T   LOGIN•S

B

L   O

•   G

G   I

R   N

E   •

PS•TOUCH   S

T   CAT•Y
PR

• O   A

W   F   C

R   F   C

I   •

T   DC

E

CMP

A

K

E

•

S

E

ADB

F   E

I   T

N   A

DD

LN

I   E

A   Q

T   N

™

UNIX† Reference Card

$\triangledown$

distributed by
Computing Information Service
BELL LABORATORIES
Murray Hill, N. J. 07974

compiled by
Lorinda Cherry

Second Edition

March, 1979

TABLE OF CONTENTS

1a. GENERAL UNIX COMMANDS

1

1b. UNIX COMMUNICATIONS

1c. UNIX MAINTENANCE COMMANDS

2. COMMAND DETAILS

3. DOCUMENT PREPARATION

4. C FUNCTIONS

adb [−w] [*objfil* [*corfil*]]   { −w = open *objfil* and *corfil*
     for both reading and writing; defaults: **a.out core**

ar
$\begin{Bmatrix} d \\ m \\ p \\ q \\ r \\ t \\ x \end{Bmatrix}$
[**abciluv**][*posname*] afile file ...

     d = delete   a = after *posname*
     m = move to end or *posname*   b or i = before *posname*
     p = print   c = suppress create message
     q = quickly (no checking)   l = local temp files
     r = replace   u = update
     t = list   v = verbose
     x = extract

as [−] [−o *objfil*] *file* ...
     − = make undefined symbols global
     −o = use next argument as output; default: **a.out**

at *time* [*day*] [*file*]
     time = hours [*minutes*] [**apnm**] (AM, PM, noon, midnight)
     day = month day-no or *day-of-week* [**week**]
     week = 7 days more

awk [−F*c*] [−f *progfile*] [*prog*] [*file*] ...
     −F*c* = use *c* as field separator
     −f = use next argument as *prog*

bas [*file*]

basename *string* [*suffix*]

bc [−l] [−c] [*file* ...]
     −l = load the function library
     −c = compile only

calendar [−]   { − sends all users their calendar entries

cat [−u] *file* ...   { −u causes output to be unbuffered

cc [−c] [−p] [−f] [−O] [−S] [−P] [−o *output*]
     [−D *name*=*def*] [−U *name*] [−I *dir*] *file*.c... [−l] ofile...
     −c = suppress loading
     −p = profile
     −f = floating-point interpreter
     −O = optimize
     −S = keep assembler code: output to *file*.s
     −P = just preprocess: output to *file*.i
     −o *output* = output to *output*; default: **a.out**
     −D *name*=*def* = define preprocessor variable
     −D *name* = *name* is set to 1
     −U *name* = undefine *name*
     −I *dir* = after home, look in *dir* for include files
     −l = loader options

cd *directory*

chgrp *group file* ...

chmod *mode file* ...   [*mode* may be symbolic [**ugoal**][+ − =][**rwxst**]
     where a = ugo, s = set id and t = sticky
     or *mode* may be the octal OR of the following:
     4000 set user id on execution

5

echo [−n] [*args*]   { −n adds no newline to output

ed [−] [−x] [*name*]
        − = no character count on e, r, w commands
        −x = file is encrypted

eqn [−d*xy*] [−f*n*] [−p*n*] [−s*n*] [*file*] ...
        −d*xy* = use *x* and *y* as delimiters
        −f*n* = use *n* as font
        −p*n* = use ±*n* for subscript point size changes; default: 3
        −s*n* = use *n* as point size

expr *arg* ...

f77 [−c] [−p] [−O] [−S] [−f] [−o *output*] [−onetrip] [−u] [−C]
   [−w] [−F] [−m] [−E*x*] [−R*x*] *file.f* ... [−*l*] *ofile* ...
        −c = suppress loading; output in *file.o*
        −p = profile
        −O = optimize
        −S = compile only; output to *file.s*
        −f = floating point interpreter
        −o *output* = output to *output* ; default: **a.out**
        −onetrip = compile onetime loops
        −u = default variable type undefined
        −C = subscript checking
        −w = no warning messages
        −w66 = no Fortran 66 warning messages
        −F = run only EFL and Ratfor preprocessors
        −m = run M4 preprocessor before EFL or Ratfor
        −E*x* = use *x* as EFL option
        −R*x* = use *x* as Ratfor option
        −*l* = loader options

factor [*number*]

file *file* ...

find *pathname expression*
        expression is made of the following primitives.
        n is integer , +*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
        −**inum** *n*   true if file has inode *n*
        −**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
        −**newer** *file*   true if file was modified before *file*

        combined with the following operators:
        !   prefix *not*
        −a   infix *and*
        −o   infix *or*
        ( )   parentheses for grouping; must be escaped with \

graph [−a] [−b] [−c *c*] [−g *n*] [−l *lab*] [−m *n*]
   [−h *f*] [−w *f*] [−r *f*] [−u *f*] [−s] [−t]
   [−x[l] *low* [*upper* [*sp*]]] [−y[l] *low* [*upper* [*sp*]]]
        −a = automatic abscissas
        −b = disconnect after each label
        −c *c* = use *c* as label for each point
        −g *n* = *n* is grid type; 0 (no grid), 1(frame + ticks), 2(full)

```
              -l lab = label graph with lab
              -m n = mode; 0(disconnected), 1(connected); default: 1
              -s = save screen; don't erase before plotting
              -h f = f is fraction of space for height
              -w f = f is fraction of space for width
              -r f = f is fraction of space to move right before plotting
              -u f = f is fraction of space to move up before plotting
              -t = transpose horizontal and vertical axes
              -x[l] = next 3 arguments are lower, upper, & spacing limits
                      for x; l for logarithmic x axis
              -y[l] = next 3 arguments are lower, upper, & spacing limits
                      for y; l for logarithmic y axis

rep [-v] [-b] [-c] [-n] [-l] [-s] [-h] [-y] [-e expr] expr [file]
   egrep [grep-option] [-f file] [expr] [file] ...
   fgrep [grep-option] [-x] [-f file] [strings] [file]
              -v = print all but those that match
              -b = print block numbers
              -c = print count of matching lines
              -n = print line number
              -l = list files with matching lines
              -s = just return status
              -h = no filename headers on output lines
              -y = fold lower and upper case (grep)
              -x = output only lines matching totally (fgrep)
              -e expr = use expr for matching
              -f file = use file for matching (egrep,fgrep)

oin [-an] [-e s] [-jn m] [-j m] [-o m.n ...] [-tc] file1 file2
              -an = output unpaired lines in file n also
              -e s = use string s for white space
              -jn m = join on m th field of file n
              -j m = join on m th field
              -o n.m ... = output m th field of n th file ...
              -tc = use c as input separator

kill [-signo] processid ...   (signo is sent with the following meanings:
              name following description is in #include <signal.h>)
              1 = hangup (SIGHUP)
              2 = interrupt (SIGINT)
              3 = quit (SIGQUIT)
              4 = illegal instruction (SIGILL)
              5 = trace trap (SIGTRAP)
              6 = IOT (SIGIOT)
              7 = EMT (SIGEMT)
              8 = floating exception (SIGFPE)
              9 = kill (SIGKILL)
              10 = bus error (SIGBUS)
              11 = segment violation (SIGSEGV)
              12 = bad system call (SIGSYS)
              13 = write on pipe with no one to read (SIGPIPE)
              14 = alarm clock (SIGALRM)
              15 = software terminate (SIGTERM) (default)
              16 = unassigned

ld [-sulxXrdnioeOD] name ...
              s = strip
              u = make following argument undefined
              lx = load library /lib/libx.a; x is a string
              x = do not save local symbols
              X = used by cc
              r = generate relocation bits
              d = define common storage
              n = share text
              i = separate instruction and data space
              o = use next argument as output not a.out
```

```
                    e = use next argument as entry point; default: 0
                    O = overlay file
                    D = use next argument as data segment size
```

learn [− *directory*] [*subject* [*lesson* [*speed*]]]

lex [−**fntv**] [*file*]
```
        −f = fast; no packing; only for small programs
        −n = no summary statistics; default
        −t = output to stdout not lex.yy.c
        −v = generate summary statistics
```

lint [−**abchnpuvx**] *file* ...
```
        a = report longs assigned to int variables
        b = report break statements not reached
        c = report unportable casts
        h = use heuristic tests
        n = don't check for standard library
        p = check portability to IBM and GCOS C
        u = don't report on defined and unused variables
        v = don't report on unused arguments in functions
        x = report externally declared variables not used
```

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

login [*username*]

look [−**df**] *string* [*file*]
```
        d = dictionary order
        f = fold upper case letters to lower case
        default input: /usr/dict/words folded
```

lorder *file* ...

ls [−**ltasdrucifg**] *name* ...
```
        l = long format
        t = sort by time modified
        a = list all entries
        s = give size in blocks
        d = list only directory names
        r = list in reverse order
        u = sort on last access time
        c = sort on last inode change
        i = print i-number
        f = interpret all entries as directories
        g = give group ID instead of owner ID
```

m4 [*file*]

mail *person* ...

mail [−**p**] [−**r**] [−**f** *file*]
```
        −r = first-in, first-out order
        −p = don't ask questions
        −f file = use file as mail file
```

make [−**f** *makefile*] [−**ikntrs**] *file* ...
```
        i = ignore returning status
        k = on bad status, continue unrelated entries
        n = trace and print, don't execute
        t = touch
        r = no suffix list
        s = work silently
```

man [−**tnkew**] [*chapter*] *title* ...
```
        t = phototypeset
        n = output to stdout (default)
        k = output to Tektronix 4014
        e = run eqn; append or prefix e with above
```

9

        w = print only path names

mesg [ny]
        n = forbid messages
        y = allow messages
        no argument: report current permission

mkdir *dirname* ...

mv *oldname newname*

mv *file ... directory*

newgrp *group*

nice [− *number*] *command* [*arguments*]
        [*number* is a priority from 1 to 20, lowest 20; default: 10.

nm [− **gnopru**] [*file*]
        g = print only global symbols
        n = sort by value instead of by name
        o = output file name on each line
        p = do not sort
        r = sort in reverse order
        u = print only undefined symbols

nohup *command* [*arguments*]

nroff [− **o**/] [− **n***n*] [− **r***an*] [− **m***name*] [− **s***n*] [− **h**] [− **i**] [− **q**]
    [− **T** *name*] [− **e**] *file* ...
         − **o**/ = list of pages to output, separated by , or − (range)
         − **n***n* = number first generated page *n*
         − **r***an* = set number register *a* to the value *n*
         − **m***name* = prepend macro file **/usr/lib/tmac/tmac.** *name*
         − **s***n* = stop after *n* pages; default: 1
         − **h** = replace spaces with tabs
         − **i** = read **stdin** after files
         − **q** = for insertions, send bell not name, do not echo
         − **T** *name* = output for terminal *name*
                37 tn300 300s 300 450
         − **e** = equally-spaced words in adjusted lines

od [− **bcdox**] [*file*] [[**+**] *offset* [.] [**b**]]
        b = bytes in octal
        c = bytes in ascii
        d = words in decimal
        o = words in octal
        x = words in hex
        offset = where to begin (octal: . for decimal: **b** for blocks)

passwd [*name*]

plot [− **T***term* [*raster*]]
        term = 4014 450 300 300S ver

pr [− *n*] [**+** *n*] [− **h** *hdr*] [− **w***n*] [− **l***n*] [− **t**] [− **s***c*] [− **m**] *name* ...
         − *n* = *n*-column output
         + *n* = begin with page *n*
         − **h** = use next argument as header
         − **w***n* = use page width *n*; default: 72
         − **l***n* = use page length *n*; default: 66

```
                    -t = do not print header or trailer
                    -sc = separate columns by the character c
                    -m = print each file in a separate column

prof [-v] [-a] [-l] [ -low [ -high ] ] [file]
                    -v = output only graphic profile
                    -a = report all symbols, not just externals
                    -l = order output by symbol value
                    low, high = percentages for plotting; default: 0,100

ps [aklx] [namelist]
                    a = give all processes with typewriters
                    k = system debugging
                    l = output long listing
                    x = give all processes

ptx [-f] [-w n] [-g n] [-o only] [-i ignore] [-b break] [-r]
    [-t] [input [output] ]
                    -f = fold upper and lower case
                    -w n = set line width to n; default: 72
                    -g n = set column gap to n; default: 3
                    -o only = use only keywords from only
                    -i ignore = don't use keywords from ignore
                    -b break = use characters in break to separate words
                    -r = use 1st word as reference; use as 5th field
                    -t = phototypeset

ratfor [-C] [-h] [-6x] [file] ...
                    -C = copy comments
                    -h = convert quoted strings to H convention
                    -6x = use x as continuation character

refer [-ar] [-b] [-cs] [-e] [-kx] [-lm,n] [-p f] [-n] [-ske
    lookbib [file] ...
                    -ar = reverse the first r author names
                    -a = reverse all author names
                    -b = omit all flags
                    -cs = capitalize fields in s
                    -e = accumulate references
                    -kx = labels specified with x
                    -lm,n = label with m name letters and n year digits
                    -p f = use file f as references
                    -n = don't search default file
                    -skeys = sort on keys

rev [file ... ]

rm [-fir] file ...
                    -f = do not ask about mode
                    -i = interactive
                    -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 [-n] [-f commandfile] [-e script] [file] ...
                    -n = only output lines operated on by p command
                    -f = next argument is command file
                    -e = next argument is editor command
```

```
sh  [−ceiknrstuvx] [arg] ...
            c = use next argument as command
            e = if non interactive, exit on bad status
            i = interactive
            k = keywords
            n = don't execute, just read commands
            r = restricted environment
            s = read from stdin
            t = exit after 1 command
            u = treat unset variables as error
            v = print input as it's read
            x = print on execution

size  [object ... ]

sleep  seconds

sort  [−cmubdfinrt x] [+m.n [ −m.n ] ] ... [−o name ] [−T dir ] [file ]
            c = just check order, don't sort
            m = merge only, files should be sorted
            u = output only 1 copy of equal lines
            b = ignore leading blanks
            d = dictionary order
            f = fold upper case to lower case
            i = ignore characters outside 040-0176
            n = sort initial string by arithmetic value
            r = reverse sort
            tx = tab character is x
            +m.n = skip m fields and n characters
            −m.n = end of key (used with +m.n)
            −o name = output to file
            −T dir = use dir for temporaries

spell  [−b] [−v] [−x] [file ... ]
            b = British
            v = output derived words from list
            x = print stems

spline  [−a] [−k] [−n] [−p] [−x]
            −a = automatic abscissas; next argument is spacing
            −k = next argument is used to compute boundary value
            −n = next argument is number of intervals between limits
            −p = make output periodic
            −x = next 1 or 2 arguments are lower and upper x limits

split  [−n] [file [name] ]     [n lines per file; default: 1000

strip  name ...

struct  [−s] [−i] [−a] [−b] [−n] [−en ] [file ]
            −s = input in standard format
            −i = don't make computed goto's switches
            −a = turn else if sequences into switch
            −b = generate goto's, not breaks
            −n = generate goto's, not nexts
            −en = loop control parameter

stty  option ...     [options, preceded by − to indicate negation are:
            even        allow even parity
            odd         allow odd parity
            raw         raw mode input
            cbreak      pass characters on as received
            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
        cr n        set delay for carriage return n=(0,1,2,3)
        nl n        set delay for linefeed n=(0,1,2,3)
        tab n       set delay for tab n=(0,1,2,3)
        ff n        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 TermiNet 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
```

su [userid]

sum file ...

tabs [−n] [terminal]
        −n = don't indent left margin
        terminal =
                1620[−12]
                300[s][−12]
                33, 37, 40−2, 43
                450[−12][−8]
                735, 745
                dumb
                hp
                4014
                tn1200, tn300
                vt05

tail [± number [lbc]] [file]   number = lines, blocks, characters
        from beginning (+) or end (−)

tar [key] [name ...]   key is a string containing at most
        one function and optionally several modifiers.
        functions are:
                r = write files on end of tape
                x = extract from tape
                t = list
                u = update
                c = create new tape (implies r)
        modifiers are:
                0,...,7 = tape drive
                v = verbose
                w = wait for user response
                f = use next argument as tape
                b = use next argument as blocking factor
                l = complain on unresolved links

tbl [file ...]

tc [−t] [−sn] [−pl] [file]
        −t = don't wait between pages
        −sn = skip the first n pages
        −pl = set page length to l [picP]
                (points, inches, centimeters, Picas): default: P

tee [−i] [−a] [file] ...
        −i = ignore interrupts
        −a = append don't overwrite

test *expr* {*expr* is made of the following primitives:
  −r *file* = true if *file* is readable
  −w *file* = true if *file* is writable
  −f *file* = true if *file* is not a directory
  −d *file* = true if *file* is a directory
  −s *file* = true if *file* longer than 0
  −t [*fildes*] = true if *fildes* is a terminal; default: 1
  −z *s1* = true if length of *s1* = 0
  −n *s1* = true if length of *s1* not 0
  s1 = *s2* = true if *s1* = *s2*
  s1 != *s2* = true if *s1* != *s2*
  s1 = true if *s1* not null
  n1 op n2 = algebraic comparison of integers
    op = **−eq**, **−ne**, **−gt**, **−ge**, **−lt**, **−le**
  combined with the following operators:
    ! = unary negation
    −a = binary *and*
    −o = binary *or*
    (*expr*) = parentheses for grouping

time *command* [*arguments*]

tk [**−t**] [**−n**] [**−p***l*] [*file*]
  −t = don't wait between pages
  −n = *n* column output
  −p*l* = set page length to *l*

touch [**−c**] *file* ... {**−c** = create *file*

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

  functions are:
    r = replace
    u = update
    d = delete
    x = extract
    t = list
  and modifiers are:
    m = magtape
    0,....7 = tape drive
    v = verbose
    c = create new tape
    f = use first file in place of tape
    i = ignore errors
    w = wait for user response

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***l*] [**−s***n*] [**−n***n*] [**−r***an*] [**−m***x*] [**−t**] [**−f**] [**−w**] [**−a**] [**−i**] [**−q**] [**−b**] [**−p***n*] [**−g**] *file* ...
  −o*l* = list of pages to output, separated by , or − (range)
  −s*n* = stop after every *n* pages
  −n*n* = number first generated page *n*
  −r*an* = set number register *a* to the value *n*
  −m*x* = prepend file **/usr/lib/tmac/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
  −i = read **stdin** after files
  −q = do **rd** in simultaneous input-output mode
  −b = just report if phototypesetter busy

```
              −pn = print all characters in size n
              −g = output for GCOS

tsort [file]

uniq [−udc] [+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

wc [−lwc] [name ...]        { −lwc = just count lines, words or chars

who [who-file] [am i]
              no arguments tells who is on
              1 argument is file to be examined
              2 arguments tells who you are

write user [ttyname]

xsend person

yacc [−vd] [grammar]
              v = make file y.output
              d = make file y.tab.h for define statements
```

COMMUNICATIONS COMMANDS

```
cu telno [−t] [−s speed] [−a acu] [−l line]
              in telno use − for delays
              −t = to dial a terminal
              −s = speed = 110 134 150 300 1200; default 300
              −a = acu pathname; default /dev/cua0
              −l = line = line name; default /dev/cul0

              The following are interpreted by the send process:
              ~.           terminate connection
              ~EOT         terminate connection
              ~<file       send contents of file as though typed
              ~!           run shell on local system
              ~!cmd ...    run cmd on local system
              ~$cmd ...    run cmd on local system and send output
              ~%take from [to]    copy file from remote to local
              ~%put from [to]     copy file from local to remote
              ~~ ...        send line ~ ...

              The receive process recognizes output diversions:
                           ~>[>][:]file
                              lines
                           ~>
              where file is created or appended to (>>)
              on the local system; : for silent

uucp [−d] [−c] [−m] source-file ... destination-file
              −d = make necessary directories
              −c = don't spool; copy from source
              −m = send mail when done
       file names may be pathname or system-name!pathname
       pathname is either a full pathname,
              ~user pathname, or anything else (current
              directory is prefixed automatically)

uulog [−s.sys] [−u user]
              −s.sys = report only about sys
              −u user = report only about user
```

15

uux [−] *command*   { − makes stdin for *command* same as
      stdin for **uux**

SYSTEM MAINTENANCE COMMANDS

ac [−w *wtemp*] [−p] [−d] [*people*] ...
    −w *wtemp* = use alternate **wtemp** file
    −p = print individual totals
    −d = print by day (midnight to midnight)
    people = only those login names specified

/etc/accton [*file*]

arcv *file* ...

clri *filesys i-number* ...

dcheck [−i *numbers*] [*filesys*]   { −i report on specified i-numbers

df [*filesys*] ...   {no argument, report on all normally mounted syste

dump [*key* [*arg*] ... *filesys*]   *key* is chosen from:
    f = place dump on next argument
    u = if successful, write date on **/etc/ddate**; default
    0−9 = dump level; default 9
    s = size of dump tape; default 2300ft
    d = tape density; default 1600

dumpdir [**f** *filename*]   { **f** causes dump to go to *filename*

icheck [−s] [−b *numbers*] [*filesys*]
    −s = reconstruct new free list
    −b = report on specified block numbers

iostat [−tisb] [*interval* [*count*]]
    −t = report characters/second
    −i = report percentages
    −s = report raw timing info
    −b = report i/o buffer usage

/etc/mkfs *special-file proto*

/etc/mknod *name* [**bc**] *major minor*
    b = block-type device (disk, tape)
    c = character-type device

/etc/mount [*device directory* [−r]]   { −r = read-only

/etc/umount *device*

ncheck [−i *numbers*] [−a] [−s] [*filesys*]
    −i = report only on specified i-numbers
    −a = also report names beginning with . and ..
    −s = only report special files

pstat [−aixptuf] [*suboptions*] [*file*]
    −a = report on all process slots
    −i = report inode table
    −x = report text table
    −p = report active processes
    −t = report on terminals
    −u *addr* = report on user process at *addr*
    −f = report on open files

quot [−n] [−c] [−f] [*filesys*]
    −n = list all files and their owners
    −c = list files by size
    −f = list space and number of files for each user

restor *key* [*argument*] ...
    { *key* is one of following with optional **f**:
    f = use first argument as tape
    r = tape is loaded into file system *argument*
    R = asks which tape to start on - allows restart
    x = extract named files
    t = print date of dump

sa [−abcjlnrstuv] [*file*]
    a = report commands used once as **other**
    b = sort by (user time + system time)/number of calls
    c = print percentages
    j = print seconds/call
    l = separate system and user time
    m = print processes and CPU minutes for each user
    n = sort by number of calls
    r = sort in reverse order
    s = merge results into **/usr/adm/savacct**
    t = report (realtime)/(usertime+systime) for each command
    u = print user ID and command name
    vn = print commands used n times or less

Special Characters

|	pipeline
^	pipeline
;	sequential command separator
&	return without waiting for command to finish
&&	only execute command following if previous command returned 0 value
||	only execute command following if previous command returned non-zero value
'...'	take enclosed characters literally
"..."	literal except for parameter substitution
\	negates special meaning of following character

Input-Output

[d] <	use next argument as standard input with file descriptor d: default 0
[d] >	use next argument as standard output with file descriptor d: default 1
[d] >>	same as > but append to file if it exists
[d] <<	use as standard input shell input up to a line that matches the next argument
[d] <& digit	duplicate standard input from digit
[d] <& -	close standard input
[d] <>	open next argument for reading & writing

File Name Generation

?	match single character
*	match string of characters (including null)
[...]	match a class of characters
	a pair of characters separated by a — matches all characters lexically between the pair

Parameter Substitution

$n	replace $n with argument n
$*	"$1 $2 ..."
$@	"$1" "$2" ...
$#	number of positional parameters
$-	options supplied on invocation or by set
$?	value returned by last command
$$	process number of this shell
$!	process number of last background command
name= value	set variable
${parm— word}	use parm if set; otherwise word
${parm= word}	if parm not set, set it to word and use it
${parm? word}	use parm if set; otherwise print word and exit
${parm+ word}	use word if parm is set; otherwise nothing
$HOME	default argument for cd
$PATH	search path for commands
$MAIL	mail file
$PS1	primary prompt string; default $
$PS2	secondary prompt string; default >
$IFS	field separators; default blank, tab, newline
$TERM	terminal type (see command tabs)

Command Language

```
for name [ in word ...] do list done
case word in [ pat1 [ | pat2 ] ... ) list ;; ] ... esac
if list then list [ elif list then list ] ... [ else list ] fi
while list [ do list ] done
until list [ do list ] done
( list )
{ list }
```

19

break [*n*]	exit from **for** or **while** *n* levels
continue [*n*]	do next iteration of *n* th **for** or **while**
exit [*n*]	exit with return value *n*
:	no-op command but may contain text

Special Commands

. *file*	read and execute commands from *file*
eval [*arg* ...]	*arg* ... is read and executed
exec [*arg* ...]	execute *arg* ...
export [*name* ...]	export *name* to environment of commands
read *name* ...	read **stdin** and assign to *name* ...
readonly [*name* ...]	mark *name* read-only
set [−**eknptuvx** [*arg* ...]]	see **sh** command for flags
shift	rename positional parameters: $1 = $2 ...
times	print accumulated process times
trap [*arg*] [*n*]	execute *arg* if signal *n* is received
umask [*nnn*]	file creation mask is complement of octal *nnn*
wait [*pid*]	wait for process *pid*

ED

Regular Expressions

c	the character *c*
c	the character *c*
.	any character except <nl>
[*str*]	only those characters in *str*.
	a−*b* all characters from ascii *a* to *b*
[´*str*]	any character except those in *str* and <nl>
re*	all adjacent occurrences of *re*
\\(*re* \\)	*re*
n	*n* th *re* enclosed in \\(\\)
´	beginning of line
$	end of line
string of *re*'s	
null-re	last *re* encountered

Addresses

.	current line
$	last line
n	*n-th* line
´*x*	line marked with name *x*
/*re* /	first line searching forward containing *re*
?*re* ?	first line searching backward containing *re*
addr ± *n*	*addr* + (or −) *n*
± [*n*]	**.** + (or −) *n* : default 1
addr ±	*addr* + (or −) 1

Commands

(**.**)**a**	append
(**.** , **.**)**c**	change
(**.** , **.**)**d**	delete
e [*filename*]	edit
E [*filename*]	edit: no diagnostics
f [*filename*]	remembered name
(1 , $)**g**/*re* /*commands*	global
(**.**)**i**	insert
(**.** , **.** +1)**j**	join
(**.**)**k***x*	mark (addressed by ´*x*)
(**.** , **.**)**l**	list
(**.** , **.**)**m***a*	move (after *a*)
(**.** , **.**)**p**	print

q	quit
Q	quit; no diagnostics
(\$)r[*filename*]	read
(. . .)**s**/*re*/*repl*/	substitute; & in *repl* is *re*
(. . .)**s**/*re*/*repl*/**g**	substitute globally
(. . .)**t***a*	move copy (after *a*)
(. . .)**u**	undo previous substitution
(1 , \$)**v**/*re*/*commands*	like global but lines that don't match
(1,\$)**w** [*filename*]	write
(1,\$)**W** [*filename*]	append to file
x	encrypt
(\$) **=**	line number
!*UNIX command*	execute
(.+1) newline	print one line

ADB

General Command Format

[*addr*] [, *count*] [*command*] [;]

Commands

?[*format*]	print from *objfil*
/[*format*]	print from *corfil*
=[*format*]	print *address*
newline	repeat last command
[?\]l *value mask*	mask words until *value* is found
	(L used for 4 bytes)
[?\]w *value* ...	write 2 byte *value* at *addr*
	(W writes 4 bytes)
[?\]m *bl el fl* [?\]	modify map
>*name*	assign *dot* to **adb** *name*
!	escape to system
\$<*file*	read commands from *file*
\$>*file*	output to *file*
\$r	print general registers
\$f	print floating registers
\$b	print all breakpoints
\$a	print ALGOL 68 stack backtrace
\$c	print C stack backtrace
\$C	print C stack backtrace and automatic variables
\$e	print external variables
\$w	set page width; default 80
\$s	set symbol match limit to *addr*, default 255
\$o	input base is octal
\$d	reset input base
\$q	exit
\$v	print **adb** variables in octal
\$m	print address map
:b[*c*]	set breakpoint; execute breakpoint *c* — 1 times
:d	delete breakpoint
:r [*args*]	run *objfil* with *args*
:c[*s*]	continue subprocess with signal *s*; default previous
:s[*s*]	single step subprocess with signal *s*; default previous
:k	kill subprocess

Printing Formats

[*n*]*c* ... where n is a repeat count for *format*
   c chosen from the following:
   (* indicates capital *c* used for field twice as long)
   ((*n*) is temporary increment of *dot*)

o	print in octal * (2)
q	print in signed octal * (2)
d	print in decimal * (2)
x	print in hexadecimal * (2)
u	print unsigned decimal * (2)

21

f	print floating point * (4)
c	print character (1)
C	print character using @ as escape char (1)
s	print characters until 0 encountered (n)
S	print characters using @ as escape (n)
b	print byte in octal (1)
Y	print date format (4)
i	print PDP11 instruction (n)
a	print *dot* in symbolic form (0)
	/ = data, ? = text, = = absolute
p	print in symbolic form (2)
t	print tab (0)
r	print a space (0)
n	print a newline (0)
" "	print enclosed string (0)
.	decrement *dot* by current *incr*; print nothing
+	increment *dot* by 1; print nothing
−	decrement *dot* by 1; print nothing

Expressions

used for *addr* and *count*

.	value of *dot*	
+	. + *incr*	
.	. − *incr*	
"	last *addr* typed	
[0#] *integer*	number; 0 = octal. # = hexadecimal	
integer . fraction	32 bit floating point number	
'cccc'	up to 4 ascii characters	
< *name*	value of *name* ; *name* as follows:	
	0	last value printed
	1	last offset part of instr. source
	2	previous value of 1
	r0 ... r5 sp pc ps	registers
	b	data segment base addr
	d	data segment size
	e	entry point
	m	magic number
	s	stack segment size
	t	text segment size
symbol	value from symbol table in *objfil*	
	initial _ or ~ will be added if needed	
routine . name	variable *name* in C *routine*	
(*exp*)	value of *exp*	
**exp*	contents of *corfil* at *exp*	
@*exp*	contents of *objfil* at *exp*	
−*exp*	integer negation	
~*exp*	bitwise complement	
e1 op e2	*op* one of [+ − * %] (% = division)	
e1 & e2	bitwise conjunction	
*e1	e2*	bitwise disjunction	
e1 # e2	*e1* rounded up to multiple of *e2*	

* — command causes a break (suppressed 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 - troff; nroff

Ni	inches	$N*432$; $N*240$
Nc	cm.	$N*170$; $N*94$
Np	points	$N*6$; $N*3$
Nu	units	N; N
Nm	Ems	$N*6*$(point size); $N*$ nominal character width
Nn	Ens	$N*3*$(point size); same as m
NP	Picas	$N*72$; $N*40$
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 $N/36$m (12/36m) [ignored]
.cs $F N M$	constant character spacing; $N/36$m $M/36$m
	N is width; M is size of Em; no N turns cs off;
	no M implies size dependent (off)
.bd $F N$	embolden font by $N-1$ (off)
.bd S $F N$	embolden special font when in font F (off)
.ft F	font change (Roman) [previous]
.fp $N F$	font position, 1 to 4 (R,I,B,S) [ignored]
.fz $F \pm N$	font F always in size N
.fz S $F \pm N$	special font in size N when in F

Page Control

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

Text Filling, Adjusting, Centering, and Underlining

.br	break *
.fi	fill * (on)
.nf	no filling & adjusting * (off)
.ad c	adjust mode: n or b(both adjusted), r(right adjusted),
	l(left adjusted), c(centered) (n) (unchanged)
.na	no adjusting (off)
.mc $c N$	specify margin character (off) [off]
.ce N	center N input lines * (off) (1)
.ul N	nroff underline alphanumerics; troff italicize (0 line)
.cu N	nroff continuous underline; troff italicize (0 line)
.uf F	underline font set to F (I) [I]

Vertical Spacing

.vs N	vertical spacing; v (12p) (1/6i) [previous]
.ls N	line spacing (1) [previous]
.sp $-N$	space vertically $-N$ * (1v)
.sv N	save vertical distance (1v)
.os	output saved vertical distance
.ns	no space mode (off)
.rs	restore spacing

Line Length and Indenting

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

23

`.ti ±N`   temporary indent * [ignored]

Macros, Diversion, and Line Traps

`.de xx yy`	define macro; end at call of *yy* (*yy*=.) [ignored]
	called by .*xx*
`.am xx yy`	append to macro [ignored]
`.ds xx str`	define string; invoked by *x or *(*xx* [ignored]
`.as xx str`	append to string [ignored]
`.rn xx yy`	rename *xx* to *yy*
`.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 xx N`	change line trap for *xx* to place *N*
`.em xx`	end-macro name specification (none) [none]
`.dt N xx`	set diversion trap [off]
`.it N xx`	set input text line count trap [off]

Number Registers

`.nr a ±N M`	number & increment register; called by \\n*a*, \\n+*a*
	or \\n−*a*
`.nr ab ±N −M`	number register; called by \\n(*ab*, \\n(+*ab*
	or \\n−(*ab*
`.af xx c`	assign format to *xx*. 1(1,2,), 001(001,002,), i(i,ii,), I(I,II,)
	a(a,b,), A(A,B,)
`.rr xx`	remove register

Tabs, Leaders, and Fields

`.ta N [RC] ...`	tab settings & types; +*N* = increment
	((.5i ...),(.8i ...) left justified) [none]
`.tc c`	tab repetition character (none) [none]
`.lc c`	leader repetition character (.) [none]
`.fc a b`	set field delimiter & pad character (off) [off]

Input and Output Conversions & Character Translations

`.ec c`	set escape character (\\) [\\]
`.eo`	turn off escape processing (off) [off]
`.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), ≠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 (\\%) [\\%]
`.hw word1 ...`	hyphenation exception list

Three Part Titles

`.tl 'l'c'r'`	title
`.lt N`	length of title (6.5i) [previous]
`.pc c`	page number character (%) [off]

Output Line Numbering

`.nm ±N M S I`	number mode on or off, set parameters [off]
`.nn n`	don't number next *n* lines [1]

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
.if *'s1' s2' anything*   if *s1* = *s2*
.if *!'s1' s2' anything*   if *s1* != *s2*
.ie *c anything*   same as **.if** but has else with it
.el *anything*   else

Input Switching

.so *filename*   switch source file-push down
.nx *filename*   next file [EOF on current file]
.pi *prog*   **nroff** pipe output to *prog*
.rd *prompt*   read insert from **stdin** [bell]
.ex   exit

Miscellaneous

.ev *N*   environment pushed down (0) [previous]
.mc *c N*   right margin character (1m, .2i) [off]
.tm *string*   typewriter message
.ig *yy*   ignore until *yy* is called (*yy*=.)
.fl   flush output buffer *
.pm *t*   print macros: *t*=print only sum of sizes
.ab *str*   print string & abort

Escape Sequences - Characters, Indicators, & Functions

\\	prevent interpretation of \
\e	print current escape character
\`	(acute accent): same as \\aa
\`	(grave accent): same as \\ga
\-	- (minus)
_	_ : same as \\ul
|	1/6 m space character: 0 width in nroff
\^	1/12 m space character: 0 width in nroff
\(space)	unpaddable space
\&	zero width character
\0	digit width space
\!	transparent line indicator
\"	beginning of comment
\%	default optional hyphenation character
\{	beginning of conditional input
\}	end of conditional input
\$	argument indicator: args 1-9
\(xx	character named xx
*	string indicator: *x. *(xx
\a	leader character
\b'abc'	bracket building function
\c	interrupt text processing
\d	forward (down) 1/2 m (troff) or line (nroff)
\f	font change function
\h	local horizontal motion function
\k	mark horizontal place
\l	draw horizontal line
\L	draw vertical line
\n	number register indicator: \nx \n(xx
\o'abc'	overstrike function
\p	break and spread output line
\r	reverse 1m or line vertical motion
\s	point size change function
\t	non-interpreted horizontal tab
\u	reverse (up) 1/2 m or line
\v	local vertical motion function
\w	width function
\x	extra line-space function
\z	zero width character function
\(newline)	concealed newline

25

Reserved Registers

(r) = read only

%	current page number
.$	number of arguments available at macro level (r)
.A	1 if **troff** **−a** or **nroff**, 0 otherwise (r)
.H	available horizontal resolution (r)
.L	current line-spacing parameter (r)
.P	1 if current page being printed; 0 otherwise (r)
.T	1 if **nroff** **−T**, 0 otherwise (r)
.V	available vertical resolution (r)
.a	most recent post-line application of \x (r)
.c	input line number in current file (r)
c.	input line number in current file
.d	current vertical place in diversion (r)
.f	physical quadrant of current font (r)
.h	high-water mark of nl for text on current page (r)
.i	current indent (r)
.j	current adjustment mode and type (r)
.k	horizontal size of current output line (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)
.u	1 in fill mode; 0 in nofill mode (r)
.v	current vertical line spacing (r)
.w	width of previous character (r)
.x	reserved version-dependent register (r)
.y	reserved version-dependent register (r)
.z	name of current diversion (r)
ct	character type
dl	width of last completed diversion
dn	height of last diversion
dw	current day of the week number
dy	current day of the month
hp	current horizontal place on input line
ln	output line number
mo	current month number
nl	current vertical place on the page
sb	depth of string below base line
st	height of string above base line
yr	last 2 digits of current year

TROFF SPECIAL CHARACTERS

'	\' close			fi	\(fi
`	\` open			fl	\(fl
-	\- minus			ff	\(ff
—	\(em			ffi	\(Fi
·	\(hy or −			ffl	\(Fl
•	\(bu			°	\(de
□	\(sq			†	\(dg
_	\(ru			'	\(fm
¼	\(14			¢	\(ct
½	\(12			®	\(rg
¾	\(34			©	\(co

+	\(pl	B	\(*B	×	\(mu
−	\(mi	Γ	\(*G	÷	\(di
=	\(eq	Δ	\(*D	±	\(+−
*	\(**	E	\(*E	∪	\(cu
§	\(sc	Z	\(*Z	∩	\(ca
´	\(aa	H	\(*Y	⊂	\(sb
`	\(ga	Θ	\(*H	⊃	\(sp
	\(ui	I	\(*I	⊆	\(ib
‾	\(sl	K	\(*K	⊇	\(ip
α	\(*a	Λ	\(*L	∞	\(if
β	\(*b	M	\(*M	∂	\(pd
γ	\(*g	N	\(*N	∇	\(gr
δ	\(*d	Ξ	\(*C	¬	\(no
ε	\(*e	O	\(*O	∫	\(is
ζ	\(*z	Π	\(*P	∝	\(pt
η	\(*y	P	\(*R	∅	\(es
θ	\(*h	Σ	\(*S	∈	\(mo
ι	\(*i	T	\(*T		\(br
κ	\(*k	Y	\(*U	‡	\(dd
λ	\(*l	Φ	\(*F	▬	\(rh
μ	\(*m	X	\(*X	▬	\(lh
ν	\(*n	Ψ	\(*Q	⊕	\(bs
ξ	\(*c	Ω	\(*W		\(or
o	\(*o	√	\(sr	○	\(ci
π	\(*p		\(rn		\(lt
ρ	\(*r	≥	\(>=		\(lb
σ	\(*s	≤	\(<=		\(rt
ς	\(ts	≡	\(==		\(rb
τ	\(*t	≅	\(~=		\(lk
υ	\(*u	~	\(ap		\(rk
φ	\(*f	≠	\(!=		\(bv
χ	\(*x	→	\(->		\(lf
ψ	\(*q	←	\(<-		\(rf
ω	\(*w	↑	\(ua		\(lc
A	\(*A	↓	\(da		\(rc

−ms Option for TROFF & NROFF

* − command causes a break; () − initial value

general calling order:

> *paper-type,* **TL**, *author & other 1st page info, abstract,*
> *heading-type, body, trailing info*

Format & Abstract

.TM x y z	BTL TM format: tm# case# file#
.EG	BTL Engineer's Notes format
.IM	BTL internal memorandum format
.MF	BTL Memorandum-for-File format
.MR	BTL Memorandum-for-Record format
.RP	released paper cover sheet
.TR x	BTL technical report format, report #
.TL	title follows *
.AU ad ex	authors names follow; address extension *

27

.AI	authors institute follows (.MH, .HO, .WH, .IH, .PY)
.AB	begin abstract *
.AE	end of abstract *
.CS *data*	cover sheet data *
	#text #other total #fig #tab #ref
.OK	other keywords follow *
.SG *x*	signature line follows - for TM's : reference line *
.AT	attachments
.CT	copies to
.BT	bottom title
.PT	top title

Headings & Paragraphs

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

Fonts, Sizes, & Emphasis

.*f x*	change fonts or print *x* in font: Bold, Italic
.R	restore font
.*sz*	size change: LG=larger, SM=smaller, NL=normal
.UL *x*	underline *x*
.TA *x* ...	set tabs in ens; default: 5 10 ...
.BX *x*	print *x* in a box

Footnotes

.FS	start footnote
.FE	end footnote
.UX	UNIX trademark footnote
.US	"the .UX operating system"

Displays, Tables, & Equations

.DS [CLIB]	begin display: C=center, L=left adjust, I=indent, B=left-justified centered * [I]
.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 *
.B1	begin boxed text
.B2	end boxed text
.QS	begin quoted text (indented, shorter)
.QE	end quoted text
.QP	begin single quoted paragraph
.EQ [CLI] *n*	begin equation: for CLI see DS, *n*=equation number
.EN	end equation *
.TS *x*	begin table: repeated heading if *x*=H
.TE	end table
.TH	end table heading section

Miscellaneous

.2C	2 column *
.1C	1 column (on) *
.DA [*x*]	current date - on for **nroff**
.ND [*date*]	change or cancel date - on for **troff**

Registers & Strings

| CF | center footer; string |
| CH | center heading; string |

CW	column width for 2C; register; (7/15 LL)
FL	footnote length; register; (11/12 LL)
FM	bottom margin; register; (1 in)
GW	intercolumn gap for 2C; register; (1/15 LL)
HM	top margin; register; (1 in)
LF	left footer; string
LH	left heading; string
LL	line length; register; (6 in)
LT	title length; register; (6 in)
PD	paragraph spacing; register; (.3 VS)
PI	paragraph indent; register; (5 ens)
PN	page number; string
PO	page offset; register; (26/27 in)
PS	point size; register; (10)
RF	right footer; string
RH	right heading; string
VS	line spacing; register; (12 pts)

EQN & NEQN KEYWORDS

sub, sup
over
sqrt
...from...to...
left c, right c
pile { ...above... }, lpile, cpile, rpile
dot, dotdot, hat, bar, tilde, under, vec, dyad
size n, gsize n
roman, italic, bold, font f, gfont f
delim
define, tdefine, ndefine
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, lim, max, min, log, ln, exp
prime, cdot, del, half
....., ...
uppercase and lowercase greek
infinity, inf, partial, grad

Other Symbols from /usr/pub/eqnchar

$\oplus$	ciplus	$\exists$	oppE	$\rangle$	rangle
$\otimes$	citimes	$\equiv$	incl	$\hbar$	hbar
$\cong$	=wig	$\in$	member		
$\doteq$	=dot	$\emptyset$	empty	$\perp$	ppd
$\triangleq$	=del	$\propto$	prop	$\angle$	ang
$*$	bigstar	$\notin$	nomen	$\llcorner$	rang
$*$	star	$\AA$	angstrom	$\longrightarrow$	<->
$\vee$	orsign	$\circ$	degree	$\longleftrightarrow$	<=>
$\wedge$	andsign	$\leq$	<wig	$\vdots$	3dot
$\forall$	oppA	$\sim$	wig	$\therefore$	thf
$\frac{1}{4}$	quarter	$\simeq$	~wig	$\cup$	cup
$\frac{1}{2}$	3quarter	$\geq$	>wig	$\cap$	cap
$\square$	square	$\prec$	'<	$\subseteq$	subset
$\bigcirc$	circle	$\prec$	'<	$\subset$	'subset
$\blacksquare$	blot	$\succ$	|>	$\supseteq$	supset
$\bullet$	bullet	$/$	langle	$\supset$	'supset

TBL

Options

center	center table
expand	expand to line length
box	enclose in a box
allbox	box all entries
doublebox	enclose in two boxes
tab(*c*)	change tab character to *c*
linesize(*n*)	line thickness is *n*
delim(*xx*)	forces eqn equations to be kept together

Formats

l	left-adjusted column
r	right-adjusted column
c	centered column
n	alligned number column
a	left-adjusted subcolumn
d	push vertical span to bottom
t	push vertical span to top
v	vertical spacing
s	spanned heading
^	vertically spanned heading
_	horizontal line
=	double horizontal line
|	vertical line
b	bold column
i	italic column
p*n*	point size for column
w(*n*)	column width
n	space between columns
e	equal width columns
.	end of format

Data

T{ ... T}	text block
_	short horizontal line
\\^	above item spans downward to this row

−mm Option for TROFF & NROFF

[] − in macro definition, optional arg; otherwise, value if no arg
() − initial value; {} − select one
general order:
   TL, *author info*, *other cover-info*, *abstract*, MT,
       body, *trailing info*

Format & Abstract

.MT [""012345]	document type: 0 = "" = no type, 1 = mm for file.
	2 = programmer's notes, 3 = engineer's notes, 4 = released paper,
	5 = letter; **must occur after** all cover sheet info [1]
.ND *date*	new date
.TL [*chg#*] [*file#*]	title follows
.AF [*co-name*]	alternate 1st page format
.AU *name* [*initl*] [*loc*] [*dept*] [*ext*] [*room*] [*arg*] ...   author info
.TM [*number*]	technical memorandum numbers
.AS [01] [*indent*]	abstract start; 0 = abstract on cover-sheet & 1st p
	1 = abstract only on cover-sheet
.AE	abstract end
.OK [*keyword* ...]	other keywords
.SG [*initls*] [1]	signature line
.NS [""0123456789]	notation start; "" = 0 = copy-to, 1 = copy-with-att,
	2 = copy-without-att, 3 = att, 4 = atts, 5 = enc, 6 = encs,
	7 = sep-cov, 8 = letter-to, 9 = mm-to
.NE	notation end

```
.CS [pgs] [other] [tot] [figs] [tbls] [ref]    cover sheet
.TC [s[lev]] [spac] lev [tab] [head] ...        table-of-contents
.TX         user exit for table-of-contents
```

Headings & Paragraphs

```
.P [01]        paragraph; 0=left-justified, 1=indented [\n(Pt]
.H {1234567} [h-text] numbered headings
.HU h-text            unnumbered headings
.HM {1 0001 Aali} ... heading mark style; 1=arabic, 0001=arabic wit
               leading 0's, A=upper case, a=lower case, I=upper roman
               i=lower roman
.HX dlev rlev h-text  user exit before headings
.HZ dlev rlev h-text  user exit after headings
```

Lists

presence of last argument on start macros = no blank line between it

```
.AL {1Aali} [t-indent] [1]        auto-incremented start [1 \n(Li]
.BL [t-indent] [1]                bullet start [\n(Pi]
.DL [t-indent] [1]                dash start; [\n(Pi]
.ML mark [t-indent] [1]           marked start; [mark-width+1]
.RL [t-indent] [1]                reference start; [6]
.VL t-indent [m-indent] [1]       variable-item start [0]
.LI [mark] [1]     list item follows; 1=mark is prefix [current ma
.LE [1]    list end; arg=blank line output [no blank]
.LB t-indent m-indent pad type [mark] [01] [01]        list begin
         type: 1=. 2=) 3=() 4=[] 5=<> 6={}
         6th arg=blank before each item; 7th arg=blank before list
.LC [lev]          clear list-status
```

Displays, Tables, Equations, & Footnotes

```
.DS [LIC CB] [01]    start static display; L=no indent, I=indent,
              C=center each line, CB=center as a block; 0=no-fill, 1=
              [L 0]
.DF [LIC CB] [01]    start floating display; args same as .DS
.DE           end display
.FG [titl] [o-ride] [012]        figure caption
         0=prefix with o-ride, 1=suffix, 2=replace
.TS           start table
.TE           end table
.TB [titl] [o-ride] [012]        table caption
.EQ [label]                      start equation display
.EN           end equation
.EC [titl] [o-ride] [012]        equation caption
.FS [lab]     start footnote
.FE           end footnote
.FD [arg] [1]       default footnote format
     arg is a decimal number < 12 created from bit definitions
         with the following meanings (each defines 1 bit):
         right-justified label; indented text; ragged right margin;
         hyphenation on
     2nd arg=reset footnote counter on 1st level heading
```

Page Headers & Footers

```
.PH 'left'center'right' page header ("page-number")
.OH 'left'center'right' odd-page header
.EH 'left'center'right' even-page header
.PF 'left'center'right' page footer
.OF           odd-page footer
.EF           even-page footer
.BS           bottom-block start
.BE           bottom-block end
.PX           user exit for page-header
.TP           top of page macro
```

31

Miscellaneous

.B [*arg*] [*prev-font-arg*]		bold
.I [*arg*] [*prev-font-arg*]		italic
.R	return to regular font	
.SP [*lines*]	skip vertically	
.SK [*pages*]	skip pages (next page)	
.OP	odd-numbered page start	
.2C	two-column	
.1C	one-column	
.SA [*a*]	set right-margin justification	
.HS *c*	set hyphenation character	
.S ± *N*	set point size; vertical spacing = point size + 2 (10) [previo

Registers & Strings

Au	inhibit author info on 1st page (1)
BU	bullet string
Cl	contents level (2)
Ds	static display pre & post space; (on)
DT	current date string
Ec	equation counter
Ej	if 1, begin 1st level headings at top of page (0)
F	footnote numberer string
Fg	figure counter
Fs	footnote separation (1)
H1 ... H7	heading counters
Hb	heading level above which there's a break (2)
Hc	heading level below which heading is centered (0)
Hi	for stand alone headings; 0 = left-justify, 1 = indent by Pt, 2 = line-up with heading (1)
Hs	heading level below which blank line added (2)
Ht	heading type; 0 = concatenated nums, 1 = single nums; (0)
Hu	heading level for unnumbered heading
Hy	hyphenation control (1)
HF	heading level font string; 1 = roman, 2 = italic, 3 = bold (3322222)
HP	heading level point size
Li	list indent (5)
Ls	list spacing between items (6)
Pi	value of paragraph indent (5)
Pt	paragraph type (2)
Si	display indent (5)
Tb	table counter

C SYSTEM CALLS

variables with no declarations are **int**

access(file, mode)   check mode access of file
char *file;   *mode* = 4(read), 2(write), 1(execute)

alarm(seconds) receive alarm signal
unsigned seconds;

char *brk(addr) set lowest location to *addr*

chdir(dirname) change working directory
char *dirname;

chmod(name, mode) change mode of file
char *name;   (for *mode* see chmod p. 5)

close(fildes)   close a file

creat(name, mode)   creat a new file
char *name;   (for *mode* see chmod p. 5)

dup(fildes) duplicate an open file descriptor

dup2(fildes, fildes2) integer *fildes2* set to refer to *fildes*

execl(name, arg0, arg1, ..., argn, 0)   execute a file
char *name, *arg0, *arg1, ..., *argn;

execv(name, argv)   execute a file
char *name, *argv[ ];

execle(name, art0, arg1, ..., argn, 0, envp); execute a file
char *name, *arg0, *arg1, ..., *argn, *envp[ ];   with an environment

execve(name, argv, envp); execute a file with an environment
char *name, *argv[ ], *envp[ ];

exit(status)   terminate process

fork( ) spawn new process

fstat(fildes, buf) get status of open file
struct stat *buf

getpid( ) get process id

getuid( ) get real user ID

geteuid( ) get effective user ID

getgid( ) get real group ID

getegid( ) get effective group ID

gtty(fildes, argp) get typewriter status
struct sgttyb *argp; include file sgtty.h

kill(pid, sig) send signal to process; see kill p. 8

link(oldname, newname) link to a file
char *oldname, *newname;

long lseek(fildes, offset, whence) move read/write pointer
long offset; *whence* 0(*offset*), 1(+*offset*), 2(end+*offset*)

mount(special, name, rwflag) mount file system
char *special, *name;

nice(incr) lower program priority; 0 < *incr* <= 20 (least)

open(name, mode) open for reading(0) writing(1), or both(2)

pause( ) wait indefinitely (or for alarm)

pkoff(fildes) turn off packet driver

pkon(fildes, size) turn on packet driver; 32 ≤ *size* ≤ 4096

pipe(fildes)   create an interprocess channel;
int fildes[2];   read[0], write[1]

profil(buff, bufsiz, offset, scale) execute time profile
char *buff;

ptrace(request, pid, addr, data) process trace
int *addr; *request* has the following meanings.
   0   process to be traced by parent
   1,2 return value at *addr* in child; for separate I&D
        1 = I space, 2 = D space
   3   returns value at *addr* in system's per-process data space
   4,5 *data* is written at *addr*. 4 = I space. 5 = D space

33

6	*data* is written at *addr* in system's per-process data space
7	continue child at *addr* with signal *data*. *data* = (int *)1 means continue from where stopped
8	terminates traced process
9	continue as 7 but stop with **SIGTRAP** as soon as possible

read(fildes, buffer, nbytes) read from file
char *buffer;

char *sbrk(incr) add to data space

setgid(gid) set process group id

setuid(uid) set process user id

(*signal(sig, func))( )　　catch or ignore signals; see kill p. 8
(*func)( );　　func=SIG_DFL default, SIG_IGN ignore
　　　　must include signal.h

stat(name, buf) get file status
char *name;　　must include sys/types.h sys/stat.h
struct stat *buf;

sync()　　update super-block

stty(fildes, argp)　　set typewriter modes
struct sgttyb *argp;　　must include sgtty.h

long time(0);　　get date and time since 00:00:00 GMT, Jan. 1, 1970

long time(tloc) store date and time in *tloc*
long *tloc;

ftime(tp) store data and time in *tp*
struct timeb *tp; must include sys/types.h sys/timeb.h

times(buffer)　　get process times
struct tbuffer {
　　long proc_user_time;
　　long proc_sys_time;
　　long child_user_time;
　　long child_sys_time;
} *buffer;

umount(special) remove file system
char *special;

umask(complmode)　　set file creation mode mask
　　　　(for *complmode* see chmod p. 5)

unlink(name)　　remove directory entry
char *name;

utime(file, timep)　　set file times
char *file; must include sys/types.h
time_t timep[2];

wait(status)　　wait for process to terminate
int *status;

write(fildes, buffer, nbytes) write on a file
char *buffer;

C STDIO ROUTINES

#include <stdio.h>　　include file for definitions

In the following routines variables must be declared as:
FILE *stream;
char *type;　　where "r" = read, "w" = write, "a" = append
char *filename;
char *format;
char *s;
char c;
all other variables are int unless otherwise specified

clearerr(stream) resets error indication

fclose(stream)　　flush and close stream

FILE *fdopen(fildes, type) returns stream associated with fildes

feof(stream)   returns non-zero on EOF

ferror(stream)   returns non-zero on read/write error

fflush(stream)   flush stream

int fgetc(stream)   function like macro **getc**

char *fgets(s, n, stream)   returns n−1 characters in s

fileno(stream)   returns integer file descriptor

FILE *fopen(filename, type)   open a stream

fprintf(stream, format [, arg ...])   output to **stream**
       see **printf** for form of **format**

fputc(c,stream)   function like macro **putc**

fputs(s,stream)   output a string

fread(ptr, sizeof(*ptr), nitems, stream)   read

FILE *freopen(filename, type, stream)   use filename for open stream

fscanf(stream, format [, pointer ...])   input and convert
       format is of the form:
          %[*][fw] c
       where * suppresses assignment and c is one of
       [cdDefoOsxX] optionally preceeded by l as in **printf**
       or [...] where characters are input until
       the first one not between the brackets or [^ ...] where
       characters are input until the first one between the brackets

fseek (stream, offset, ptrname)   ptrname=0(beginning).
long offset;   1(current place), 2(end)

long ftell(stream)   returns current offset

fwrite(ptr, sizeof(*ptr), nitems, stream)   write

int getc(stream)   returns next character: EOF on EOF or error

int getchar()   returns next character from **stdin**

char *gets(s)   returns string from **stdin** in s

int getw(stream)   returns next word

pclose(stream)   to close stream opened with **popen**

FILE *popen(command, type)   create a pipe
char *command, *type;

printf(format [, arg ...])   output to **stdout**
      format consists of characters and conversion
        specifications of the form:
          %[−][fw[.d] c
      where − specifies left adjustment; fw, field width.
      d. precision; and c may be
          [l]d   decimal (l or D =long)
          [l]o   octal (l or O = long)
          [l]x   hexadecimal (l or X = long)
          [l]u   unsigned integer (l or U = long)
          f   float of form [−]dd.ddd
          e   float of form [−]d.dde±dd
          g   float using minimum space(d, f. or e)
          c   character
          s   string
          %   escape %

rewind(stream)   rewind

int putc(c, stream)   output a character

putchar(c)   output a character to **stdout**

putw(w,stream)   output a word

puts(s)   output string to **stdout**

scanf(format [, pointer ...])   read **stdin** and convert
       see **fscanf** for form of **format**

setbuf(stream, buf)   buffer i/o in buf
char buf[BUFSIZ];

sprintf(s,format [, arg...])   output to string s
       see **printf** for form of **format**

sscanf(s,format [, pointer...])   read from string s and convert
       see **fscanf** for form of **format**

ungetc(c, stream)   push c back on input stream

C LIBRARY ROUTINES

#include <math.h> include file for math functions
double x, y;   declaration for x & y in math functions

abs(i)   absolute value of **i**

double acos(x)   arc cosine function; range 0 to π

double asin(x)   arc sin function; range $-\pi/2$ to $\pi/2$

double atan(x)   arc tangent function; range $-\pi/2$ to $\pi/2$

double atan2(x,y)   arc tangent of x/y; range $-\pi$ to π

double atof(nptr)   converts ascii to float
char *nptr;

atoi(nptr) converts ascii to integer
char *nptr;

long atol(nptr) converts ascii to long integer
char *nptr;

double cos(x)   cosine function

double cosh(x)   hyperbolic function

char *ctime(clock)   converts time to ascii
long *clock;

char *ecvt(value, ndigit, decpt, sign)   converts *value* to ascii
double value;
int ndigit, *decpt, *sign;

double exp(x)   exponential function

char *getenv(name)   search environment for **name**
char *name;   returns NULL for not found; value for found

double log(x)   natural log function

double log10(x)   log function base 10

char *malloc(size)   memory allocation
unsigned size;   release with **free(ptr)**

   char *realloc(ptr, size)   change block size or compact

   char *calloc(nelm, elsize)   allocate & clear array space
   unsigned nelm, elsize;

char *mktemp(template)   make unique file name
char *template; template of form *name*XXXXXX

perror(s) print s: last error from system call
char *s;

double pow(x,y) power function; returns x^y

qsort(base, num_elts, width, compar) quicksort
char *base;
int (*compar)();

rand()   random number generator; initialized with **srand(seed)**

double sin(x)   sine function.

double sinh(x)   hyperbolic function

sleep(seconds) suspend execution
unsigned seconds;

double sqrt(x)   square root function

system(string) sends **string** to system for execution
char *string;

double tan(x)   tangent function

double tanh(x)   hyperbolic function

SOURCES

1. Bell Laboratories, *UNIX^{TM} TIME-SHARING SYSTEM: UNIX PROGRAMMER's MANUAL*, Seventh Edition, Vol. 1 & 2 (January 1979).
2. D. W. Smith, J. R. Mashey, *PWB/MM Programmer's Workbench Memorandum Macros*, (October 1977).

ASCII

000 nul	001 soh	002 stx	003 etx	004 eot	005 enq	006 ack	007 bel
010 bs	011 ht	012 nl	013 vt	014 np	015 cr	016 so	017 si
020 dle	021 dc1	022 dc2	023 dc3	024 dc4	025 nak	026 syn	027 etb
030 can	031 em	032 sub	033 esc	034 fs	035 gs	036 rs	037 us
040 sp	041 !	042 "	043 #	044 $	045 %	046 &	047 '
050 (	051)	052 *	053 +	054 ,	055 -	056 .	057 /
060 0	061 1	062 2	063 3	064 4	065 5	066 6	067 7
070 8	071 9	072 :	073 ;	074 <	075 =	076 >	077 ?
100 @	101 A	102 B	103 C	104 D	105 E	106 F	107 G
110 H	111 I	112 J	113 K	114 L	115 M	116 N	117 O
120 P	121 Q	122 R	123 S	124 T	125 U	126 V	127 W
130 X	131 Y	132 Z	133 [	134 \	135]	136 ^	137 _
140 `	141 a	142 b	143 c	144 d	145 e	146 f	147 g
150 h	151 i	152 j	153 k	154 l	155 m	156 n	157 o
160 p	161 q	162 r	163 s	164 t	165 u	166 v	167 w
170 x	171 y	172 z	173 {	174		175 }	176 ~	177 del