

## TABLE OF CONTENTS

### 1. UNIX COMMANDS

ar	archive & library maintainer...3
as	assembler...3
bas	basic...3
bc	arbitrary precision interactive language...3
cat	concatenate & print...3
cc	C compiler...3
cdb	C debugger...3
chdir	change working directory...3
chmod	change mode of file...3
cmp	compare 2 files...3
comm	print lines common to 2 files...3
cp	copy...3
cpall	copy files into directory...3
cref	make cross reference listing...3
date	print or set date...4
db	debug...4
dc	desk calculator...4
dd	convert & copy a file...4
diff	differential file comparator...4
dsw	delete interactively...4
du	summarize disk usage...4
echo	echo arguments...4
ed	text editor...4
eqn	typeset mathematics...4
exit	terminate command file...4
fc	Fortran compiler...4
fed	edit form letter memory...4
file	determine file type...4
find	find files...4
form	form letter generator...5
goto	command transfer...5
grep	search a file for a pattern...5
gsi	interpret extended character set for GSI terminal...5
if	conditional command...5
kill	terminate a process...5
ld	link editor...5
ln	make a link to a file...5
login	sign on...6
ls	list contents of directory...6
m6	general purpose macroprocessor...6
mail	send mail to designated users...6
man	run off section of UNIX manual...6
mesg	permit or deny messages...6
mkdir	make a directory...6
mv	move or rename a file...6
neqn	typeset mathematics on terminal...6
newgrp	log in to a new group...6
nice	run a command at low priority...6
nm	print name list...6
nohup	run a command immune to hangups...6
nroff	format text...6
od	dump a file...6
opr	off line print...7
passwd	change login password...7
pfe	print floating exception...7
pr	print file...7
prof	display profile data...7
ps	process status...7
ptx	make permuted index...7
pwd	print working directory name...7
rc	Ratfor compiler...7
rev	reverse lines of a file...7

rm	remove files...7
rmdir	remove directories...7
roff	format text...7
sed	stream editor...8
sh	command interpreter...8
shift	adjust shell arguments...8
size	size of an object file...8
sleep	suspend execution for an interval...8
sno	Snobol interpreter...8
sort	sort or merge files...8
spell	find spelling errors...8
split	split a file into pieces...8
strip	remove symbols and relocation bits...8
stty	set typewriter options...8
sum	sum a file...8
tail	print end of a file...8
tbl	format tables for nroff or troff...8
tee	pipe fitting...9
time	time a command...9
tmg	compiler-compiler...9
tp	manipulate DECtape and magtape...9
tr	transliterate...9
troff	format text for phototypesetter...9
tty	get typewriter name...9
typo	find possible typos...9
uniq	report repeated lines in a file...9
wait	await completion of process...9
wc	word count...9
who	who is on the system...9
write	write to another user...9
yacc	yet another compiler-compiler...10

## 2. COMMAND DETAILS

ed	text editor commands...11
eqn,neqn	key words for mathematics typesetter...11
db	debugger commands...12
cdb	C debugger commands...12
fc	Fortran runtime diagnostics...12
sh	command interpreter special characters...13

## 3. TROFF & NROFF

troff,nroff	text formating commands...15
	escape sequences...17
	reserved registers...18
	roff simulator...18
	special characters...19
	macros for manuscripts...20

## 4. C SYSTEM CALLS...21

## 5. SITE DEPENDENT COMMANDS

### a. Center 127

cb	C beautifier...23
factor	find prime factors in a number...23
fget	retrieve files from HIS 6000...23
fsend	send files to HIS 6000...23
lbn	submit off-line job to HO IBM 370...23
iget	get files from HO IBM 370...23
isend	send files to HO IBM 370...23
nfs	communicate with Spider file store...23
tcatsim	simulate phototypesetter of Tektronix terminal...23
tekstare	convert Tektronix picture to hard copy graphics...23
tss	call MH-TSS...23

**ar**  $\begin{cases} d \\ r \\ t \\ u \\ x \end{cases}$  [v] *af file file ...*  $\begin{cases} d = \text{delete} \\ r = \text{replace} \\ t = \text{tabulate} \\ u = \text{update} \\ x = \text{extract} \\ v = \text{verbose} \end{cases}$

**as** [-] *file ...* [- causes undefined symbols to be global

**bas** [*file*]

**bc** [-l] [*file ...*] [-l loads the function library

**cat** *file ...*

**cc** [-c] [-p] [-f] [-O] [-S] [-P] *file.c...[-l] ofile...*  
 -c = suppress loading  
 -p = profile  
 -f = floating-point interpreter  
 -O = optimize  
 -S = keep assembler code  
 -P = just preprocess  
 -l = loader options

**cdb** [*a.out [core]* ]

**chdir** *directory*

**chmod** *octal file ...* octal mode is an OR of the following:  
 4000 set user id on execution  
 2000 set group id on execution  
 1000 save shared pure-procedure programs  
 0700 read, write, execute by owner  
 0070 read, write, execute by group  
 0007 read, write, execute by others

**cmp** [-l] [-s] *file1 file2*  
 -l = print byte number and differing bytes  
 -s = print nothing; return codes  $\begin{cases} 0 = \text{same} \\ 1 = \text{different} \\ 2 = \text{problem} \end{cases}$

**comm** [-[123]] *file1 file2*  $\begin{cases} -1 = \text{suppress lines only in file1} \\ -2 = \text{suppress lines only in file2} \\ -3 = \text{suppress lines in both files} \end{cases}$

**cp** *oldfile newfile*

**cpan** *file ... directory*

**cref**  $\left[ - \left[ \begin{cases} a \\ c \\ o \\ s \\ l \\ t \\ u \\ x \end{cases} \begin{cases} i \\ o \\ s \\ l \\ t \\ u \\ x \end{cases} \begin{cases} 1 \\ 2 \\ 3 \end{cases} \end{cases} \right] \right] .file ...$   
 a = assembler format (default)  
 c = C format  
 i = ignore symbols in *ifile*  
 o = list symbols in *ofile* only  
 s = current symbol in column 3 (default)  
 l = line numbers in column 3  
 t = user supplied temporary files  
 u = only output unique symbols  
 x = only output C external symbols

1 = sort on column 1 (default)  
 2 = sort on column 2  
 3 = sort on column 3

**date** [*s*] [*mmddhhmm[yy]*]      *s* = set date from TIU interface  
 set month-day-hour-minute-year  
 no arguments - print date

**db** [*core [namelist]*] ||**-**      **-** for non-core image with namelist

**dc** [*file*]

**dd** [*option = value*] ...      options are:  
*if*= input file  
*of*= output file  
*ibs*= input block size (default 512)  
*obs*= output block size (default 512)  
*bs*= block size  
*cbs*= conversion block size  
*skip*= skip *n* input records  
*seek*= skip *n* output records  
*count*= copy only *n* input records  
*conv*=ascii,ebcdic,lcase,ucase,swab,sync,noerror

**diff** [**-**] *file1 file2*      **-** output ed commands to make *file2* from *file1*

**dsw** [*directory*]

**du** [**-s**] [**-a**] [*file* ...]      **-s** = only give grand total  
**-a** = give entry for each file

**echo** [*arg* ...]

**ed** [**-**] [*file*]      **-** suppresses character count on **e**, **r**, **w** commands

**eqn** [*file* ...]

**exit**

**sed**

**sc** [**-c**] *file.f* ... [**-l**] *ofile* ...      **-c** = suppress loading  
**-l** = loader options

**file** *file* ...

**find** *pathname expression*

*expression* is made of the following primitives  
 where *n* is a decimal integer and **+n** means more  
 than *n*, **-n** means less than *n*  
**-name** *filename*      true if *filename* matches current file  
**-perm** *onum*      true if permission flags = *onum* (octal)  
**-type** *c*      true if file type is **b,c,d,f**  
**-links** *n*      true if file has *n* links  
**-user** *uname*      true if file belongs to *uname*  
**-group** *gname*      true if file belongs to *gname*  
**-size** *n*      true if file is *n* blocks long  
**-atime** *n*      true if file has been accessed in *n* days  
**-mtime** *n*      true if file has been modified in *n* days  
**-exec** *command*      true if exit status of *command* is 0  
**-ok** *command*      like **-exec** but asks  
**-print**      true; prints current pathname

combined with the following operators:

! prefix not  
-a infix and  
-o infix or  
( ) parentheses for grouping

**form** *proto* [*arg* ...] { where *proto* is the letter prototype in memory

**goto** *label* { used with : *label*

**grep** [ -v ] [ -b ] [ -c ] [ -n ] *expression* [ *file* ] ...

    -*v* = print all but those that match  
    -*b* = print block numbers  
    -*c* = print count of matching lines  
    -*n* = print line number

**gsi**

**If** *expression* *command* [ *arguments* ]

*expression* is made up of the following primitives:

    -*r* *file* true if *file* is readable  
    -*w* *file* true if *file* is writeable  
    *S1* = *S2* true if *S1* and *S2* are equal  
    *S1* != *S2* true if *S1* and *S2* are not equal  
    {*command*} true if exit status of *command* is 0

combined with the following operators:

! prefix not  
-a infix and  
-o infix or  
( ) parentheses for grouping

**kill** [ -*signo* ] *processid* ... { *signo* is sent with the following meanings

    1 = hangup  
    2 = interrupt  
    3 = quit  
    4 = illegal instruction  
    5 = trace trap  
    6 = IOT  
    7 = EMT  
    8 = floating exception  
    9 = kill  
    10 = bus error  
    11 = segment violation  
    12 = bad system call  
    13 = write on pipe with no one to read  
    14 = alarm clock

**ld** [ -*sulxrdni* ] *name* ...

<i>s</i> = strip
<i>u</i> = make following argument undefined
<i>lx</i> = load library /lib/lib <i>x.a</i> ; <i>x</i> is a string
<i>x</i> = do not save local symbols
<i>r</i> = generate relocation bits
<i>d</i> = define common storage
<i>n</i> = shared text
<i>i</i> = separate instruction and data space

**ln** *oldname* [ *newname* ] { *newname* is the name of the link

**login** [ *username* ]

**ls** [−ltasdrufg] *name* ...

**l** = long format  
**t** = sort by time modified  
**a** = list all entries  
**s** = give size in blocks  
**d** = list only directories names  
**r** = list in reverse order  
**u** = sort on last access time  
**i** = print i-number  
**f** = interpret all entries as directories  
**g** = give group ID instead of owner ID

**mb** [*file*]

**mail** [−yn] [*person* ...]

**y** = add to *mbox*  
**n** = throw away

**man** [*section*] [*title* ...]

**mesg** [ny]

**n** = forbid messages  
**y** = allow messages  
 no argument reverses current permission

**mkdир** *dirname* ...

**mv** [−f] *oldname newname* [−f = do not ask about file mode]

**neqn** [*file* ...]

**newgrp** *group*

**nice** [−number] *command* [*arguments*]

(number is a priority from 1 to 20, lowest 20, default 4.

**nm** [−cgnpru] [*file*]

**c** = list only C-style symbols  
**g** = print only global symbols  
**n** = sort by value instead of by name  
**p** = do not sort  
**r** = sort in reverse order  
**u** = print only undefined symbols

**nohup** *command* [*arguments*]

**nroff** [−o/] [−nn] [−ran] [−mx] [−s] [−h] [−q] *file* ...

**−o/** = list of pages to output, separated by , or −(range)  
**−nn** = number first generated page *n*  
**−ran** = set number register *a* to the value *n*  
**−mname** = prepend macro file /usr/lib/tmac.*name*  
**−s** = stop after each page— restart with nl  
**−h** = replace spaces with tabs  
**−q** = for insertions, send bell not name, do not echo

**od** [−abcdo] [*file*] [+] *offset*[.] [**b**]

**a** = op-codes  
**b** = bytes in octal  
**c** = bytes in ascii  
**d** = words in decimal  
**h** = words in hex  
**o** = words in octal  
*offset* = where to begin (octal; . for decimal; **b** for blocks)

**opr** [*destination*] [*-erm*] [*name ...*]

*destination* = 
$$\begin{cases} \mathbf{lp} = \text{local line printer} \\ \mathbf{mh} = \text{GCOS (default)} \\ \mathbf{sp} = \text{Spider network printer} \\ \mathbf{xx} = \text{station } xx \text{ at Murray Hill} \end{cases}$$

**r** = remove files

**c** = copy files

**m** = send mail when job transmitted

**passwd** *name password*

**pfe**

**pr** [*-h hdr*] [*-n*] [*+n*] [*-wn*] [*-ln*] [*-t*] [*-sc*] [*-m*] *name ...*

**-n** = *n*-column output

**+n** = begin with page *n*

**-h** = use next argument as header

**-wn** = use page width *n* (default 72 characters)

**-ln** = use page length *n* (default 66 lines)

**-t** = do not print header or trailer

**-sc** = separate columns by the character *c*

**-m** = print each file in a separate column

**prof** [*-v*] [*-k*] [*-a*] [*-l*] [*file*]

**-v** = output profile plot to 611 display

**-k** = output profile plot to Tektronix terminal

**-a** = report all symbols, not just externals

**-l** = order output by symbol value

**ps** [*aklx*] [*namelist*]

$$\begin{cases} \mathbf{a} = \text{give all processes with typewriters} \\ \mathbf{k} = \text{system debugging} \\ \mathbf{l} = \text{output long listing} \\ \mathbf{x} = \text{give all processes} \end{cases}$$

**ptx** [*-t*] *input [output]* · {*-t* produces phototypesetter output}

**pwd**

**rc** [*-c*] [*-f*] [*-v*] [*-r*] [*-U*] *file.r ... file.f ... [-l] ofile ...*

**-c** = suppress loading

**-f** = save Fortran intermediate files

**-v** = do not list intermediate file names while compiling

**-r** = Ratfor only

**-U** = flag use of undeclared variables

**-l** = loader options

**rev** [*file ...*]

**rm** [*-f*] [*-r*] *name ...*

**-f** = do not ask about mode

**-r** = remove directory contents recursively

**rmdir** *dir ...*

**roff** [*+n*] [*-n*] [*-s*] [*-h*] *file ...*

**+n** = start with page *n*

**-n** = stop after page *n*

**-s** = pause before each page

**-h** = use tabs for spaces

**sed** [**-g**] [**-n**] [**-f** *commandfile*] ... [**-e**] *command* ] ... [*file*] ...

**-g** = treat substitute commands globally

**-n** = only output lines operated on by **p** command

**-f** = next argument is command file

**-e** = next argument is editor command

**sh** [**-t**] [**-c** ...] [*name* [*arg1*] ... [*arg*] ] ]

**-t** = read standard input for one line

**-c** = use next argument as a command line

**shift**

**size** [*object-program* ...]

**sleep** *seconds*

**sno** [*file*]

**sort** [**-abdnrtx**] [**+m.n**] [**-m.n**] | ... [**-mo**] [*name*] ...

**a** = do not map lower case

**b** = ignore leading blanks

**d** = dictionary order

**n** = sort initial string by arithmetic value

**r** = reverse sort

**tx** = tab character is *x*

**-m** = merge only, files should be sorted

**-o** = next argument is output file

**+m.n** = skip *m* fields and *n* characters

**-m.n** = end of key (used with **+m.n**)

**spell** [**-v**] *file* ...    **{-v** = print all words not in the dictionary}

**split** [**-n**] [*file* [*name*]]    **{n** specifies the number of lines per file}

**strip** *name* ...

**stty** *option* ...    options, preceded by **-** to indicate negation are:

**even**    allow even parity

**odd**    allow odd parity

**raw**    raw mode input

**nl**    accept only new-line to end lines

**echo**    echo back every character typed

**lcase**    map upper case to lower case

**tabs**    preserve tabs

**ek**    reset erase and kill to # and @

**erase** *c*    set erase character to *c*

**kill** *c*    set kill character to *c*

**hup**    hang up on last close

**0**    hang up immediately

**crn**    set delay for carriage return *n*=(0,1,2,3)

**nl**    set delay for linefeed *n*=(0,1,2,3)

**tabn**    set delay for tab *n*=(0,1,2,3)

**ffn**    set delay for formfeed *n*=(0,1)

**tty33**    modes for Teletype model 33

**tty37**    modes for Teletype model 37

**vt05**    modes for DEC VT05

**tn300**    modes for GE Terminal 300

**ti700**    modes for Texas Instruments 700

**tek**    modes for Tektronix 4014

**50 75 110 134 150 200 300 600 1200 1800 2400**

**4800 9600 exta extb**    set baud rate

**sum** *file* ...

**tail** [**-n**] [*file*]

**tbl** [*file* ...]

**tee** [*name* ...]

**time** *command* [*arguments*]

**tmg** *name* {file for input is *name*, t

**tp** [*key*] [*name* ...] *key* is a character string containing at most one function and possibly several modifiers.

functions are: 
$$\begin{cases} r = \text{replace} \\ u = \text{update} \\ d = \text{delete} \\ x = \text{extract} \\ t = \text{list} \end{cases}$$

and modifiers are 
$$\begin{cases} m = \text{magnetic tape} \\ 0, \dots, 7 = \text{tape drive} \\ v = \text{verbose} \\ c = \text{create new tape} \\ f = \text{fake new entries} \\ i = \text{ignore errors} \\ w = \text{wait for user response} \end{cases}$$

**tr** [−cds] [*string1* [*string2*]]

**c** = complement characters in *string1*

**d** = delete all characters in *string1*

**s** = make repeated characters in *string2* one character

**troff** [−o/] [−sn] [−nn] [−ran] [−mx] [−t] [−f] [−w] [−a] *file* ...

**−o/** = list of pages to output, separated by , or −(range)

**−sn** = stop after every *n* pages

**−nn** = number first generated page *n*

**−ran** = set number register *a* to the value *n*

**−mx** = prepend file /usr/lib/tmac.x

**−t** = output to standard output

**−f** = do not feed paper or stop phototypesetter at end

**−w** = wait until phototypesetter available

**−a** = send printable approximation to standard output

**tty**

**typo** [−1] [−n] *file* ... 
$$\begin{cases} -1 = \text{single column output with no header} \\ -n = \text{do not read English statistics} \end{cases}$$

**uniq** [−ude] [+n] [−n] [*input* [*output*]]

**u** = output lines not repeated

**d** = output one copy of repeated lines

**c** = output count with each line

**+n** = skip first *n* fields in each line

**−n** = skip first *n* characters in each line

**wait**

**wc** [*name* ...]

**who** [*who-file*] [*am*]]

no arguments tells who is on

1 argument is file to be examined

2 arguments tells who you are

**write** *user* [*ttyno*]

**yacc** [**-vor**] [*grammar*]

**v** = make file *y.output*  
**o** = use optimizer  
**r** = for Ratfor instead of C

## ED

### Regular Expressions

#### strings of characters

\$	beginning of line
<i>re*</i>	end of line
[ <i>str</i> ]	all adjacent occurrences of <i>re</i>
[ ^ <i>str</i> ]	only those characters in <i>str</i>
null- <i>re</i>	any character except those in <i>str</i> and <nl>
	last <i>re</i> encountered

### Addresses

\$	current line
^	last line
'x	<i>n</i> -th line
/ <i>re</i> /	line marked with name <i>x</i>
? <i>re</i> ?	first line searching forward containing <i>re</i>
addr ± <i>n</i>	first line searching backward containing <i>re</i>
± <i>n</i>	addr plus (or minus) <i>n</i>
addr ±	'.' plus (or minus) <i>n</i> ( <i>n</i> =1 if not specified)
	addr plus (or minus) 1

### Commands

( . )a	append
( . . )c	change
( . . )d	delete
e [filename]	edit
f [filename]	remembered name
( . . \$ )g/re/commands	global
( . . )i	insert
( . . )kx	mark (addressed by 'x')
( . . )l	list
( . . )ma	move (after a)
( . . )p	print
q	quit
(\$ )r[filename]	read
( . . . )s/re/repl/	substitute
( . . . )s/re/repl/g	substitute globally
( . . . )ta	move copy (after a)
( . . \$ )v/re/commands	like global but lines that don't match
(1,\$ )w [filename]	write
(\$ )-	line number
!UNIX command	execute
<nl>	print next line

### EQN & NEQN

sub, sup  
over  
sqrt  
...from...to...  
left *c*, right *c*  
pile { ...above... }, lpile, cpile, rpile  
dot, dotdot, hat, bar, tilde, under  
size *n*, gsize *n*  
roman, italic, bold, font *f*, gfont *f*  
delim  
define, tdefine, undefine  
mark, lineup  
up, down, fwd, back  
matrix, lcol, ccol, rcol, col  
sum, int, integral, prod, union, inter  
>=, <=, !=, ==, +, -, >, <, approx  
sin, cos, tan, tanh, coth, sinh, cosh  
for, if

arc, times, llim, max, min, log, ln, exp  
prime, cdot, del, half

.....  
uppercase and lowercase greek  
infinity, inf, partial

## DB

/	word in octal
\	byte in octal
"	word in ascii
.	byte in ascii
?	word in decimal
&	instruction
<nl>	name of symbol
,	next word or byte
:	preceding word
%	exit
=	print ' ' or value of exp
:	symbolic address
!	store exp at " "
\$	general registers

## CDB

/o	octal
/i	decimal
/f	single-precision floating
/d	double-precision floating
\	byte in octal
<nl>	next word or byte
=	address
:	characters
"	characters pointer points to
&	symbol pointer points to
?	instruction
\$	stack
\$r	general registers
\$f	floating registers
%b	set breakpoint
%d	delete breakpoint
%r	run program
%c	continue after breakpoint

## FC RUNTIME DIAGNOSTICS

1	invalid log arg.
2	bad arg count to <b>amod</b>
3	bad arg count to <b>atan2</b>
4	excessive arg to <b>cabs</b>
5	exp too large in <b>cexp</b>
6	bad arg count to <b>cmplx</b>
7	bad arg count to <b>dim</b>
8	excessive arg to <b>exp</b>
9	bad arg count to <b>idim</b>
10	bad arg count to <b>isign</b>
11	bad arg count to <b>mod</b>
12	bad arg count to <b>sign</b>
13	illegal arg to <b>sqrt</b>
14	assigned/computed goto out of range
15	subscript out of range
16	real**real overflow
17	(negative real)**real
100	illegal I/O unit number
101	inconsistent use of I/O unit

102	cannot create output file
103	cannot open input file
104	EOF of input file
105	illegal character format
106	format does not begin with (
107	no conversion in format but non-empty list
108	excessive parenthesis depth in format
109	illegal format specification
110	illegal character in input field
111	end of format in hollerith specification
112	bad arg to <b>setfil</b>
120	bad arg to <b>rror</b>
999	unimplemented input conversion

## SH SPECIAL CHARACTERS

l	filter
:	filter
&	sequential command separator
<	return to user without waiting for command to finish
>	use next argument as standard input
>>	use next argument as standard output
?	same as > but append to file if it exists
*	match single character
[...]	match string of characters (including null)
\	match a class of characters
..	a pair of characters separated by a -
...	matches all characters alphabetically between the pair
\.	negates special meaning of following character
... " .."	take enclosed characters literally
... \$ ..	take enclosed characters literally
\$n	replace \$n with argument n (only for executing command file)
\$\$	replace \$\$ with process number
:	no-op command but may contain label

## TROFF & NROFF

\* - command causes a break (supressed by ', see c2)  
( ) - initial value (*troff,nroff*)  
[] - value if no argument  
F = R, I, B, S, G, C, P, etc.  
point size = 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 28, 36  
 $\pm N$  may be  $N$ ,  $+N$  (increment) or  $-N$  (decrement)  
 $-N$  may be  $N$  or  $-N$   
| $N$  is distance to place  $N$  from current place

### Scale Indicators

M	inches	$N^432$ internal units
Nc	cm.	$N^170$ internal units
Np	points	$N^6$ internal units
Nu	units	$N$ internal units
Nm	Ems	$N^6$ (point size) internal units
Nn	Ens	$N^3$ (point size) internal units
Nv	spaces	$N$ current line spacing (see .vs)

### Font and Character Size Control - troff only

.ps  $\pm N$  point size (10) [previous]  
.ss  $N$  space character size  $N/36$  Em (12/36 Em) [ignored]  
.cs  $F N M$  constant character spacing;  $N/36$  Em  $M/36$  Em  
 $N$  is width;  $M$  is size of Em; no  $N$  turns cs off;  
no  $M$  implies size dependent (off)  
.ft  $F$  font change (Roman) [previous]  
.fp  $N F$  font position, 1 to 4 (R,I,B,S) [ignored]  
.cp  $N$  constant output point-size (off) [off]

### Page Control

.pl  $\pm N$  page length (11 in, 66 lines) [11 in, 66 lines]  
.bp  $\pm N$  begin page;  $N$  is page number \* (1)  
.pn  $\pm N$  page number (1) [ignored]  
.po  $\pm N$  page offset (26/27 in,0) [previous]  
.ne  $N$  need  $N$  vertical space [1v, 1 line]  
.mk  $a$  mark vertical place in register  $a$  (none,0) [internal]  
.rt  $-N$  return to vertical place;  $N$  from top;  
- $N$  relative to current place;  
last marked place [internal]

### Text Filling, Adjusting, Centering, and Underlining

.br break \*  
.f fill \* (on)  
.nf no filling & adjusting \* (off)  
.ad  $c$  adjust mode;  $n$ (both adjusted), $r$ (right adjusted),  
 $c$ (centered) ( $n$ ) [ $n$ ]  
.na no adjusting (off)  
.mc  $c N$  specify margin character (off) [off]  
.ce  $N$  center  $N$  input lines \* (off) [1 line]  
.ul  $N$  nroff underline alphanumerics; troff italicize [1 line]  
.cu  $N$  nroff continuous underlining [1 line]

### Vertical Spacing

.vs  $N$  troff vertical spacing;  $v$  (12 points) [previous]  
.ls  $N$  nroff line spacing (1 line) [previous]  
.sp  $-N$  space vertically  $-N$  \* [1v,1 line]  
.sv  $N$  save vertical distance [1v,1 line]  
.os output saved vertical distance  
.ns no space mode (off)  
.rs restore spacing  
.xh nroff extra-half-line mode (off)

### Line Length and Indenting

.ll  $\pm N$  line length (6.5 in,65 lines) [previous]  
.in  $\pm N$  indent \* (0) [previous]

.ti  $\pm N$  temporary indent \* [ignored,1 space]

### Macros, Diversion, and Line Traps

.de xx define macro; end with ..; invoked by .xx [ignored]  
 .am xx append to macro [ignored]  
 .ds xx define string; invoked by \\*x or \\*(xx [ignored]  
 .as xx append to string [ignored]  
 .rm xx remove string or macro [ignored]  
 .di xx divert output to macro [end]  
 .da xx divert and append to xx [end]  
 .wh  $-N$  xx invoke xx when at or beyond place N;  
 -N means with respect to bottom of page  
 .ch  $-N -M$  change line trap from place N to M  
 .ch xx  $-M$  change line trap for xx to place M  
 .em xx end-macro name specification (none) [none]  
 .rn xx yy rename xx to yy  
 .dt N xx troff set diversion trap [off]  
 .lt N xx troff set input line count trap [off]

### Number Registers

.nr a  $\pm N -M$  number & increment register; called by \na or \n+a  
 .nr ab  $\pm N -M$  number register; called by \n(ab or \n(+ab  
 .nc c number character (\n) [\n]  
 .af xx c assign format to xx  
 1(0,1,...),i(0,i,ii,...),I(0,I,II,...),a(0,a,b,...),  
 A(0,A,B,...) [remove]

### Tabs, Leaders, and Fields

.ta N... tab settings ((.5,1,...in),(9,17,...)) [none]  
 .tc c tab replacement character (none,space) [none,space]  
 .lc c leader replacement character (.) [.]  
 .fc a b set field delimiter & pad character (off) [off]

### Input and Output Conversions & Character Translations

.ec c set escape character (\) N  
 .eo turn off escape processing  
 .lg N troff ligature mode; 0=off (on) [on]  
 .cc c basic control character (.) [.]  
 .c2 c nobreak control character (') [']  
 .li N accept input lines literally [1 line]  
 .tr abcd... translate on output; a to b,... (none)

### Hyphenation

.nh no hyphenation (off)  
 .hy N hyphenation; N=0(off),  $\neq 0$ (on), -2(don't hyphenate  
 last lines), -4(don't split off last 2 characters),  
 -8(don't split off first 2 characters) (1)  
 .hc c hyphenation indicator character (none) [none]  
 .hw word1 ... hyphenation exception list

### Three Part Titles

.tl 'l'c'r' title  
 .lt N length of title (6.5 in, 65 char) [previous]

### Output Line Numbering

.nm  $\pm N M S I$  number mode on or off, set parameters [off]  
 .np M S I number parameters set or reset; M(every Mth line)[1]  
 S(separation)[2v], I(number indent)[0]; (none) [reset]

## Conditional Input Line Acceptance

.if *c anything* if *c* true accept *anything*, *c*=e (even page number),  
.if !*c anything* o (odd page number) t (troff), n (nroff) !=not  
.if *N anything* if *N* > 0 accept *anything*, *N* is a number register  
.if !*N anything* !=not

## Environment Switching

.ev *N* environment pushed down (0) [previous]

## Insertions from Standard Input

.rd *prompt* read insert [bell]  
.ex exit

## Input File Switching

.so *filename* switch source file-push down  
.nx *filename* next file [EOF on current file]

## Miscellaneous

.tm *string* typewriter message  
.hs *N* troff control high-speed multiple scan (on) [on]  
.lg ignore until ..  
.fl flush output buffer \*  
.ab abort  
.pi *prog* nroff pipe output to *prog*

## ESCAPE SEQUENCES (Characters, Indicators, and Functions)

\* = troff only

\*\* = nroff only

\\	prevent interpretation of \
\\e	print current escape character
\\.	*
\\-	-
\\_	+
\\	1/6 Em space character *
\\(space)	unpaddable space
\\&	zero width character
\\:	transparent line indicator
\\?	raw transmission indicator **
\\\$	argument indicator
\\(	character name indicator *
\\+	string indicator
\\:	ASCII ETX
\\a	leader character
\\b	bracket building function *
\\c	interrupt text processing
\\d	forward (down) 1/2 Em (troff) or line (nroff)
\\f	font change function
\\h	local horizontal motion function *
\\k	mark horizontal place
\\l	draw horizontal line
\\n	number register indicator
\\o	overstrike function *
\\p	break and spread output line
\\r	reverse 1Em or line vertical motion
\\s	point size change function
\\t	non-interpreted horizontal tab
\\u	reverse (up) 1/2 Em or line
\\v	local vertical motion function *
\\w	width function
\\x	extra line-space function *
\\x	ASCII shift-out **
\\y	ASCII shift-in
\\z	zero width character function *

\(newline) concealed newline \*  
 \L draw vertical line \*  
 \0 horizontal motion equal to width of a number

## RESERVED REGISTERS

(r) = read only  
 \* = troff only

%	current page number
.\$	number of arguments available at macro level (r)
.A	1 if troff —a called, 0 otherwise (r) *
.a	most recent post-line application of \x (r) *
.c	input line count in current file (r)
.f	physical quadrant of current font (r) *
.h	high-water mark of nl for text on current page (r)
.i	current indent (r)
.l	current line length (r)
.n	length of text on last output line (r)
.o	current page offset (r)
.p	current page length (r)
.s	current point size (r) *
.t	distance to next trap (r)
.v	current vertical line spacing (r)
.x	reserved version-dependent register (r)
ct	character type *
dn	vertical size of last diversion
dw	current day of the week number
dy	current day of the month
hp	current horizontal place on input line
nl	current vertical place on the page
no	current month number
yr	last 2 digits of current year

## -mr OPTION FOR NROFF SIMULATES ROFF

\* — command causes a break  
 () — initial value

### Page Control

.bl n insert blank lines on new page \*

### Vertical Spacing

.ss single space \* (on)  
 .ds double space \*

### Titles

.he 'l'c'r	head title
.eh 'l'c'r	even head title
.oh 'l'c'r	odd head title
.fo 'l'c'r	foot title
.ef 'l'c'r	even foot title
.of 'l'c'r	odd foot title
.m1 n	n blank lines bet. top of page and head title (2)
.m2 n	n blank lines bet. head title and text (2)
.m3 n	n blank lines bet. text and foot title (1)
.m4 n	n blank lines bet. foot title and bottom of page (3)

### Line Numbers

.n1 line number from 1 on each page; add 5 to page offset  
 .n2 n line number from n; add 5 to page offset

## TROFF SPECIAL CHARACTERS

,	\` close	½	\(34	□	\(sb
,	\` open	—	\(mi	□	\(sp
—	\`- dash	fi	\(fi	□	\(ib
—	\`- hyphen	fl	\(fl	□	\(ip
—	\`(em	ff	\(ff	□	\(if
.	\`(hy	—	\(en	□	\(pd
•	\`(bu	¶	\(pg	□	\(gr
—	\`(ru	°	\(de	□	\(no
½	\`(14	†	\(dg	□	\(is
½	\`(12	§	\(sc	□	\(pt
			\(fm		\(es
,	\`(aa	Θ	\`(*H	□	\(co
,	\`(ga	I	\`(*I	□	\(ct
,	\`(ul	K	\`(*K	□	\(br
—	\`(sl	Λ	\`(*L	□	\(ci
α	\`(*a	M	\`(*M	□	\(it
β	\`(*b	N	\`(*N	□	\(rk
γ	\`(*g	Ξ	\`(*C	□	\(bv
δ	\`(*d	O	\`(*O	□	\(if
ε	\`(*e	Π	\`(*P	□	\(rf
ζ	\`(*z	P	\`(*R	□	\(lc
η	\`(*y	Σ	\`(*S	□	\(rc
θ	\`(*h	T	\`(*T	□	\(fm
ι	\`(*i	Υ	\`(*U	□	
κ	\`(*k	Φ	\`(*F	□	
λ	\`(*l	X	\`(*X	□	
μ	\`(*m	Ψ	\`(*Q	□	
ν	\`(*n	Ω	\`(*W	□	
ξ	\`(*c	√	\`(*sr	□	
ο	\`(*o		\`(*rn	□	
π	\`(*p	≥	\`(>=	□	
ρ	\`(*r	≤	\`(<=	□	
σ	\`(*s	≡	\`(==	□	
ς	\`(*s	≈	\`(=	□	
τ	\`(*t	≈	\`(=	□	
υ	\`(*u	~	\`(*ap	□	
φ	\`(*f	~	\`(*!=	□	
χ	\`(*x	→	\`(>	□	
ψ	\`(*q	↑	\`(<-	□	
ω	\`(*w	↑	\`(*ua	□	
Α	\`(*A	↓	\`(*da	□	
Β	\`(*B	↓	\`(*tf	□	
Γ	\`(*G	↓	\`(*mu	□	
Δ	\`(*D	×	\`(*di	□	
Ε	\`(*E	÷	\`(*+-	□	
Ζ	\`(*Z	±	\`(*cu	□	
Η	\`(*Y	□	\`(*ca	□	

## -ms OPTION FOR TROFF &amp; NROFF

\* - command causes a break  
 () - initial value

## Format &amp; Abstract

.TM *x y z* TM cover sheet *tm# case# file#*  
 .RP released paper cover sheet  
 .TL title follows \*  
 .AU *ad ex* authors names follow; address extension \*  
 .AI authors institute follows (.MH, .HO, .WH) \*  
 .AB begin abstract \*  
 .AE end of abstract \*  
 .CS *data* cover sheet data \*  
 #text #other total #fig #tab #ref  
 .OK other keywords follow \*  
 .SG signature line follows - for TM's \*

## Headings &amp; Paragraphs

.PP paragraph \*  
 .IP *x y* indented paragraph; hanging tag *x, y* ens indentation \*  
 .LP block paragraph (on) \*  
 .NH *n* numbered headings in bold, *n*=level \*  
 .SH bold headings, no numbers \*

## Fonts &amp; Sizes

.B bold  
 .I italic  
 .R roman (on)  
 .LG larger  
 .SM smaller  
 .NL normal (on)

## Footnotes

.FS start footnote  
 .FE end footnote

## Displays &amp; Tables

.DS *x* begin display; *x*=C(center), L(left adjust), I(indent) \*  
 .DE end display \*  
 .CD long centered display \*  
 .LD long left adjusted display \*  
 .ID long indented display \*  
 .KS begin keep \*  
 .KE release keep \*  
 .KF keep floating \*

## Multicolumn

.2C 2 column \*  
 .1C 1 column (on) \*

## Multiple indenting

.RS increment level of indent \*  
 .RE decrement level of indent \*

## Date

.DA current date - on for nroff  
 .ND no date - on for troff

## Equations

.EQ *x n* begin equation; for *x* see DS, *n*=equation number \*  
 .EN end equation \*

## C SYSTEM CALLS

<b>access(file, mode)</b>	check mode access of file
char *file;	
<b>alarm(n)</b>	receive alarm signal in <i>n</i> seconds
char *brk(addr)	set lowest location to addr
char *sbrk(incr)	add to data space
<b>chdir(dirname)</b>	change working directory
char *dirname;	
<b>chmod(name, mode)</b>	change mode of file
char *name;	(for mode see <b>chmod</b> command)
<b>close(fd)</b>	
<b>creat(name, mode)</b>	create a new file
char *name;	(for mode see <b>chmod</b> command)
<b>getcswo()</b>	read console switches
<b>dup(fd)</b>	duplicate an open file descriptor
<b>exec(name, arg0, arg1, ..., argn, 0)</b>	execute a file
char *name, *arg0, *arg1, ..., *argn;	
<b>execv(name, argv)</b>	execute a file
char *name, *argv[1];	
<b>exit(status)</b>	terminate process
<b>fork()</b>	spawn new process
<b>fstat(fd, buf)</b>	get status of open file
struct inode *buf	see <b>stat</b> for format of <i>buf</i>
<b>getgid()</b>	get group id; low byte is real, high byte is effective
<b>getpid()</b>	get process id
<b>getuid()</b>	get user id; low byte is real, high byte is effective
<b>gtty(fd, arg)</b>	get typewriter status
int arg[3];	see <b>stty</b> for format of <i>arg</i>
<b>kill(pid, sig)</b>	send signal to process; see <b>kill</b> command
<b>link(olddname, newname)</b>	link to a file
char *olddname, *newname;	
<b>mknode(name, mode, addr)</b>	make a directory or special file
char *name;	
<b>mount(special, name, rwflag)</b>	mount file system
char *special, *name;	
<b>nice(priority)</b>	set program priority; -220 to 20, 20 lowest
<b>open(name, mode)</b>	open for reading(0) writing(1), or both(2)
<b>pause()</b>	wait indefinitely (or for alarm)
<b>pipe(fd)</b>	create an interprocess channel
int fd[2];	read[0], write[1]
<b>profile(buff, bufsiz, offset, scale)</b>	execute time profile
char buff[1];	
int bufsiz, offset, scale;	
<b>ptrace(request, pid, addr, data)</b>	process trace
<b>read(fd, buffer, nbytes)</b>	read from file
char *buffer;	

<b>seek(fd, offset, ptrname)</b>	move read/write pointer
<b>setgid(gid)</b>	set process group id
<b>setuid(uid)</b>	set process user id
<b>signal(sig, func)</b> int (*func)();	catch or ignore signals; see <b>kill</b> func=0 is default, func=1 is ignore
<b>sleep(seconds)</b>	stop execution for interval
<b>stat(name, buf)</b> char *name; struct inode *buf; struct { char minor; char major; int inumber; int flags; char nlinks; char uid; char gid; char size0; int size1; int addrl8i; int actime[2]; int modtime[2]; };	get file status minor device of i-node major device of i-node high-order low-order block numbers or device number top 4 bits of <i>flags</i> 10 node is allocated 00 plain file 04 directory 02 character-type special file 06 block-type special file 01 large file for last 12 bits see <b>chmod</b> command
<b>stime(tbuf)</b> set time int tbuf[2];	
<b>stty(fd, arg)</b>	set typewriter modes struct { char ispeed, ospeed; char erase, kill; int mode; } *arg;
<b>time(tvec)</b> get date and time since 00:00:00 GMT, Jan. 1, 1970 int tvec[2];	
<b>times(buffer)</b>	get process times struct tbuffer *buffer; struct tbuffer { int proc_user_time; int proc_system_time; int child_user_time[2]; int child_system_time[2]; };
<b>unlink(name)</b>	remove directory entry char *name;
<b>wait(status)</b>	wait for process to terminate int *status;
<b>write(fd, buffer, nbytes)</b>	write on a file char *buffer;

## SITE DEPENDENT COMMANDS

Center 127

cb

factor [number]

fget [−m] [u GCOScat] [d UNIXdir] [file...]

−m = report snumb by mail

−u = use next argument as GCOS catalog name to get files

−d = use next argument as UNIX directory to put files

fsend [−c] [−r] [−m] [f GCOSfile] [u GCOScat] [−a] [−b]

−c = copy files to be sent before returning

−r = remove files after sending them

−m = report snumb by mail

−f = use next argument as GCOS file for succeeding file

−u = use next argument as GCOS catalog for succeeding files

−a = send files in ASCII (default)

−b = send files in binary

ibm [−J] file... {− suppresses all JCL}

iget [−unix] [−his] source destination

−unix = send file to UNIX (default)

−his = send file to GCOS

lsend [bn] unix ibm [+] [bcd]

−bn = fixed length records of n characters (default 80)

+ = file does not exist on IBM 370, create it

deb = use DD statement in file

nfs −key [name...] key is a character string containing at most one function and possibly several modifiers.

functions are:	w = write
	u = update
	d = delete
	r = read
	t = list
	l = long list
	s = report space allocation
	m = make a directory
modifiers are:	c = current directory is in first name argument
	v = verbose
	j = wait for user response
	x = interpret directory names explicitly

teatsim [−pnl] [file] {−n is page length in inches}

tekstare [−s] [−m] [−v] [files]

−s = STARE output (default)

−m = microfilm output

−v = viewgraph output

tss

commands to the interface routine are:

‐<file use UNIX file as input

‐>file send TSS output to UNIX file

‐p pop output file

‐q disconnect from TSS

‐‐file receive file from HIS routine csr/daccopy

‐‐‐file send file to HIS routine csr/daccopy

to send files to TSS run: csr/daccopy (s) a/fname

with ‐s above

to receive files from TSS run: csr/daccopy (r) a/fname

with ‐r above

## SOURCES

1. K. Thompson, D. M. Ritchie, *UNIX Programmer's Manual*, Sixth Edition (May 1975).
2. *Documents for use with the UNIX Time-Sharing System*, Sixth Edition.
3. J. F. Ossanna, *TROFF User's Manual*, [REDACTED] (April 19, 1974).