COMPUTER SYSTEMS LABORATORY
WASHINGTON UNIVERSITY
ST. LOUIS, MO. 63110

LINC Document No. 45 March 28, 1969

PRINT-7

M. J. Stucki

Abstract

PRINT-7 is a general purpose subroutine providing a flexible teletype printout facility for both the classic LINC and the SPEAR LINC. TTCODE-7 is the table of teletype codes used by PRINT-7.

The Computer Systems Laboratory is supported in part by the Advanced Research Projects Agency of the Department of Defense under contract SD-302, and in part by the Division of Research Facilities and Resources of the National Institutes of Health under grant FR-00396.

PRINT-7

PRINT-7 is a general purpose subroutine providing a flexible printing facility for the teletype on the classic LINC and on the SPEAR LINC. The routine occupies 370 octal locations in memory, and works in conjunction with TTCODE-7, a table of teletype codes that may be located in upper or lower memory. Tags 9A through 9Z are used by the two manuscripts. Printing is requested by a three word instruction of the following form:

```
JMP 9P
Address of Format Statement
Address of Variables List
```

The contents of the accumulator and the contents of beta-registers are not altered by a print request. The format statement and the variables list may be located in either upper or lower memory; they may also straddle the two memories. They may not, however, occupy locations 0,1, or 2.

FORMAT STATEMENT

A format statement consists of a string of print commands terminated by a period. Spaces and commas may be used to separate commands within a statement, but are not required. Spaces may not be used internal to a print command (this does not apply to the printing of character strings, since a space is a legal character). The following notation is used in defining the print commands:

1. "n" denotes a decimal number such that $0 \leq n \leq 1024$.
2. "x" denotes any LAP6 character except upper and lower quotes.
3. "v" denotes the next item in the variables list.
4. "a" denotes the contents of the accumulator when printing was requested.
5. "c(z)" denotes the contents of location z. A useful extension of this is

$$\begin{aligned} c^0(z) &= z \\ c^1(z) &= c(z) \\ c^2(z) &= c(c(z)) \\ c^3(z) &= c(c(c(z))) \\ &\text{etc.} \end{aligned}$$

PRINT COMMANDS

Ln - carriage return and n line feeds.
L - same as L1.

Sn - print n spaces.
S - same as S1.

M - define new mode. Address of new mode statement is v.

Vn - print datum $c^n(v)$ under mode control.
V - same as V1.

An - print datum $c^n(a)$ under mode control.
A - same as A1.

"xx···x" - print everything between the upper quotes.

MODE STATEMENT

A mode statement consists of a string of editing commands terminated by an upper quote. Spaces and commas may not be used to separate commands inasmuch as they are themselves legal editing commands. Mode statements may be located in either upper or lower memory; they may also straddle the two memories. They may not, however, occupy locations 0,1, or 2. The following notation is used in defining the editing commands:

1. "d" denotes a decimal digit such that $1 \leq d \leq 9$.
2. "y" denotes any LAP6 character except digits, upper quotes and lower quotes.

EDITING COMMANDS

The datum word to be printed is scanned from left to right under the control of the following set of editing commands:

0 - ignore the next bit of the datum word.

d - print the character defined by the next d bits of the datum word.

y - print character y.

It should be noted that the scanning is done by rotating the datum word, and hence bit 11 follows bit zero if scanning is allowed to proceed past bit zero.

EXAMPLES OF MODE STATEMENTS

Assume that the datum word is 010100000111.

Mode Statement	Resulting Output
3333"	2407
33"	24
111111111111"	010100000111
111,111,111,111"	010,100,000,111
000000.33"	.07
000000 .33"	.07
.023E-33"	.24E-07
66"	A7
03"	5
00030003"	47

PROGRAMMING EXAMPLES

It is assummed that the following examples are to be assembled by LAP6. Because of this, the mode and format statements are shown as LAP6 text mode statements. This is an absolute necessity --- mode and format statements must appear in the converted program as character strings.

Example 1: The following piece of program will print the contents of locations 6A,6B,6C, and 6D. The values will be printed in octal, on the same line, with three intervening spaces.

```
     Main Program
       -----
       -----                    #1A  "L,M,V,S3,V,S3,V,S3,V.""      [format statement
       -----
       -----                    #1B  1C                           [variables list
       JMP 9P                        6A
        1A                           6B
        1B                           6C
       -----                         6D
       -----
       -----                    #1C  "3333""                      [mode statement
       -----
```

The same thing could be accomplished by changing the format and mode statements to

```
                                #1A  "L,M,V,V,V,V.""              [format statement

                                #1C  "3333   ""                   [mode statement
```

Example 2: The following program gives a memory dump of locations zero through 777. The address column will be labeled "ADDRESS" and the contents column will be labeled "CONTENTS". There will be five intervening spaces between the columns, addresses will be in octal, and contents will be in binary.

```
       JMP 9P
        1A                      #1A  "L,"ADDRESS",S2,"CONTENTS"."" [format statement
        0
       CLR
    ┌→ JMP 9P
    │   1B                      #1B  "L,M,A0,S5,M,A.""            [format statement
    │   1C
    │  ADAi                     #1C  1D                           [variables list
    │   1                            1E
    │  SAEi
    │   1000                    #1D  "3333""                      [mode statement
    └─ JMP p-7                  #1E  "111,111,111,111""           [mode statement
       HLT
```

GENERAL NOTES

1. If the LAP6 character EOL is encountered when printing a character string or when scanning a mode statement, a question mark will be printed and the program will move on to the next character. There will be no carriage returns or line feeds.

2. Carriage returns and line feeds are the sole responsibility of the programmer. The program will not automatically execute a carriage return if an attempt is made to print to the right of the rightmost print position.

3. If a syntax error in a format statement is encountered, the program will halt.

4. The contents of the Link bit, the Z register, and the overflow bit are destroyed by PRINT-7. The contents of the R register is untouched on the SPEAR LINC; on the classic LINC R_0 is destroyed and the rest of R is untouched.