MEMORANDUM LISP Memo 2

TO: SDS 940 LISP Users
FROM: Warren Teitelman
SUBJECT: Recent Improvements to 940 LISP Library
DATE: 10 April 1967

---

time[x;n;g] Time executes the computation x, n number of times, and prints out the number of conses and computation time per iteration. Garbage collection time is not included, i.e. it is subtracted out. If n is NIL, it is set to 1. If g is T, garbage collection time is also printed.

Example:

```
TIME ((CONS NIL NIL) 1000 T)
GARBAGE COLLECTION
2458 CELLS
1 CONSES
0.01 SECONDS
GARBAGE COLLECTION TIME: 23 SECONDS
(NIL)

TIME ((CONS NIL NIL))
1 CONS
0.0 SECONDS
(NIL)
```

prettyfile[l;f]	Prettyfile performs a prettydef on the list of functions l, writing them out onto drum file /f/, complete with STOP.
brkfn[??]	BRKFN is a convenient function to use as a break function when this facility is working. BRKFN calls BREAK1 with ?? as the form in question. Thus if you do BREAKFN[BRKFN], the next time an error occurs, you will instead go to BREAK1 which will print (COMPUTATION BROKEN) and then you can interrogate ??, and either go on, or quit, etc.
break[l]	BREAK is analagous to the old BREAKLIST which no longer exists. It takes a list of functions and sets up a BREAK as before, except it also allows specifying conditions other than T. Thus, BREAK(FOO FOO1 (FOO2 (GREATERP X 5) Y)) is the same as BREAKLIST(FOO FOO1) plus BREAK(FOO2 (GREATERP X5) Y) in the old system.
breakØ[fn;when;what]	This is the old BREAK.

break1[zbrklexp;brklwhen;brklfn;brklwhat]
: This is relatively unchanged except that it has been made more error proof, especially when it is used by trace. BREAK1 now recognizes the command ↑, and calls RESET which takes you back to evalquote.

unbreak[l]
: Same as the old UNBREAKLIST, which no longer exists

breakonce[fn]
: BREAKONCE is a new breaking function. It is especially useful with recursive functions. BREAKONCE establishes a break on fn in the normal way but this break occurs only on the first time that the function is entered. For example, you can now do BREAKONCE(MAPLIST) and only one break will occur for each call to MAPLIST, regardless of how long the list is.

trace[l]

TRACE now works in conjunction with BREAKl. It takes a list of functions similar to BREAK, and redefines them using a call to BREAKl so that BREAKl will print the value of the arguments and the value of the function without actually breaking. Recent improvements to TRACE are:

(1) The user can specify the values of interest to him in addition to or instead of the arguments of the function, by writing a list headed by the function followed by the values of interest, in place of just the function name.

Example:

```
TRACE(FOO (FOOl Y (CAR Z)))
(FOO FOOl)

FOO(A B (C D))
FOO:
X = A          ... arguments of FOO
Y = B
Z = (C D)

FOOl:
Y = A
(CAR Z) = NIL

etc.
```

(2) The user can specify the level to which the arguments, or values, are to be printed by writing (FN N X Y Z ...) in the call to TRACE. N is taken to be 4 if not specified by this device.

(3) If an error occurs, or RUBOUT is pressed, while a function is being traced, a normal BREAK occurs and, the user can proceed from that point.

Example:

```
TRACE(FACTORIAL)
(FACTORIAL)

FACTORIAL(2)
FACTORIAL:
N = 2

FACTORIAL:
N = 1

FACTORIAL:
RUBOUT      ... RUBOUT pressed here

(FACTORIAL BROKEN)  ... BREAK occurs
N
0
EVAL
```

```
                        FACTORIAL EVALUATED
                        FACTORIAL
                        1
                        OK
                        FACTORIAL   ... exit from BREAK

                        FACTORIAL = 1

                        FACTORIAL = 2
                        2
```

untrace[x] No longer exists: use unbreak

All of the break and tracing functions may be done any number of times without harm, i.e. it is unnecessary to unbreak before breaking with different conditions, tracing or using breakonce. Similarly, all of these functions add the functions broken or traced to a list which is the value of the atom ALL. By doing UNBREAK(ALL) at any time, all functions that have been broken or traced since the last time ALL was set to NIL will be restored to their original form. ALL will then be set to NIL.