1971 BCPL Roff Manual

The manual reproduced below records a stage in the evolution of the family of text-formatting programs from Jerry Saltzer's runoff through roff and Joe Ossanna's nroff/troff to Gnu's nroff/groff. In writing this program in BCPL for Multics and GECOS, I imitated and expanded on Saltzer's model. Dennis Ritchie based the PDP-7 Unix implementation of roff on my program as it was circa 1970.

The name of the program changed from “runoff” to “roff” between the original version of September 17, 1969 and this one of January 12, 1971. Features introduced in the interim include automatic hyphenation, footnotes and merge patterns. These requests are new:

.hy, .hc, .fn/.en, .mg, .ls, .ju/.nj

These requests have been dropped from the manual:

.ss/.ds undocumented in favor of .ls
.ad/.na undocumented in favor of .ju/.nj
.eo/.oo (even and odd page offsets) completely abandoned

In a puzzling demonstration of the new .mg request, unexplained marginal asterisks flag changes from a lost intermediate state of the program.

The pages below were scanned from my personal copy. The scrawled annotation on page A-6 suggests a future amendment: “The input that did this looks innocent enough, but the two long underlined stretches were done by backspacing over the whole stretch and then underlining it in one go.” A paragraph on page A-4 is marked because it only partially makes its point: a break is needed before both instances of .in.

M. Douglas McIlroy
September 17, 2019

COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE Roff

MM 71-1371-2

CASE CHARGED— 39199

FILING CASES— 39199-11

DATE— January 12, 1971

AUTHOR M. D. McIlroy
Ext. 6050

FILING SUBJECTS— Computer editing
Publication formatting

ABSTRACT

Roff is a publication formatting program for GECOS that is considerably more flexible than its competitors. It does page, paragraph and line layout, titling, footnoting, hyphenation and tabulation, plus some other specialized formatting tasks to the user's specifications. It runs in batch or time sharing from ascii or BCD data. This manual is a major revision of MM-69-1371-12. It describes new automatic hyphenation and footnoting capabilities, and includes a tutorial appendix.

Text - 5 pages
Attached - 2 appendices

SEE REVERSE SIDE FOR DISTRIBUTION LIST

Bell Laboratories

subject: Roff - Case 39199-11

date: January 12, 1971

from: M. D. McIlroy
MM-71-1371-2

MEMORANDUM FOR FILE

Purpose: Roff is a simple publications formatting routine. This manual is an example of its work.

Input: Text to be printed and instructions for printing occur intermixed on separate lines. Instruction lines, called requests, begin with a control character (usually ".").

Output: Output lines may be filled as nearly as possible with words without regard to input lineation, or may be copied one-for-one from input text. Right margin justification may be performed on filled lines. Breaks between output lines are forced by certain requests and by input text lines beginning with blanks. Indentation, centering, line length, line spacing, page length, titling, page numbering, hyphenation and footnotes are controllable.

Usage: This dialog works on TSS with TSS files.

```
SYSTEM ?roff
input file name? name1
output file name? name2
skip how many pages? number
load paper, hit return
```

An empty name1, name2 or number means respectively *SRC, print at terminal, or start at first page. To avoid the questions, respond

```
SYSTEM ?roff name1
or SYSTEM ?roff name1,name2,number
```

To do the printing off line use the special "qed proff" command[8].

```
SYSTEM ?./qed ./proff name1
```

Batch: Roff works in batch GECOS, reading either TSS files or card files as input, and producing output for line printers, model 37 teletypes or the IBM 360 printer. This sample deck runs the program. Input is normally cards on file IN, output is normally on the printer. To print n thousand lines, n>2, add $ LIMITS n,20 after the $ SELECT.

```
$       IDENT
$       SELECT  ./ROFF
option card (not required)
$       PRMFL   IN,R,S,cat/file
$       ENDJOB
```

One or more of these options may be specified on the option card, separated by blanks:

RTSS read input file IN in GE-TSS format
WTSS write output file OT in GE-TSS format

Best copy: Final printing for publication is best done on the IBM 360/30. To prepare a magnetic tape for the 360 from a TSS file type:

```
SYSTEM ?./qed ./proff360 filename tapenumber
```

In batch GECOS prepare a magnetic tape this way.

```
$       SELECT  ./ROFF
option card (not required)
$       PRMFL   IN,R,S,cat/file
$       SELECT  ./PRINT360
$       TAPE9   OT,R1D,,tapenumber
```

Submit the resulting tape for printing on the 360 with a DOS job deck like the one below. Ask for quality print using the text train.

```
// JOB job.number name.and.extension
* INPUT TAPE IS tape.number
* 1 PART UNLINED PAPER
// PAUSE MOUNT INPUT TAPE ON 180
// EXEC M0335D
/&
// PAUSE
```

The character ~ does not print on the text train.

GUTS: The GUTS[7] command ROFF, issued with file IN (and perhaps I* or OT) appropriately attached invokes Roff. Output ordinarily goes into P* and the working file.

Requests: Signs + or - may precede numbers denoted +n; signed numbers augment, and unsigned replace current values. n=1 and t='''' are assumed for missing fields. Certain requests cause line breaks. Default settings are in effect initially.

request	break	default	meaning	
.ar	no	yes	arabic page numbers	
.bm +n	no	n=4	bottom margin is n lines below foot	
.bp	yes		begin page	
.cc c	no	c=.	control character becomes c	
.ce n	yes		center next n lines, break on each	
.cn	no	yes	case normal on input	
.cr	no	no	case reversed, exchange upper and lower case letters only on input	
.ef t	no	t=''''	even page foot title is t	
.eh t	no	t=''''	even page head title is t	
.en	yes		end of footnote	*
.fi	yes	yes	fill output lines	
.fn	no		beginning of footnote	*
.fo t	no	t=''''	all foot titles are t	
.fs t	no	t=''''	footnote separator is t (see p4)	*
.hc c	no		hyphenation character is c (see p4)	*
.he t	no	t=''''	all head titles are t	
.hy n	no	n=3	hyphenation mode is n, $0 \leq n \leq 3$	*
.in +n	no	n=0	indent margin n spaces	
.ju	yes	yes	justify right margin	*
.li n	no		literal, treat next n lines as text	
.ll +n	no	n=60	line length including indent is n	
.ls n	yes	n=1	line spacing is n	
.mq n	no	empty	next line sets merge pattern n, $1 \leq n \leq 4$	
.ne n	no		need room for n output lines with present spacing, do .bp if necessary.	
.nf	yes	no	nofill, break on each input text line	
.nj	yes	no	no justification of right margin	*
.of t	no	t=''''	odd page foot titles are t	
.oh t	no	t=''''	odd page head titles are t	
.op	yes		begin an odd page	
.pa +n	yes	n=1	begin page n	
.pl +n	no	n=66	paper length is n including margins	
.po +n	no	n=0	page offset is n, i.e. move all output n spaces right	
.ro	no	no	roman page numbers	
.sc c	no	c=#	shift character is c (see p5)	
.sk +n	no		skip at next new page to page n	
.sp n	yes		insert n extra spacing lines	
.ta	no	all	tabs set by next line (see p5)	
.ti +n	yes		temporary indent, for one line only	
.tm +n	no	n=4	top margin is n lines above head	
.tr cd...	no		translate c into d (see p4)	
.ul n	no		underline alphanumerics, next n lines	

Synonyms: .br=.sp 0 break, .ds=.ls 2 double space, .ss=.ls 1 single space, .un n=.ti-n undent

Titles: Running titles appear at top and bottom of every page, separated from the body by one extra blank line. Head and foot titles for even- and odd-numbered pages are all independently settable by patterns of the form:

'part1'part2'part3'

Part1 is left justified, part2 centered, and part3 right justified with respect to the margins current when the title was set. Any % sign in a title is replaced by the current page number. Any nonblank may serve as a quote.

Translate: Every output character is translated, normally into itself. The request .tr cdcd... specifies that characters c are to be translated into corresponding characters d. The first c must not be a space.

Merge: Fixed information to be put in all output lines may be specified by merge patterns. Blank positions of every nonempty output line are replaced from merge pattern 1, then positions still blank are replaced from merge pattern 2, and so on.

Merge pattern n is given by a special input line immediately after the request .mg n. The pattern is permanently positioned according to the indentation and page offset current when the pattern was set. All merge patterns are initially empty.

Footnotes: Lines between .fn and .en are formatted normally then held for the bottom of the page. All requests except .mg are effective within footnotes. Line formatting is handled completely independently in body and footnote text. In particular, separate settings are maintained for .ce, .fi, .in, ,ju, .ll, .nf, .nj, .ti and their synonyms. The footnote separator (normally a blank line) between text body and footnotes is set like a title by .fs.

Hyphens: The hyphenation mode, set by .hy n, controls attempts to split words at the end of filled lines. Breaks may be inserted within words:

if n=0, nowhere
if n≥1, at -'s or at hyphenation characters
if n≥2, before certain suffixes
if n≥3, between certain pairs of letters

The hyphenation character, set by .hc, disappears from output.

Tabs: Tab characters in input separate fields, which may be left justified, right justified or centered on specified positions of the line. The first field is always left justified; other fields are positioned according to tab stops set by a special input line following a .ta request. Characters 'l', 'r' and 'c' in this line denote left, right and centered tab stops. Initially 'l' stops are set in every column.

Backspace: Results with backspace are not guaranteed when a space is overstruck, nor when backspace appears elsewhere than in text or titles.

Blanks: Trailing blanks are stripped from input lines. If no characters remain, .sp is assumed; otherwise one blank is appended, or two after a period. *

Errors: Only file errors cause stops or diagnostics; other errors have reasonable symptoms. Lines are limited to 360 characters, footnotes to 800. *

Cards: Card columns 73-80 are ignored. Each character is read as lower case ascii, unless immediately preceded by a shift character (usually #). If the last nonblank on a card is a shift, then the next card is taken to be a continuation appearing in place of the shift. The table below lists all meaningful characters and their punched representations.

Card code	0	1	2	3	4	5	6	7	8	9	[	#	@	:	>	?
Lower case	0	1	2	3	4	5	6	7	8	9	[	#	@	:	>	?
Upper case	0	1	2	3	4	5	6	7	8	9	[	#	@	:	>	?
Card code	sp	A	B	C	D	E	F	G	H	I	&	.	]	(	<	\
Lower case	sp	a	b	c	d	e	f	g	h	i	&	.	]	(	<	\
Upper case	ht	A	B	C	D	E	F	G	H	I	&	.	]	{	<	~
Card code	↑	J	K	L	M	N	O	P	Q	R	-	$	*	)	;	'
Lower case	|	j	k	l	m	n	o	p	q	r	-	$	*	)	;	'
Upper case	ff	J	K	L	M	N	O	P	Q	R	bs	$	^	}	;	`
Card code	+	/	S	T	U	V	W	X	Y	Z	←	,	%	=	"	!
Lower case	+	/	s	t	u	v	w	x	y	z	_	,	%	=	"	!
Upper case	nl	/	S	T	U	V	W	X	Y	Z	_	,	%	=	"	!

bs = backspace, ff = formfeed (not useful), ht = horizontal tab, nl = newline, sp = space

MH-1371-MDM M. D. McILROY

Attached
Appendix 1, 2

January, 1971

Acknowledgements, etc.

Roff derives from the Runoff program by J. E. Saltzer[1]. Other ideas in it spring from observations by B. W. Kernighan, K. L. Thompson and J. N. Sturman. It owes some debt to [4] and [6], which also descend from [1].

The program was written in BCPL[2,3], which was upgraded by D. M. Ritchie to cope with Runoff's special typographic needs. The auxiliary program for use with the 360, which is described in Appendix 2, was written by J. N. Sturman. Mr. Sturman also put Roff into GUTS[7]. The programming for TSS and TSS files was done by S. C. Johnson and Mrs. M. P. Wagner. The novel hyphenation procedure is due to Mrs. Wagner.

Roff is similar to other programs -- Runoff[4], Mace[5] and Pubedit[6] -- available under GECOS. Unlike them, it accepts input from cards and ordinary Teletypes as well as Model 37 Teletypes, it works in batch or time-sharing, and it is not bound to a peculiar file format.

Roff's facilities for running titles, page numbering, widow suppression, footnoting and figure insertion are superior to all but Pubedit's. Its merging, tabulation and hyphenation capabilities are unrivaled. Pubedit has the unique advantage of supporting layout macros; Runoff handles multiline titles; Mace has font control and proportional spacing with the ultimate intent of driving typesetting equipment as well as computer output devices.

---

[1] P. A. Crisman, The Compatible Time-Sharing System, M. I. T. Press (1965), section AH.9.01
[2] M. Richards, BCPL: a tool for compiler writing and system programming, AFIPS Conference Proceedings 34 (1969), pp 557-566
[3] R. H. Canaday and D. M. Ritchie, Bell Laboratories BCPL, MM-69-1371-7
[4] GE-625/635 GECOS-III Time-Sharing Text Editor, General Electric Documentation Standards and Publications CPB 1515A, Phoenix (1968)
[5] J. E. Miller, J. Kohut, C. C. Lochbaum and M. V. Mathews, Machine Aided Composition and Editing, Bell Telephone Laboratories, Murray Hill (1969)
[6] R. J. Elliot and A. Kaiman, Computer aided publications editor, MM 69-1373-8
[7] J. N. Sturman, A Users' Manual - GERTS Users' Terminal System, September 1969
[8] A. D. Hall, S. C. Johnson, B. W. Kernighan, D. M. Ritchie, Utility programs for GE-TSS users, MM-70-1373-10

Tutorial Guide

The title line, "Tutorial Guide," and this first paragraph were set up by the input that follows. The title was spaced down one extra line (.sp) and centered (.ce). The paragraph begins after another spacing line and is double spaced (.ds). I typed the paragraph without attention to lines; Roff took care of filling them. I put each sentence on a separate line to make editing easier.

```
.sp
.ce
Tutorial Guide
.sp
.ds
The title line, "Tutorial Guide," and this first
paragraph were set up by the input that follows.
The title was spaced down one extra line (.sp) and
centered (.ce).
The paragraph begins after another spacing line
and is double spaced (.ds).
I typed the paragraph without attention to lines;
Roff took care of filling them.
I put each sentence on a separate line to make
editing easier.
```

Heads and feet. The footing lines for this page and the opposite one were specified by two requests, .ef and .of, for even- and odd-numbered pages. Similar requests, .eh and .oh, set up the headings. Quote marks split the foot up into parts for the left and right side of the page. Between the two middle quote marks is stuff (nothing in this case) for the middle of the line. The % mark shows where the page number is to go.

```
.ef 'A-%''January, 1971'
.of 'January, 1971''A-%'
```

If you want the same heading on all pages, then a single .he request will do. This one sets up the standard style of page numbering for BTL memoranda. It should be placed somewhere after the beginning of the first page so as to cause numbering of pages 2, 3, and so on.

```
.he ''- % -''
```

Line breaks. In ordinary paragraph text, Roff takes care of filling lines with words, but there are places where one wants to control exactly where the lire ends. The title on page A-2, "Tutorial Guide," should have a line tc itself, and each paragraph is expected to begin on a new line, not just run on to the paragraph before. Certain requests, among them .ce and .sp, cause line breaks. The list on page 3 tells exactly which.

Sometimes every line break is important. One could place a simple break (.br) request between every pair of lines, as in X. J. Kennedy's first stanza below, but it is easier to shut off all filling of lines by .nf (no fill) as in the second. .fi turns filling on again.

```
In a prominent bar in Secaucus one day
.br
Rose a lady in skunk with a topheavy sway,
.br
Raised a knotby red finger--all turned from their beer--
.br
While with eyes bright as snowcrust she sang high and clear:
.sp
.nf
'Now who of you'd think from an eyeload of me
That I once was a lady as proud as could be?
Oh, I'd never sit down by a tumbledown drunk
If it wasn't, my dears, for the high ccst of junk.'
.fi
```

Justification and indentation. All the paragraphs up to now have been justified, that is enough extra spaces have been put into every line to align the right margin exactly. Ordinary typewriter style with an uneven right margin results from a no justification (.nj) request. Lines may still be filled even with justification off, but justification is never done when nofill (.nf) is in effect.

```
.nj
.ti 10
```

This paragraph is not justified. It was surrounded by the no justification (.nj) and justification (.ju) requests that you see, made visible by some trickery. The first line was indented 10 spaces by a temporary indent (.ti) request that holds for exactly one line.

```
.ju
```

```
.ss
.in5
.ll55
.ti+5
```

This paragraph was indented (.in) 5 spaces and the right margin was placed at 55 instead of the usual 60 by setting the line length (.ll). (The line length includes the indentation.) Indentation and line length may also be changed by addition and subtraction as in .ti+5. Notice that a break (in this case caused by .sp) precedes the resetting of margins after the paragraph. Neither .in nor .ll cause a break, and strange things can happen if you unconsciously assume they do!

```
.sp
.ll60
.in0
```

Pages and widows. Several requests affect the numbering of pages, or cause a new page to begin.

```
.bp begins a page
```

.pa n begins a page and sets its number to n

.sk n sets the number of the next page to n, but continues building the present page

Both .pa and .sk (skip) allow relative settings. Unless the next page number has been set by .sk, .bp has exactly the same effect as .pa+1. .sk is useful for setting aside space for full page illustrations.

Ordinarily Roff begins a new page only when the page before is filled up. It can very easily create widows, isolated lines that really belong with the page before or the page after. Typical cases are a subhead that falls at the bottom of a page, or the last couple of words of a paragraph that fall at the top. When a widow does crop up, you can fix it with a need (.ne) request. In the next example, .ne 4 says no more lines should be put on the current page unless four lines can be put there, thus assuring that at least the first two lines of the single-spaced (.ss) paragraph go along with the subhead.

```
.ss
.ne 4
.ce
Control of Radioactive Pollution
.sp
No really effective therapy is known for preventing or
curing the harmful effects of internal contamination
by radioactive nuclides.
```

Need requests come in handy as well for guaranteeing that a table or a set of equations not be split across pages.

Underlining. The best way to do underlining is to use the .ul request, which causes the next line of input to be un-

derlined regardless of line filling. Putting underlines into filled text by backspacing and overstriking is a ticklish procedure that can give troublesome effects, as this sentence shows.

```
which causes the
.ul
next line of input
to be underlined regardless of line filling.
Putting underlines into filled text by
backspacing and overstriking is a
ticklish procedure
that can give troublesome effects, as this
sentence shows.
```

Punctuation characters are not touched by .ul. To underline punctuation you will have to overstrike, observing the cardinal rule for typing filled text: Never backspace across a space.

Tabulation. Tab stops may be set in Roff as on a typewriter, but in addition to creating columns of data aligned along their left edges, tabs may also create columns aligned along their right edges, or columns aligned by centering. The alignment is done during input before any other formatting. Thus tabs work best in ncfill mode, but filling of the rightmost column is quite possible. (For an example, see the tabbed, filled, but unjustified table of requests on page 3.)

Tabs are set by a .ta request together with the immediately following line, which shows in what columns tabs are to be set. Tabs for left-aligned columns of data are set by an "L" in the leftmost position of the column, for

right-aligned columns by an "R" in the rightmost position, and for centered columns by a "C" in the middle position. Tabs are actuated by the teletype TAB character. It is not necessary for your teletype tab stops to be correctly positioned; Roff replaces all tabs by an appropriate number of spaces.

In the example below, I wish to right-align the left sides of equations, (left-) align the equals signs, and right-align the equation numbers. To make them visible, I have replaced tab characters by colons. Notice that there must be a tab character wherever spaces are to be inserted, including the space at the left margin before the first tabbed field.

```
.nf
.in5
.ta
        R I                                R
:sin x:= x - x**3/3! + x**5/5! - ...:(9)
:sinh x:= x + x**3/3! + x**5/5! + ...:(10)
```

Here is the result.

```
      sin x = x - x**3/3! + x**5/5! - ...      (9)
     sinh x = x + x**3/3! + x**5/5! + ...     (10)
```

Footnotes. Footnotes can be placed anywhere; Roff collects them and places them at the bottom of the page. A footnote is set off by .fn and .en (end note). With only minor exceptions, .fn, .en and everything in between are completely parenthetical to the surrcunding text. Indentation, line length, filling, justification, single- and double-spacing, line breaks, etc. are all independently handled in footnotes

and their settings are remembered from footnote to footnote. You may want to include a .ne request to assure that a footnote and the reference* to it have room on the same page, as I did here.

```
It is good practice to include a .ne request to
assure that a footnote and the
.ne 3
reference*
.fn
* Otherwise it is possible for some or all of the
footnote to spill over to the next page, or for the
line containing the reference to get pushed over.
.en
to it have room on the same page, as I did here.
```

Because .ne counts in units of the current line spacing (in this case double spacing), 3 lines is enough to take care of the reference, the space between body text and footnotes, and the three-line footnote.

Literal. Occasionally you may want to type a line beginning with a period, which Roff will try to interpret as a request--even when no such request exists. A literal (.li) request put right before the line will prevent Roff from mistakenly trying to interpret it and cause the line to be accepted as ordinary text.

```
.li
.sp this couldn't be printed without .li
```

Hyphenation. To improve the fit, Roff ordinarily tries to hyphenate at the end of filled lines. Errors of commission

* Otherwise it is possible for some or all of the footnote to spill over to the next page, or for the line containing the reference to get pushed over.

and omission sometimes happen, as here where I have set the column width to 6 letters.

```
ces-
spool
cour-
thouse
tetrabromomethane
```

Hyphenaticn mode (.hy) requests control the boldness with which Roff inserts hyphens. Words are never broken under .hy 0; words are quite frequently broken under .hy 3. The easiest way to recover from a hyphenation gaffe is to bracket the offending word by .hy 0 and .hy 3. If you find automatic hyphenation generally displeasing, the best setting will probably be .hy 1. In that mode already hyphenated words, like "run-of-the-mine," may be broken, but nothing else will.

Other places may be marked as candidates for hyphenation by means of a hyphenation character, an otherwise unused character, designated in a .hc request. Setting .hc \ and supplying syllabification for the unfortunate "tetra\bromo\meth\ane", I can arrange for satisfactory hyphenation with almost any column width. Here the widths are 6, 12 and 18.

```
 6 tetra-
   bromo-
   methane
12 tetrabromo-
   methane
18 tetrabromomethane
```

Translation, described on page 5, can be used to overcome some unpleasing results of justification and filling, such as

```
(1)   extra spaces between an "undented"
    paragraph number and its paragraph
(2)  an unwanted break inside a formula: 1 +
    x + x**2/2! + x**3/3! + ...
```

To fix up such annoyances, you may designate an otherwise unused character, say $, to be translated (.tr) into a space. Though it will print as a space, Roff does not treat it as such, and so won't pad or break lines there.

```
.ll 50
.in 10
.tr $
.ti-4
(1)$extra spaces between an "undented" paragraph
number and its paragraph
.ti-4
(2)$an unwanted break inside a fcrmula:
1$+$x$+$x**2/2!$+$x**3/3!$+$...
```

The prettied-up result follows.

```
(1) extra spaces between an "undented"
    paragraph number and its paragraph
(2) an unwanted break inside a formula:
    1 + x + x**2/2! + x**3/3! + ...
```

Page offset (.po) requests come in handy when the machine you use for printing places the output too close to the left side of the paper. A page offset request before everything else will move all the output right. I used .po 10 for this manual.

Merge patterns, the use of which is described on page 5, accomplish various special effects. The vertical bars surrounding this paragraph were set by the following pattern, except that the spacing of bars was wider. I used .po to emulate a typewriter's margin release and place text left of the normal .in 0 position.

```
.po-3
.mg 1
|                    |
.po+3
```

Envoi. This appendix only hits the high spots. It will get you off the grcund, but to get the most out of Roff, you will want to study the manual proper to learn all its capabilities. I entertain comments on Roff and the manual. Happy Roffing.