January 8, 1965

PROGRAMMING STAFF NOTE 40

SUBJ: Experimental Additions to the RUNOFF Command

FROM: J.H. Saltzer

A number of experimental features have been added to the control word language of the RUNOFF command, primarily to learn of possible directions in which such a language should develop. It is hoped that this note, and usage of the command itself will spark discussion and creative suggestions to aid in the development. The experimental features are described here, rather than in a publication for general distribution, since they are not necessarily smoothly implemented or bugfree, and they are subject to change pending reassessment of their value.

Although the ability to produce acceptable flow diagrams with a typewriter is limited, some useful things can be done even in this medium. The following control words are designed to simplify the composition of a flow diagram:

.figure

This control word turns control over to a figure processor, which creates in core memory a representation of a flow diagram under the control of a few special control words. When the control word ".end figure" is encountered, the completed picture is printed immediately on the page being generated if there is room on that page; otherwise the figure will appear at the top of the next page. Text following the ".end figure" control word will be smoothly attached to text before the ".figure"; no break is generated. (Restriction: if a figure is being held for placement at the top of the next page, another figure may not be encountered before the first one is printed.) The <u>only</u> control words which are recognized when in the figure processor are the following three:

.frame m n

This control word initializes the figure processor by giving the height and width of the figure to be produced. "m" is the height, in lines; and "n" is the width, in characters. (Note that a 1050 types 6 lines per inch, and 10 characters per inch.) Any attempt to place items in the picture which extend beyond the boundaries will cause an error comment to be generated. M and n must both be less than 100 and their product must be smaller than 5400. We may now think of the figure to be produced as an array of m x n elements.

.box i j

The text on the lines following this control word will be placed in the figure such that the first character on the first line following the ".box" will appear in row "i", character position "j". The end of the text is indicated by a ".box" control word for another piece of text or the ".end figure" control word. Temporarily, the text should not include underlined or overtyped characters.

.end figure

This control word causes control to return to the regular control processor of the RUNOFF command, for the decision to print the picture. Note that another ".figure" control word may not appear until after this figure has been printed.

One further control word has been added which is intended to facilitate bringing out revised editions of a memorandum.

.flag

The next line to be printed after this control word is encountered will have an asterisk placed two spaces to the right of the right margin, as illustrated. *

.define symbol

This control word defines the value of the symbol "symbol" to be the number of the page currently being printed. The symbol may be used later with the ".use" control word to cause printing of the page number in text. The characters in the symbol must be mappable into the six-bit character set, and all symbols must be six or fewer characters.

.use symbol

The value of the symbol "symbol" is inserted into the text with a single blank preceding and no blank following. If the symbol has not been previously defined, its value is "0". Text may continue following a blank typed after the symbol.

Here is an example of the use of these control words.

In one area of text:

> We now discuss the operation of the typewriter
> .define ref1
> coordinator module, which . . .

In a later area of text:

> As we saw in the discussion of the typewriter coordinator on page
> .use ref1 , the rest of . . .

If the first area of text were on page 14, the later line would read:

> As we saw in the discussion of the typewriter coordinator on page 14, the rest of . . .

Further Study

A number of suggestions have been made for extending the control word language of RUNOFF, and its capabilities. These are listed here, primarily to elicit comment and discussion, both on the language which describes these operations and the less important problem of their implementation.

1. Word division. This is a whole are of study in itself.

2. Automatic footnote insertion. This was handled somewhat awkwardly in the DITTO command, although the basic approach was probably reasonable.

3. Automatic page references, perhaps via some symbolic reference scheme. This would enable the page number in "as was described on page 32" to be inserted by the program. The analogy with an assembly program should be hotly pursued for ideas.

4. Special provision for printing facing pages. This would require alternate running heads, placing page

numbers alternately at right and left, and matching line counts on facing pages.

5. Improved page-division rules, to prevent the last line of a paragraph appearing alone at the top of a page, for example. At present, copy must be run off to check by hand that awkward page divisions have not been made.

6. Automatic generation of page numbers for a table of contents. Again, the analogy of an assembly program symbol table appears fruitful.

7. Automatic generations of an index. The problem here is obtaining too many references to a given word, many irrelevant.

8. Arrangement of tabulated data. This problem may have already been partly approached with the above-described figure generator, or the facilities already available in RUNOFF, but automatic setup of column widths and positions would be desirable. One could include in this category the ability to call on other programs to compute numbers to place in tables, although this is going pretty far afield.

9. Placing figures in a "cut" or inset. The control language is the most difficult problem here.

10. Equation typing and numbering. Again, the control language appears formidable.