYSTEM

JOUT FUNCTION

The JOUT subsystem permits manipulating output from the following types of batch jobs at a time-sharing terminal:

- Those submitted via CARDIN with a DISPOSITION response (after a RUN command) that includes J or JOUT.
- Those submitted via GRTS remote batch.
- Those submitted at the central site.

The output to be manipulated must be designated for a remote destination and must contain a \$ USERID control card.

The output file to be scanned will fall into one or more of the following format categories:

GMAP assembly listing (GMAP)

FORTRAN compilation listing (FORT)

COBOL compilation listing (COBOL)

Core dump (DUMP)

General Loader output: load map, etc. (LOAD)

User-generated output -- that is, not in any standard format

JOUT OPERATIONAL DESCRIPTION

The JOUT question/answer sequence is as follows:

QUESTION: SYSTEM?

Response: JOUT or JOUT sssss

Where: sssss - SNUMB for the job whose output is to be manipulated. If only JOUT is entered, the system requests SNUMB?

A null response causes a return to SYSTEM?

If the job output is not available for manipulation, the system transmits one of the following messages to the terminal and returns to the SYSTEM? level (sssss is the job SNUMB):

sssss OUTPUT IS BUSY

The job was found but is being printed at the central site, at a remote batch station, or at another time-sharing terminal.

sssss OUTPUT NOT FOUND

The desired job was not found in the system.

sssss NOT YOUR JOB

The log-on USERID does not match the USERID found in the job stream.

BATCH SYSTEM FULL--TRY LATER

Time-sharing cannot communicate with SYSOUT at this time, or not enough room exists to read the file.

sssss READING-CR

The job is being input through the central site card reader.

sssss READING-MT

The job is being read from magnetic tape.

sssss READING-RMT

The job is being read from a remote station.

sssss WAITING-ALLOC

The job is not yet a candidate for peripheral allocation.

sssss WAIT-PERIF

The job is waiting for peripheral allocation.

sssss WAIT-CORE

The job is waiting for core allocation.

sssss IN HOLD

The central site operator has placed a hold on this job.

sssss IN LIMBO
 The job is waiting for a RUN command from the central site operator.

sssss EXECUTING
 The job is in execution.

sssss WAIT-TAPE
 The job is waiting for the operator to mount or ready a specified tape.

sssss TOO BIG
 This job's requirements exceed the sieve limits set by the central site.

sssss OVERDUE
 The allocation of this job is overdue.

sssss IN RESTART
 The job is waiting to be restarted by the operator after a system failure.

sssss TERMINATING
 This job is engaged in the termination procedure.

Question: *SNUMB?*

QUESTION: FUNCTION?

Response: One of the following commands. When the function has been completed, the system returns to FUNCTION?

- ACTIVITY

ACTIVITY n *n = \$ \$*

JOUT prepares to read the activity specified by n.

- DIRECT

a. DIRECT id

Direct the output to the remote station specified by id.

b. DIRECT ONL

Print the output at the central site.

EPRINT rc

Simulate printer report output. The report code rc may be any of the codes received from the LIST command, or \$\$ may be substituted for a report code. The \$\$ causes the printing of the J* file (control card list and execution report) at the terminal. Trailing blanks and blank lines are suppressed.

- LIST

LIST

List the report codes associated with the current activity.

- PRINT

PRINT rc

Simulate printer report output. The report code rc may be any of the codes received from the LIST command, or \$\$ may be substituted for a report code. \$\$ causes the printing of the J* file (control card list and execution report) at the terminal. Multiple blanks are suppressed by the PRINT command.

- REMOVE

REMOVE

Remove the output from the system.

- SCAN

SCAN rc

Scan the job output with report code rc. The system requests FORM? From this point, the question/answer sequence and the facilities available are the same as for the SCAN subsystem (see Section III), with the exceptions noted below:

1. The following SCAN verbs are not available: BATCH, REM, REM text, and BYE.
2. Output in core dump format may be scanned. (Answer DUMP to the FORM? question. There is no initial subsystem response to this answer; the EDIT? question appears immediately.)
3. DONE returns the user to the FUNCTION level.

APPENDIX A

INTERFACE USAGE EXAMPLES

The following sample program illustrates the use of interrelated time-sharing subsystems and batch programming features. The program is submitted by means of the time-sharing CARDIN subsystem. Direct conversation between the program and the user's terminal is then initiated, and use is made of two conversational batch dimension features--Conversational Debug Routine (RBUG) and conversational I/O extensions to File and Record Control. Text within brackets is not part of the program but has been added to illustrate particular features.

[The program, submitted under CARDIN, makes use of conversational I/O extensions to File and Record Control. The STAB option on the \$ FORTRAN card is not a requirement initially but is required for use with the subsequent RBUG portion of the program.]

```

0100$;IDENT;VXEEO,JDOE
0200$;OPTION;FORTRAN
201$;USE;.RTYP ← Required when $ DAC cards are
0300$;FORTRAN;NDECK,STAB present.
400#2;WRITE(6,3)
0500#3;FORMAT(27HPROGRAM TO CALCULATE RECOIL)
0600#1;WRITE(6,4)
0700#4;FORMAT(9HRIFLE WT.)
0800;READ(5,5)WR
0900#5;FORMAT(F6.2)
1000;WRITE(6,7)
1100#7;FORMAT(10HBULLET WT.)
1200;READ(5,5)WB
1300#9;WRITE(6,10)
1400#10;FORMAT(8HVELOCITY)
1500#11;READ(5,5)VB
1600;WRITE(6,13)
1700#13;FORMAT(10HPOWDER WT.)
1800;READ(5,5)WP
1900;X=WB*VB+4700.*WP
2000;Y=7000.*WR
2100;Z=WR/64.4
2200;E=Z*(X/Y)**2
2300#15;WRITE(6,16)E
2400#16;FORMAT(8HENERGY= F6.2,9H FT. LBS.)
2500;GO TO 1
2600;END
2700$;EXECUTE
2800$;DAC;05
2900$;DAC;06
3000$;ENDJOB

```

READY

[The program is then formatted for legibility.]

```
*PRINT
CARD FORMAT ?
NORM(;
```

```
09/10/69    09.69
```

```
0100  $    IDENT  VXE00,JDOE
0200  $    OPTION  FORTRAN
201   $    USE     .RTYP
0300  $    FORTRAN NDECK,STAB
400   2    WRITE (6,3)
0500  3    FORMAT (27HPROGRAM TO CALCULATE RECOIL)
0600  1    WRITE (6,4)
0700  4    FORMAT (9HRIFLE WT.)
0800          READ (5,5)WR
0900  5    FORMAT (F6.2)
1000          WRITE (6,7)
1100  7    FORMAT (10HBULLET WT.)
1200          READ (5,5)WB
1300  9    WRITE (6,10)
1400  10   FORMAT (8HVELOCITY)
1500  11   READ (5,5)VB
1600          WRITE (6,13)
1700  13   FORMAT (10HPOWDER WT.)
1800          READ (5,5)WP
1900          X=WB*VB+4700.*WP
2000          Y=7000.*WR
2100          Z=WR/64.4
2200          E=Z*(X/Y)**2
2300  15   WRITE (6,16)E
2400  16   FORMAT (8HENERGY= F6.2,9H FT. LBS.)
2500          GO TO 1
2600          END
2700  $    EXECUTE
2800  $    DAC     05
2900  $    DAC     06
3000  $    ENDJOB
```

[The program is then passed to the batch system for processing; the TALK option permits direct-access connection.]

```
*RUN
  SNUMB # 0165T
CARD FORMAT, DISPOSITION ?
NORM(;),TALK
PROGRAM TO CALCULATE RECOIL
RIFLE WT.
05?8.5
BULLET WT.
05?150
VELOCITY
05?3200
POWDER WT.
05?58
ENERGY=320.06 FT. LBS.
RIFLE WT.
05? ← Carriage return; null response.
*****ERROR
  TRACE OF CALLS IN REVERSE ORDER
  CALLING      ID  ABSOLUTE  ARGUMENT      ARGUMENT
  ARGUMENT    ARGUMENT  ARGUMENT
  ROUTINE     #    LOCATION   #1            #2
  #3          #4          #5
  .FEOF.      19   035114   000000000042
                                     ERROR
  .FRDD.      954  036771
                                     ERROR
  .....      5    037556   000000000005   352606330255
                                     ERROR
  END OF FILE READING      FILE CODE 05 OPTIO
  OPTIONAL RETURN NOT REQUESTED013068
*****ERROR
CLOSING FILE 05
CLOSING FILE 06

ACTIVITY TERMINATED
NORMAL TERMINATION
```


[A \$ USE RBUG card is substituted for \$ USE .RTYP to initiate the RBUG subroutine, and breakpoints are inserted. The program is then formatted in its new version.]

```
*201$;USE;RBUG
*202$;DUMP;.....
*203;DEBUG;2/(BREAKPOINT)
*204;DEBUG;15/(BREAKPOINT)
*PRINT
CARD FORMAT ?
NORM(;
```

09/10/69 09.85

```
0100 $ IDENT VXE00,JDOE
0200 $ OPTION FORTRAN
201 $ USE RBUG
202 $ DUMP .....
203 DEBUG 2/(BREAKPOINT)
204 DEBUG 15/(BREAKPOINT)
0300 $ FORTRAN NDECK,STAB
400 2 WRITE(6,3)
0500 3 FORMAT(27HPROGRAM TO CALCULATE RECOIL)
0600 1 WRITE(6,4)
0700 4 FORMAT(9HRIFLE WT.)
0800 READ(5,5)WR
0900 5 FORMAT(F6.2)
1000 WRITE(6,7)
1100 7 FORMAT(10HBULLET WT.)
1200 READ(5,5)WB
1300 9 WRITE(6,10)
1400 10 FORMAT(8HVELOCITY)
1500 11 READ(5,5)VB
1600 WRITE(6,13)
1700 13 FORMAT(10HPOWDER WT.)
1800 READ(5,5)WP
1900 X=WB*VB+4700.*WP
2000 Y=7000.*WR
2100 Z=WR/64.4
2200 E=Z*(X/Y)**2
2300 15 WRITE(6,16)E
2400 16 FORMAT(8HENERGY= F6.2,9H FT. LBS.)
2500 GO TO 1
2600 END
2700 $ EXECUTE
2800 $ DAC 05
2900 $ DAC 06
3000 $ ENDJOB
```

[The program is again passed to the batch system, along with the TALK option. Control of the program is obtained at breakpoints, interrogations are made, and the program is then permitted to continue and run to termination.]

```

*RUN
  SNUMB #0166T
CARD FORMAT, DISPOSITION ?
NORM(;),TALK
***ROUTINE ..... LOC 2          COUNT    000001
???R
PROGRAM TO CALCULATE RECOIL
RIFLE WT.
05?8.5
BULLET WT.
05?150
VELOCITY
05?3200
POWDER WT.
05?58
***ROUTINE ..... LOC 15        COUNT    000001
???FWR CHECK ANSWER
WR      0.85000000E 01
???FE  I CHECKED THE WRONG THING
E       0.38320059E 05
???FWB CHECK BULLET WEIGHT
WB      0.15000000E 04
???R#2 BULLET WEIGHT FUNNY, TRY AGAIN WITH DECIMAL POINTS
***ROUTINE ..... LOC 2          COUNT    000002
???R BEGIN NORMAL RUN
PROGRAM TO CALCULATE RECOIL
RIFLE WT.
05?8.5 8 AND 1/2 POUNDS
BULLET WT.
05?150.0 150 GRAINS
VELOCITY
05?3200.0 FEET PER SECOND
POWDER WT.
05?58.0 GRAINS
***ROUTINE ..... LOC 15        COUNT    000002
???FE GET A PEEK AT ANSWER
E
???R LOOKS GOOD
ENERGY= 21.11 FT. LBS.
RIFLE WT.
05?7.5
BULLET WT.
05?150.
VELOCITY
05?2175
POWDER WT.
05?31.0 I MADE A MISTAKE. NO DECIMAL ON VELOCITY
***ROUTINE ..... LOC 15        COUNT    000003
???R#2 TRY AGAIN. THIS TIME WITH DECIMAL POINT.
***ROUTINE ..... LOC 2          COUNT    000003
???R
PROGRAM TO CALCULATE RECOIL

```

```

RIFLE WT.
05?7.5
BULLET WT.
05?150.
VELOCITY
05?2175
POWDER WT.
05?0. AGAIN FORGOT THE DECIMAL POINT
***ROUTINE ..... LOC 15          COUNT    000004
???R#2 TRY AGAIN
***ROUTINE ..... LOC 2          COUNT    000004
???R TRY A "NORMAL" RUN
PROGRAM TO CALCULATE RECOIL
RIFLE WT.
05?7.5
BULLET WT.
05?150.0
VELOCITY
05?2175.0
POWDER WT.
05?31.0
***ROUTINE ..... LOC 15          COUNT    000005
???FE PEEK AT ANSWER
E 0.94112817E 01
???R ANSWER SEEMS ABOUT RIGHT FOR 30/30 WITH LIGHT LOAD
ENERGY= 9.41 FT. LBS.
RIFLE WT.
05?7.5 TRY 30.06 TYPICAL LOAD
BULLET WT.
05?180.0
VELOCITY
05?2505.0
POWDER WT.
05?45.0
***ROUTINE ..... LOC 15          COUNT    000006
???R LET IT GO NORMALLY
ENERGY= 18.54 FT. LBS.
RIFLE WT.
05?0.
BULLET WT.
05?0.
POWDER WT.
05?0.
      DIV CHECK   AT LOCATION 037651
      EXP OVERFLO AT LOCATION 037400
***ROUTINE ..... LOC 15          COUNT    000007
???T LET THE PROGRAM QUIT NORMALLY
**EXIT
CLOSING FILE 05
CLOSING FILE 06

ACTIVITY TERMINATED
NORMAL TERMINATION

*BYE

```

INDEX

\$ DAC	
\$ DAC	6-4
\$ DAC	6-1
\$ IDENT	
\$ IDENT card	2-6
\$ IDENT?	
\$ IDENT?	2-15
\$ IDENT?	3-5
\$ SELECTA	
\$ SELECTA Card Option	2-16
\$ USE	
\$ USE .RTYP	6-1
\$ USE .RTYP	A-4
\$ USE RBUG	A-4
\$ USE RBUG	5-1
\$ USERID	
\$ USERID control card	7-1
.RTYP	
\$ USE .RTYP	6-1
\$ USE .RTYP	A-4
ABCDEIJ	
abcde(i,j)	2-13
ACTIVITY	
ACTIVITY	7-3
ASCBCD	
ASCBCD	2-5
ASCBCD Question/Answer Sequence	2-12
ASCII-TO-BCD	
ASCII-to-BCD conversion	2-2
ASIS	
ASIS	2-12
ASIS	2-7
AUTO	
AUTO	2-14
BACK	
BACK	3-5
BATCH	
BATCH	3-5

BCDASC		
BCDASC		2-6
BCDASC Question/Answer Sequence		2-14
BLOCK TO BE READ		
BLOCK TO BE READ		4-2
BPRINT		
BPUNCH and BPRINT		2-6
BPUNCH or BPRINT Question/Answer Sequence		2-15
BPUNCH		
BPUNCH and BPRINT		2-6
BPUNCH or BPRINT Question/Answer Sequence		2-15
BREAKPOINT		
breakpoint		5-1
BYE		
BYE		3-7
CARD		
\$ IDENT card		2-6
\$ SELECTA Card Option		2-16
\$ USERID control card		7-1
CARD FORMAT		
CARD FORMAT, DISPOSITION?		2-7
CARDIN		
CARDIN Command Language		2-3
CARDIN DETAILED OPERATION		2-3
CARDIN FUNCTION		2-1
CARDIN SUBSYSTEM		2-1
TALK option of CARDIN		2-1
CARDIN OPERATION		
CARDIN OPERATION		2-1
CHARACTER		
standard character		2-12
multiple tab characters		2-12
Tab characters		2-2
tab characters		2-12
CLOSE		
CLOSE		6-3
CODE		
CODE		3-6
Line codes		3-2
COMMAND		
CARDIN Command Language		2-3
Special command verbs		3-1
CONSIDERATIONS		
FILE AND RECORD CONTROL I/O CONSIDERATIONS		6-3

CONTROL		
\$ USERID control card		7-1
FILE AND RECORD CONTROL I/O CONSIDERATIONS		6-3
CONTROL CARDS		
CONTROL CARDS FOR FILE AND RECORD CONTROL I/O		6-1
CONVERSATIONAL		
CONVERSATIONAL DEBUG		5-1
CONVERSATIONAL FILE AND RECORD INPUT/OUTPUT		6-1
CONVERSION		
ASCII-to-BCD conversion		2-2
COPY		
copy file		4-4
COPY FUNCTION		
FDUMP Copy Function Details		4-4
DEBUG		
CONVERSATIONAL DEBUG		5-1
DESCRIPTION		
FDUMP OPERATIONAL DESCRIPTION		4-1
JOUT OPERATIONAL DESCRIPTION		7-2
RBUG GENERAL DESCRIPTION		5-1
SCAN OPERATION DESCRIPTION		3-2
DIRECT		
DIRECT		7-3
DISPOSITION?		
CARD FORMAT, DISPOSITION?		2-7
DISPOSITION?		2-9
DONE		
DONE		3-7
DUMP		
file dump		4-1
EDIT		
EDIT		3-7
EDIT?		
EDIT?		3-3
EPRINT		
EPRINT		7-4
ERROR		
ERROR		3-6

FDUMP		
FDUMP		2-6
FDUMP Copy Function Details		4-4
FDUMP Error Indication and Replies		4-3
FDUMP FUNCTION		4-1
FDUMP OPERATIONAL DESCRIPTION		4-1
FDUMP SUBSYSTEM		4-1
FILE		
CONVERSATIONAL FILE AND RECORD INPUT/OUTPUT		6-1
copy file		4-4
FILE AND RECORD CONTROL I/O CONSIDERATIONS		6-3
FILE NAME; TYPE		4-1
file dump		4-1
FILE AND RECORD CONTROL		
File and Record Control routines		6-1
FILE AND RECORD CONTROL I/O		
CONTROL CARDS FOR FILE AND RECORD CONTROL I/O		6-1
FILE-BUILDING		
file-building		2-1
FILES		
permanent files		4-1
FIND		
FIND		3-4
FIRST		
First Line Reformatting Information		2-15
FLAG		
FLAG		3-6
FORM?		
FORM?		3-2
FUNCTION?		
FUNCTION?		4-2
GET		
GET		6-2
GET		6-3
I/O		
FILE AND RECORD CONTROL I/O CONSIDERATIONS		6-3
I/O CALLS		6-2
INPUT/OUTPUT		
CONVERSATIONAL FILE AND RECORD INPUT/OUTPUT		6-1
INSTRUCTIONS		
RBUG INSTRUCTIONS		5-2
INTERFACE		
INTERFACE USAGE EXAMPLES		A-1

JABT		
JABT		2-5
JDAC		
JDAC		2-4
JOUT		
JOUT		2-7
JOUT		2-10
JOUT FUNCTION		7-1
JOUT OPERATIONAL DESCRIPTION		7-2
JOUT SUBSYSTEM		7-1
JSTS		
JSTS		2-5
LABELS?		
LABELS?		2-12
LABELS?		2-15
LANGUAGE		
CARDIN Command Language		2-3
LINE		
First Line Reformatting Information		2-15
LINE		3-5
Line codes		3-2
line numbers		2-7
LINE NUMBERS?		2-14
LIST		
LIST		3-4
LIST		7-4
LITERAL		
literal string		3-4
LOAD		
LOAD MAP		3-6
MOVE		
MOVE		2-14
MOVE		2-12
MOVE		2-7
NAME; TYPE		
FILE NAME; TYPE		4-1
NORM		
NORM		2-8
NORM		2-12
OPEN		
OPEN		6-2

OPERATIONAL		
FDUMP OPERATIONAL DESCRIPTION		4-1
JOUT OPERATIONAL DESCRIPTION		7-2
PERMANENT		
permanent files		4-1
PRINT		
PRINT		3-4
PRINT		7-4
PRINT		2-4
RUN or PRINT Question/Answer Sequence		2-7
PUT		
PUT		6-3
QUESTION/ANSWER		
ASCBCD Question/Answer Sequence		2-12
BCDASC Question/Answer Sequence		2-14
BPUNCH or BPRINT Question/Answer Sequence		2-15
RUN or PRINT Question/Answer Sequence		2-7
RBUG		
\$ USE RBUG		A-4
\$ USE RBUG		5-1
RBUG		5-1
RBUG		A-1
RBUG GENERAL DESCRIPTION		5-1
RBUG INSTRUCTIONS		5-2
RECORD		
CONVERSATIONAL FILE AND RECORD INPUT/OUTPUT		6-1
FILE AND RECORD CONTROL I/O CONSIDERATIONS		6-3
REFORMATTING INFORMATION		
First Line Reformatting Information		2-15
REM		
REM		3-7
REMOVE		
REMOVE		7-4
REPLIES		
FDUMP Error Indication and Replies		4-3
ROUTINES		
File and Record Control routines		6-1
RUN		
RUN		2-4
RUN or PRINT Question/Answer Sequence		2-7

SCAN		
SCAN		2-6
SCAN		7-4
SCAN FUNCTION		3-1
SCAN OPERATION DESCRIPTION		3-2
SCAN SUBSYSTEM		3-1
SCAN VERBS		3-3
SETTINGS		
tab settings		2-12
tab settings		2-2
SNAPSHOTS		
snapshots		5-1
SPACE		
SPACE		3-5
STAB		
STAB option		A-1
STANDARD		
standard character		2-12
STATION CODE?		
STATION CODE?		3-5
STRING		
literal string		3-4
STRIP		
STRIP		2-12
STRIP		2-7
SUBSYSTEM		
CARDIN SUBSYSTEM		2-1
FDUMP SUBSYSTEM		4-1
JOUT SUBSYSTEM		7-1
SCAN SUBSYSTEM		3-1
TAB		
multiple tab characters		2-12
Tab characters		2-2
tab characters		2-12
tab settings		2-12
tab settings		2-2
TAB CHARACTERS AND SETTINGS?		
TAB CHARACTERS AND SETTINGS?		2-11
TAB CHARACTERS AND SETTINGS?		2-15
TAB CHARACTERS AND SETTINGS?		2-13
TALK		
TALK		2-10
TALK		A-3
TALK		A-5
TALK option of CARDIN		2-1

UNDEFINED	
UNDEFINED	3-6
URGCXX	
URGC(xx)	2-10
VERBS	
SCAN VERBS	3-3
Special command verbs	3-1
WAIT	
WAIT	2-9

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form*

TITLE: SERIES 600/6000 GCOS TIME-SHARING
SYSTEM TERMINAL/BATCH INTERFACE
FACILITY

ORDER No.: BR99, REV. 1

DATED: JANUARY 1972

ERRORS IN PUBLICATION:

[Empty box for reporting errors in publication]

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION:

[Empty box for providing suggestions for improvement to publication]

(Please Print)

FROM: NAME _____
COMPANY _____
TITLE _____

DATE: _____

COPYING LINE

*Your comments will be promptly investigated by appropriate technical personnel, action will be taken as required, and you will receive a written reply. If you do not require a written reply, please check here.

CUT ALONG

FOLD ALONG LINE

FOLD ALONG LINE

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS,
MASS. 02181

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

POSTAGE WILL BE PAID BY:

HONEYWELL INFORMATION SYSTEMS
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATTN: PUBLICATIONS, MS 050

Honeywell

HONEYWELL INFORMATION SYSTEMS

Publications Remarks Form*

TITLE: SERIES 600/6000 GCOS TIME-SHARING SYSTEM
TERMINAL/BATCH INTERFACE FACILITY
ADDENDUM B

ORDER No.: BR99B, REV. 1

DATED: JANUARY 1973

ERRORS IN PUBLICATION:

[Empty box for reporting errors in the publication]

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION:

[Empty box for providing suggestions for improvement to the publication]

(Please Print)

FROM: NAME _____
COMPANY _____
TITLE _____

DATE: _____

C. LONG LINE

*Your comments will be promptly investigated by appropriate technical personnel, action will be taken as required, and you will receive a written reply. If you do not require a written reply, please check here.

CUT ALONG LINE

FOLD ALONG LINE

FOLD ALONG LINE

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS,
MASS. 02181

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

POSTAGE WILL BE PAID BY:

HONEYWELL INFORMATION SYSTEMS
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATTN: PUBLICATIONS, MS 050

Honeywell

The Other Computer Company:
Honeywell

HONEYWELL INFORMATION SYSTEMS

7492
2573
Printed in U.S.A.

In the U.S.A.: 200 Smith Street, MS 061, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario

BR99, Rev. 1