B0057

Date: May 10, 1966

From: V. A. Vyssotsky

Subject: Character Conversion for PRT-202 Line Printer

To: Multics Repository

The PRT-202 line printer is designed to print the proposed revised ASCII standard character set. It could have been designed to accept data for printing in the form of 9 bit characters, with the 94 ASCII graphics and the space represented as 95 different 9 bit patterns; we shall call this alternative A. However, in order to maintain compatibility with existing hardware and software, the printer has been designed to accept its input as 6 bit characters. The full ASCII character set is obtained by means of case shifts. At the beginning of each line the printer is in "upper case". A particular character (octal 77) is designated as an escape, and the occurrence of two consecutive escape characters is interpreted as a case reversal. We shall call this actual printer design alternative B. The reader should consult the printer specification, EPS M50EB00070, for details.

During the past six months various individuals have raised the question of how much extra CPU time is required to take text in core storage in a form suitable for printing with alternative A, and reformat it for printing under alternative

B. Oral estimates of this time have ranged from "negligible" to "eighty percent of the time of a CPU to keep four printers busy". No written estimates appear to be available. It seems appropriate, therefore, to have such an estimate in writing, with the assumptions behind the estimate as clearly defined as possible.

Figure 1 displays the main loop of a subroutine with four arguments. The subroutine is called by

```
call convert(input,n,output,n)
```

At time of call, input is a string of n 9 bit characters, starting on a word boundary. At time of return, output is a string of n 6 bit characters, starting on a word boundary. The output string is the input string, transformed for printing on a PRT-202. Characters in the input string other than ASCII graphics and space are converted into something or other. Figure 1 does not show the initialization and exit code of the subroutine; these comprise about fifty instructions, executed once per call.

The conversion loop of this routine consumes about 16 microseconds for each character, plus about 15 microseconds for each case shift. The initialization and exit routines consume about 100 microseconds for each line. Let us translate this into CPU load in two extreme situations. First, suppose every

line can be printed in upper case, and furthermore all characters are among those repeated on the printer drum. Suppose the average line length is 100 characters. Conversion of one line requires 1.7 milliseconds. Since the PRT-202 will print 20 such lines per second, the conversion routine requires 34 milliseconds per second, or 3.4% of the time of one CPU, to keep one printer busy. On the other hand, suppose every line to be printed requires both upper and lower case. Suppose the average line length is 100 characters, and suppose 40 case shifts are required for the average line. Conversion of one line requires 2.3 milliseconds. Since the PRT-202 will print 10 such lines per second, the conversion routine requires 23 milliseconds per second, or 2.3% of the time of one CPU, to keep one printer busy.

A routine employed in actual operation of the line printer would have to do more than the routine outlined above. It would have to insert slew characters, transform horizontal tabs into skips, cope with such control characters as red shift and bell, and check for excessive line length. In addition, CPU time is required for getting the material to be printed from secondary store, for making up GIOC DCWs, issuing connects and examining status returns. A rough guess at the time required to perform 9 bit to 6 bit conversions plus all of these other functions is that it will average out to about

twice as much as the conversion indicated above. If this guess is approximately correct, then about five or six percent of the time of one CPU is required to keep one PRT-202 busy. In particular, even if 9 bit to 6 bit conversion were not required, some such scan loop as is shown in figure 1 would have to be executed in order to edit the ASCII control characters. Hence, it is not clear how much CPU time may correctly be attributed to conversion from 9 bit to 6 bit characters. If this conversion is performed by a separate scan which serves no other purpose, then 9 bit to 6 bit conversion for one busy printer will occupy two to three percent of the time of one CPU. If, as seems likely, the 9 bit to 6 bit conversion can be done during the scan to edit control characters, the load attributable to 9 bit to 6 bit conversion for one printer would be well under one percent of one CPU.

Figure 1

```
            even
downshift:  lda     63,dl
            sta     bp|0,sc
            sta     bp|0,sc
            tra     lower
nextup:     cmpq    128,dl
            tmi     upper
            lda     0,dl
            tra     pa
upper:      xec     tabup,ql
pa:         sta     bp|0,sc
enter:      ldq     ap|0,sc
            ttf     nextup
            tra     leave
            even
upshift:    lda     63,dl
            sta     bp|0,sc
            sta     bp|0,sc
            tra     upper
nextlo:     cmpq    128,dl
            tmi     lower
            lda     0,dl
            tra     pb
lower:      xec     tablo,ql
pb:         sta     bp|0,sc
            ldq     ap|0,sc
            ttf     nextlo
leave:
```

Tabup is a table of 128 entries. For each character printable in upper case, the corresponding entry in tabup is an lda instruction with dl modifier, and address containing the six bit character code. For each character printable only in lower case, tabup contains

tra downshift

Figure 1 (continued)

Tablo is a table of 128 entries. For each character printable in lower case, the corresponding entry in tablo is an lda instruction with dl modifier, and address containing the six bit character code. For each character printable only in ~~lower~~ upper case, tablo contains

```
tra  upshift
```