Identification

Manuscript Printing under OS/360: ROFF
B. W. Kernighan

Description

ROFF is an OS/360 program for producing esthetically pleasing manuscripts from punched card source text. "Esthetic" includes provision of automatic lower case letters, automatic right margins on pages, automatic page numbering if desired, and numerous other formatting operations.

Introduction

The acquisition of a printer chain with lower case letters for the IBM 360 Model 67 at Princeton's Computer Center has made possible the production of high quality printed documents on the computer.

ROFF is a Fortran program for producing such printouts. ROFF operates on an input deck of alphabetic text, produced on an ordinary 029 keypunch, and produces a printed copy in manuscript form. Since the 029 keypunches do not have any direct provision for entering lower case letters, all input to ROFF is upper case; ROFF changes upper case letters into lower case when appropriate. For example, all the letters in any sentence are converted to lower case, with the exception of the first one. The conversion may be overridden by means of special "escape" characters, discussed below.

The output is formatted as the user wishes. He is able to start pages or paragraphs at will, produce blank lines, cause margins to be placed on the right side of the page, change the line length and indenting, and other functions of this sort.

These operations are all handled by "control words" which the user inserts in his input deck at the appropriate points. TSS users will recognize the ROFF control words as a nearly compatible subset of the RUNOFF commands.

Character mapping

Input to ROFF is a card deck, generally of upper case letters and punctuation. The contents of the input deck are converted to lower case as follows.

1. The first letter of each sentence is left in upper case. All other letters are set to lower case. A sentence is defined to be a set of characters ending in a period, followed by a blank, right parenthesis, single or double quote, or the end of a card, or a question mark or exclamation mark.

2. Upper case can be forced for the next alphabetic character encountered (A-Z only) by inserting a cent sign ¢ anywhere before the letter. Thus to capitalize a proper name in the middle of a sentence:

```
INPUT:  IT IS A LOUSY DAY IN¢PRINCETON
OUTPUT: It is a lousy day in Princeton
```

The cent sign evaporates, leaving a blank space in the output. The effect of the cent sign carries over all non-alphabetic characters, which is often useful for capitalizing letters in the middle of a string of digits.

3. Lower case may be forced at the beginning of a sentence by using a dollar sign. It will also vanish, leaving a blank space in the output.

4. An entire string of characters can be capitalized by preceding it with a circumflex (a 12-11-0-8-6 punch), which comes out as ^. The effect of the circumflex is terminated by the next blank character in the input. It also vanishes.

5. Any string of characters can be underlined by preceeding it with an underscore (_). This operates in the same fashion as the circumflex.

For both the underscore and the circumflex, a cent or dollar sign is considered to be a blank. Thus to underscore and capitalize the first letter, it is necessary to use '¢_' rather than '_¢'. It is also necessary to use '^_'.

6. Arbitrary strikeovers may be created by using the at-sign '@'; the at-sign is roughly equivalent to the backspace key on an ordinary typewriter. For example, to make a less than or equal to sign, use <* to produce ≤ (which can also be done by _<).

If a sequence is to be overstruck, place all of the at-signs together. If one of the overstrike characters is an underscore, make it last rather than first.

Caveat: The at-sign feature is *not* guaranteed; use it cautiously.

7. Any percent sign '%' in the input is treated as a non-blank character, but vanishes on output. This is often useful as a place holder: if the space between two words of input is filled with percent signs, the program will not insert or delete any extra blanks between the words in the output. The percent sign can also be used to reserve space in the output for later insertion by hand of various special symbols which are not on the printer.

Control words

The format of the output may be controlled by "control words" as in RUNOFF. To distinguish control words from the rest of the text, they must appear on their own individual cards. Control words all have the same format -- a period in card column 1, followed by a two letter abbreviation for the control word in columns 2 and 3, and (sometimes) an operand anywhere in columns 4-80. No other text may appear on a control word card. Control words affect the printed format, but are never printed themselves.

The following is a description of the currently available control words, in alphabetical order.

In this discussion, the word "break" associated with a control word will indicate that the two input cards between which the control word lies will not be run together, as they normally would be in 'FILL' mode. Thus at a break, all text read so far will be printed out, and all following input text will appear on a new line of output.

"Default" means the value of a parameter that ROFF assumes if it is not specified explicitly by the user. For example, the default value of line spacing is single space, unless the user sets it to double space.

.AD adjust
break, turn on mode in which all text is right justified by inserting blanks and moving input words as necessary. ROFF starts in adjust mode. When adjust is turned on, so is "fill".

.BP begin page
break, start next line on a new page. Capitalize first letter of next line.

.BR break
current line is printed out; the next input line is started on a new line with the first letter of that line capitalized. This also signals the start of a new paragraph.

.CO copy
enter mode in which all text (except control words) is printed in upper case (no mapping to lower case) and escape characters have no effect.

.DS double space
break, print succeeding output double spaced.

.EF end of file
break, terminate job. This should be the last card in the input text.

.FI fill
break, move words from following cards as necessary to place as many words as possible on each line of output. ROFF starts in fill mode.

.IN n indent
break, print the following text indented n spaces (n spaces from the normal position.) Default is n=0, which restores non-indenting.

.LL n line length
break, set output line length to n characters. Initial value for n is the default value, which is 60.

.MA map
enter character mapping mode, the inverse of copy. ROFF starts in map mode.

.NE n need
if n lines are left on the current page, no action. Otherwise, break, and skip to a new page. Default: n=0.

.NF no fill
break, turn off fill mode. Only mapping takes place; no words are moved.

.NJ no justification
break, turn off right justification of margins. Nojust also turns off fill.

.PA n page
break, start next line on a new page numbered n. Default: n=1. Capitalize first word on new page.

.PM n paging mode
if n=1, print page numbers at the top of each page. If n=0, don't print page numbers, but continue computing them. Default is n=1.

.PP n paragraph
break, start a new paragraph with initial indentation n spaces relative to current indent value. If n is defaulted, use the previous value for paragraph indenting. Initially n is 5. Set capitalization on.

.SK n skip
at the first opportunity, skip n blank pages. Default: n=1. If further skips are encountered before the previous ones are executed, the values of n are added, and all are executed at the first opportunity.

.SP n space
break, insert n blank lines. Default: n=1. If the request cannot be satisfied on the current page, a skip to a new page is executed first.

.SS single space
break, enter single space mode. ROFF starts in single space mode.

.TR c1 c2 translate

henceforth, when 'c1' is encountered as the output is about to be printed, convert it to 'c2' for printing. c1 and c2 may be any arbitrary characters, except that c1 cannot be blank. Spaces are allowed between c1 and c2. ROFF starts with (effectively) '.tr% '.

.^

Print next word all in capitals. Useful primarily in nofill mode, to avoid an unwanted space in first column of output.

._

Underscore next word. Useful primarily in nofill mode, to avoid an unwanted space in first column of output.

General use of ROFF

To use ROFF, create the input deck as described in this manual, using control words and escape characters as needed. Remember to reset any paramters which you want to be different from ROFF default values.

Set up a job deck as follows:

details will be filled in when the system stabilizes.

The page count can be estimated as about #cards/25 for typical double space text, #cards/50 for single space. ROFF processes between 700 and 1000 cards/minute, so T on the job card can be set accordingly.

Hints and warnings

1. As a general rule, place each sentence on a separate card if running in fill or adjust mode. This makes editing the deck significantly easier.

2. The at-sign feature is not guaranteed, since there are some subtle interactions between it and other escape

characters. It may be safer to use a '.tr' if one of the overstrike characters is to be an escape character.
Also, don't try to overstrike by using 'c1@c2c3@c4...'. Extra and probably unwanted spaces will be inserted between the characters. Put all characters, then all at-signs, then all overstrikes.

3. Only one overstrike is made for any given character.

4. RUNOFF users should beware of the few incompatibilities and differences of conventions. They are largely concerned with the problems of automatic capitalization.

5. The percent sign is very useful for controlling spaces when in fill or adjust mode. Its use can prevent insertion of blanks where they would spoil the format of the output.

Disclaimer

ROFF was designed specifically for printing the author's doctoral disseration, and hence is not as general and flexible as it might be if designed for a wide class of users.

Interested users who feel that the published version does not satisfy their precise needs are invited to modify it as they see fit. The author will be happy to supply a ROFF source deck (entirely in Fortran, with comments) as a starting point.

Any comments, criticisms, and suggestion for improvement of the product and the manual, reports of bugs, and so on, are welcome. Send them to

B. W. Kernighan
Engineering Quadrangle, Room B404
Princeton University
or
Bell Telephone Laboratories
Room 2C521
Murray Hill, N. J. 07974.