TO: M. Bernstein

FROM: J. Burger

SUBJECT: NEW I/O FOR LISP

COPIES TO: E. Book, T. Diller, K. Fogt, C. Kellogg, R. Long
             A. Leal, V. Schorre, C. Weissman, T. Williams

All disc copies of SDC's LISP-360 system dated FEB 24, 1972 and
higher will feature a new modification of the I/O sub-system
causing LISP to read and write Form 7 (F7) files on both disc
and tape. This, for the first time, makes LISP compatabile with
other ADEPT library routines, notably OLE, DEBE and UTIL.

User conventions will permit reading of any F7 file provided that
total record size does not exceed 4092 bytes and provided that
sequence numbers of the cards (strings) are in the last eight
columns (bytes) of the string. LISP will write F7 files in three
formats as tabulated below:

FORMAT	STRING SIZE	ORIGINAL PACKING FACTOR	FULL PACKING FACTOR	USABLE BYTES/STRING	USABLE BYTES/RECORD
CARD	80	30	50	72	2160
PRINTER	132	30	30	124	3720
PACKED	255.	16	16	247	3952

Selection of the desired format is made simply by inserting the
format name in the LISP File Descriptor List. The default for-
mat is CARD.

No format name is necessary when opening existing files since this
is then determined by the Packed-String File Header (PSFH) in the
file.

If an existing file is to be overwritten it is required only to open
it. If it is desired that additional data is to be added to the end
of an existing file, it is then also necessary to call CONTINUE
(file:name) which will position the file to the end and set the
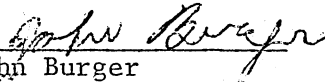necessary variables such as the sequence number counter.

When opening existing files LISP expects to find files of form F7.
If such a file is not found under the given name and volume, a

second search is made for the file as form F4. Thus both F4 and
F7 files may be read, and if both forms exist with the same name
and volume, the F7 file will be opened.

Another new feature permits use of the word POL to designate the
unit type and volume in the file descriptor list. This may be
used for both new (N) and existing (E) files.

Paperwork estimates indicate that speed increases of up to 16:1
are possible in our new use of SPAM. Real numbers are difficult
to get since SPAM time is not indicated in total CPU time, but
real time improvements of almost 2:1 have already been recorded.

John Burger

JB:sp