I. COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar</td>
<td>archive (combine) files</td>
</tr>
<tr>
<td>as</td>
<td>assembler</td>
</tr>
<tr>
<td>b</td>
<td>compile B program</td>
</tr>
<tr>
<td>bas</td>
<td>BASIC dialect</td>
</tr>
<tr>
<td>bcd</td>
<td>convert ASCII to BCD</td>
</tr>
<tr>
<td>boot</td>
<td>reboot system</td>
</tr>
<tr>
<td>cat</td>
<td>concatenate (or print) files</td>
</tr>
<tr>
<td>chdir</td>
<td>change working directory</td>
</tr>
<tr>
<td>check</td>
<td>check consistency of file system</td>
</tr>
<tr>
<td>chmod</td>
<td>change access mode of files</td>
</tr>
<tr>
<td>chown</td>
<td>change owner of files</td>
</tr>
<tr>
<td>cmp</td>
<td>compare file contents</td>
</tr>
<tr>
<td>cp</td>
<td>copy file</td>
</tr>
<tr>
<td>date</td>
<td>get date and time of day</td>
</tr>
<tr>
<td>db</td>
<td>symbolic debugger</td>
</tr>
<tr>
<td>dbppt</td>
<td>write binary paper tape</td>
</tr>
<tr>
<td>dc</td>
<td>desk calculator</td>
</tr>
<tr>
<td>df</td>
<td>find free disk space</td>
</tr>
<tr>
<td>dsw</td>
<td>delete files interactively</td>
</tr>
<tr>
<td>dtpf</td>
<td>format DECTape</td>
</tr>
<tr>
<td>du</td>
<td>find disk usage</td>
</tr>
<tr>
<td>ed</td>
<td>text editor</td>
</tr>
<tr>
<td>find</td>
<td>find file with given name</td>
</tr>
<tr>
<td>for</td>
<td>compile Fortran program</td>
</tr>
<tr>
<td>form</td>
<td>generate form letter</td>
</tr>
<tr>
<td>hup</td>
<td>hang up typewriter</td>
</tr>
<tr>
<td>lbppt</td>
<td>read binary paper tape</td>
</tr>
<tr>
<td>ld</td>
<td>link editor (loader)</td>
</tr>
<tr>
<td>ln</td>
<td>link to file</td>
</tr>
<tr>
<td>ls</td>
<td>list contents of directory</td>
</tr>
<tr>
<td>mail</td>
<td>send mail to another user</td>
</tr>
<tr>
<td>mesg</td>
<td>permit or deny messages</td>
</tr>
<tr>
<td>mkdir</td>
<td>create directory</td>
</tr>
<tr>
<td>mkfs</td>
<td>initialize file system</td>
</tr>
<tr>
<td>mount</td>
<td>mount detachable file system</td>
</tr>
<tr>
<td>mv</td>
<td>move or rename file</td>
</tr>
<tr>
<td>nm</td>
<td>print namelist</td>
</tr>
<tr>
<td>od</td>
<td>octal dump of file</td>
</tr>
<tr>
<td>pr</td>
<td>print file with headings</td>
</tr>
<tr>
<td>rew</td>
<td>rewind DECTape</td>
</tr>
<tr>
<td>rkd</td>
<td>dump disk to tape</td>
</tr>
<tr>
<td>rkf</td>
<td>format RK disk</td>
</tr>
<tr>
<td>rkl</td>
<td>load disk from tape</td>
</tr>
<tr>
<td>rm</td>
<td>remove (delete) file</td>
</tr>
<tr>
<td>rmdir</td>
<td>remove (delete) directory</td>
</tr>
<tr>
<td>roff</td>
<td>run off (format) text</td>
</tr>
<tr>
<td>sdate</td>
<td>adjust date and time</td>
</tr>
<tr>
<td>sh</td>
<td>command interpreter</td>
</tr>
<tr>
<td>stat</td>
<td>get file status</td>
</tr>
<tr>
<td>strip</td>
<td>remove symbols, relocation bits</td>
</tr>
<tr>
<td>su</td>
<td>become super-user</td>
</tr>
</tbody>
</table>
II. SYSTEM CALLS

break ......................... set program break
cemt ........................ catch EMT traps
chdir ......................... change working directory
chmod ........................ change mode of file
chown ........................ change owner of file
close ......................... close open file
creat ........................ create file
exec ........................ execute program file
exit .......................... terminate execution
fork .......................... create new process
fstat ........................ status of open file
getuid ....................... get user ID
getty ........................ get typewriter mode
ilgins ....................... catch illegal instruction trap
intr .......................... catch or inhibit interrupts
link .......................... link to file
mkdir ........................ create directory
mount ........................ mount file system
open .......................... open file
quit .......................... catch or inhibit quits
read .......................... read file
release ...................... release processor
seek .......................... move read or write pointer
setuid ....................... set user ID
smdate ...................... set date modified of file
stat .......................... get file status
stime ........................ set system time
stty ........................ set mode of typewriter
tell ........................ find read or write pointer
time ........................ get time of year
umount ...................... dismount file system
unlink ...................... remove (delete) file
wait ........................ wait for process
write ....................... write file

III. SUBROUTINES

atof ........................ convert ASCII to floating
atoi ........................ convert ASCII to integer
ctime ........................ convert time to ASCII
exp .......................... exponential function

- vi -
fptrap ...................... floating-point simulator
ftoa ........................ convert floating to ASCII
get ........................... get character
itoa .......................... convert integer to ASCII
log ................................ logarithm base e
msg ................................ print string on typewriter
ptime ............................ print time
putc ............................. write character or word
sin ................................ sine, cosine
switch ............................ transfer depending on value

IV. SPECIAL FILES

mem .............................. core memory as file
ppt .............................. punched paper tape
rf0 .............................. RF disk file
rk0 .............................. RK disk file
tap0,...,tap7 ................. DECTape file
tty .............................. console typewriter
tty0,...,tty5 ................. remote typewriter

V. FILE FORMATS

a.out .......................... assembler and loader output
archive ......................... archive file
bppt ............................ binary paper tape format
core ............................. core image file
directory ....................... directory format
file system ..................... file system format
passwd ........................ password file
uid .............................. map names to user ID's
utmp ............................ logged-in user information

VI. USER MAINTAINED PROGRAMS

basic ........................... DEC supplied BASIC
bj .............................. the game of black jack
cal .............................. print calendar
chess ........................... the game of chess
das .............................. disassembler
dl1 .............................. load DEC binary paper tapes
dpt .............................. read DEC ASCII paper tapes
moo .............................. the game of MOO
sort .............................. sort a file
ttt .............................. the game of tic-tac-toe

VII. MISCELLANEOUS

as2 ............................ assembler's pass 2
ascii .......................... map of ASCII
ba .............................. B assembler
bc .............................. B compiler

- vii -
<table>
<thead>
<tr>
<th>Word(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bilib</td>
<td>B interpreter library</td>
</tr>
<tr>
<td>bproc</td>
<td>boot procedure</td>
</tr>
<tr>
<td>brt1, brt2</td>
<td>B start and finish</td>
</tr>
<tr>
<td>f1, f2, f3, f4</td>
<td>Fortran compiler passes</td>
</tr>
<tr>
<td>glob</td>
<td>argument expander</td>
</tr>
<tr>
<td>init</td>
<td>initializer process</td>
</tr>
<tr>
<td>kbd</td>
<td>map of TTY 37 keyboard</td>
</tr>
<tr>
<td>liba</td>
<td>standard assembly-language library</td>
</tr>
<tr>
<td>libb</td>
<td>standard B library</td>
</tr>
<tr>
<td>libf</td>
<td>standard Fortran library</td>
</tr>
<tr>
<td>login, logout</td>
<td>logging on and logging off the system</td>
</tr>
<tr>
<td>msh</td>
<td>mini Shell</td>
</tr>
<tr>
<td>suftab</td>
<td>roff's suffix table</td>
</tr>
<tr>
<td>tabs</td>
<td>set tab stops on typewriter</td>
</tr>
</tbody>
</table>
chmod(I): change access mode of files
sdate(I): adjust date and time
mail(I): send mail to another user
write(I): write to another user
ar(I): a.out(V): assembler and loader output
archive(V): archive (combine) files
glob(VII): archive file
dpt(VI): read DEC ASCII paper tapes
bc(I): convert ASCII to BCD
atof(III): convert ASCII to floating
atoi(III): convert ASCII to integer
ascii(VII): map of ASCII
ctime(III): convert time to ASCII
convert floating to ASCII
itoua(III): convert integer to ASCII
as(II): assembler
liba(VII): standard assembly-language library
libb(VII): standard B library
b(I): compile B program
bstart, bstop(VII): B start and finish
log(III): logarithm base e
bas(II): BASIC dialect
basic(VI): DEC supplied BASIC
cbc(I): convert ASCII to BCD
bc(VII): B compiler
su(I): become super-user
b(I): compile B program
bilib(VII): B interpreter library
cp(BP): compile B program
bppt(V): binary paper tape format
dbppt(II): write binary paper tape
lbppt(II): read binary paper tape
dll(VI): load DEC binary paper tapes
remove symbols, relocation
bits...strip(I): bj(VI): the game of black jack
bj(VI): the game of blackjack
bproc(VII): boot procedure
  boot(I): reboot system
  bppt(V): binary paper tape format
  bproc(VII): boot procedure
break(I): set program
break(I): set program break
brt1,brt2(VII): B start and finish

dc(I): desk calculator
     cal(VI): print calendar
     cal(VI): print calendar
     cmt(I): change access mode of files
cmt(I): change access mode of files
chdir(I): change working directory
chdir(I): change working directory
check(I): check consistency of file system
check(I): check consistency of file system
chess(VI): the game of chess
chess(VI): the game of chess
chmod(I): change access mode of files
chmod(I): change access mode of files
chown(I): change owner of files
chown(I): change owner of files
chown(I): change owner of files
chdir(I): change working directory
chdir(I): change working directory
check(I): check consistency of file system
check(I): check consistency of file system
putc(III): write character or word
get(III): get character
chdir(I): change working directory
chdir(I): change working directory
chdir(I): change working directory
chdir(I): change working directory
check(I): check consistency of file system
check(I): check consistency of file system
chess(VI): the game of chess
chess(VI): the game of chess
chmod(I): change access mode of files
chmod(I): change access mode of files
chown(I): change owner of files
chown(I): change owner of files
chown(I): change owner of files
chdir(I): change working directory
chdir(I): change working directory
check(I): check consistency of file system
check(I): check consistency of file system
putc(III): write character or word
get(III): get character
chdir(I): change working directory
chdir(I): change working directory
chdir(I): change working directory
chdir(I): change working directory
check(I): check consistency of file system
check(I): check consistency of file system
ar(I): archive (combine) files
  sh(I): command interpreter
  cmp(I): compare file contents
  b(I): compile B program
  for(I): compile Fortran program
f1,f2,f3,f4(VII): Fortran compiler passes
bc(VII): B compiler
cat(I): concatenate (or print) files
check(I): check
check(I): check consistency of file system
check(I): check
check(I): check consistency of file system
check(I): check consistency of file system
check(I): check consistency of file system
check(I): check consistency of file system
tty(IV): console typewriter
ls(I): list contents of directory
cmp(I): compare file contents
bcq(I): convert ASCII to BCD
atof(III): convert ASCII to floating
atoi(III): convert ASCII to integer
ftoa(III): convert floating to ASCII
itoa(III): convert integer to ASCII
cftime(III): convert time to ASCII
cp(I): copy file
core(V): core image file
mem(IV): core memory as file
core(V): core image file

sin(III): sine,
cosine
wc(I): get (English) word
count
cp(I): copy file
mkdir(I): create directory
mkdir(II): create directory
creat(II): create file
fork(II): create new process
creat(II): create file
ctime(III): convert time to ASCII
das(VI): disassembler
date(I): get date and time of day
date and time
date modified of file
date(I): get date and time of day
date(I): get date and time of day
db(I): symbolic debugger
dbpt(I): write binary paper tape
dc(I): desk calculator
db(I): symbolic debugger
dcpt(I): write DEC paper tape
dc(I): DEC paper tape
DCsup(B): DECSupplied BASIC
tap0,...,tap7(IV): DECTape file
dtf(I): format
re)w(I): rewind
DECTape
tap(I): manipulate
DECTape
rmdir(I): remove
directory
rm(I): remove
directory
dsw(I): delete files interactively
directory
unlink(II): remove
directory
msg(I): permit or deny messages
switch(III): transfer
depending on value
dc(I): desk calculator
detachable file system
df(I): find free disk space
dialect
directory(v): directory directory
chdir(I): change working directory
chdir(II): change working directory
ls(I): list contents of directory
mkdir(I): create directory
mkdir(II): create directory
rmdir(I): remove (delete) directory
directory
directory(v): directory directory

das(VI): disassembler
rfo(IV): RF disk file
rk0(IV): RK disk file
rk1(I): load disk from tape
df(I): find free disk space
dkd(I): dump disk to tape
du(I): find disk usage
rkn(I): format RK disk

- xi -
umount(II): dismount file system
umount(I): dismount removable file system
dli(VI): load DEC binary paper tapes
dpt(VI): read DEC ASCII paper tapes
dsw(I): delete files interactively
dtf(I): format DECTape
du(I): find disk usage

rkd(I): dump disk to tape
od(I): octal dump of file
ed(I): text editor
ed(I): editor (loader)

ld(I): link
ed(I): text

log(III): logarithm base

cemt(II): catch
wc(I): get

exec(II): execute program file
exec(II): execute program file
exec(II): execution
exit(II): terminate execution

exp(III): exponential function
exp(III): exponential function

exp(III): exponential function
cmp(I): compare
type(I): print
stat(I): get
stat(II): get

file system(V): file system format
check consistency of file system
mkfs(I): initialize file system

mount(I): mount detachable file system
mount(II): mount file system

umount(I): dismount removable file system
umount(II): dismount file system

find(I): find file with given name
pr(I): print file with headings

archive(V): archive file
chmod(II): change mode of file
chown(II): change owner of file
close(II): close open file
core(V): core image file
cp(I): copy file
creat(II): create file
exec(II): execute program file
fstat(II): status of open file

link(II): link to file
ln(I): link to file

mem(IV): core memory as file
mv(I): move or rename file

od(I): octal dump of file
open(II): open file

passwd(V): password file
read(II): read file

rf0(IV): RF disk file
rk0(IV): RK disk file

rm(I): remove (delete) file
files interactively
files
files...cat(I):
files...chmod(I):
files
file...smdate(II):
file
file
file
file
find disk usage
find file with given name
find free disk space
find name of terminal
find read or write pointer
find undefined symbols
find file with given name
finish
floating to ASCII
floating
floating-point simulator
for(I): compile Fortran program
fork(II): create new process
form letter
format DECtape
format RK disk
(format) text
format
format
format
format
Fortran compiler passes
Fortran library
Fortran program
floating-point simulator
free disk space
from tape
status of open file
convert floating to ASCII
function
Fortran compiler passes
game of black jack
game of chess
game of MOO
game of tic-tac-toe
generate form letter
call
get character
date(I):
get date and time of day
wc(I):
get (English) word count
stat(I):
get file status
stat(II):
get file status
tm(I):
get time information
time(II):
get time of year
gtty(II):
get typewriter mode
get user ID
get(III): get character
find(I): find file with
hup(I): hang up typewriter
pr(I): print file with
getuid(II): get user ID
glob(VII): argument expander
getty(II): get typewriter mode
hang up typewriter
headers
hup(I): hang up typewriter
IBM 2741
ID
ID’s
ilgins(II): catch illegal instruction trap
illegal instruction trap
image file
information
information
inhibit interrupts
inhibit quits
initialize file system
initializer process
initializer process
init(VII): initializer process
instruction trap
integer to ASCII
interactively
interpreter library
interpretation
intr(II): catch or inhibit
intr(II): catch or inhibit interrupts
itoo(III): convert integer to ASCII
interrup
itoo(III): convert integer to ASCII
bj(VI): the game of black
kbd(VII): map of TTY 37
kbd(VII): map of TTY 37 keyboard
lbppt(I): read binary paper tape
letter
liba(VII): standard assembly-language
library...liba(VII):
library
libb(VII): standard B library
libf(VII): standard Fortran library
libf(VII): standard Fortran library
ld(I): link editor (loader)
link(II): link to file
ln(I): link to file
ln(I): link to file
ls(I): list contents of directory
logout(VII): logging on and
logged-off user information
logging off the system...login,
login, logout(VII): logging on and logging off the system
log(III): logarithm base e
logging off the system...login,
the system...login,
mail(I): send

tap(I):
uids(V):
ascii(VII):
kbd(VII):
mem(IV): core memory as file

cmsg(I): permit or deny messages
msh(VII): mini Shell

chmod(I): change mode of file
chmod(I): change access mode of files
stty(II): set mode of typewriter

getty(II): get typewriter mode

smdate(II): set date modified of file

moo(VI): the game of MOO

mount(I): mount detachable file system

mv(I): move or rename file

om(I): move read or write pointer

chmod(I): create directory
chown(I): create directory
mkfs(I): initialize file system

find(I): find file with given name
find(I): find file with given name
nm(I): print namelist

fork(II): create new process

fopen(II): status of open file

close(II): close open file

getp att(V): binary paper tape format
dbppt(I): write binary paper tape

lbppt(I): read binary paper tape

ppt(IV): punched paper tape
dli(VI): load DEC binary
dpt(VI): read DEC ASCII
as2(VII): assembler's
Fortran compiler

passwd(V):
  msg(I):
seek(II): move read or write
tell(II): find read or write

cal(VI):
type(I):
pr(I):
cat(I): concatenate (or
   nm(I):
   msg(III):
   ptime(III):
   bproc(VII): boot
   fork(II): create new
   init(VII): initializer
   rele(II): release
   wait(II): wait for
     break(II): set
   exec(II): execute
   b(I): compile B
   for(I): compile Fortran

   ppt(IV):
quit(II): catch or inhibit
lbppt(I):
dpt(VI):
read(II):
seek(II): move
tell(II): find

boot(I): reboot system
rele(II): release processor

strip(I): remove symbols,
tty0,...,tty5(IV):
unmount(I): dismount
rmrdir(I):
rm(I):
unlink(II):
strip(I):
mv(I): move or

rew(I): rewind DECTape
rf0(IV): RF disk file
rk0(IV): RK disk file
rxf(I): format

paper tapes
pass 2
passes...f1,f2,f3,f4(VII):
password(V): password file
password file
permit or deny messages
pointer
pointer
ppt(IV): punched paper tape
pr(I): print file with headings
print calendar
print file on IBM 2741
print file with headings
print (or
print namelist
print string on typewriter
print time
procedure
process
process
processor
program break
program file
program
program
ptime(III): print time
punched paper tape
putc(III): write character or word
quit(II): catch or inhibit quits
quits
read binary paper tape
read DEC ASCII paper tapes
read file
read or write pointer
read or write pointer
read(II): read file
release processor
rele(II): release processor
relocation bits
remote typewriter
removable file system
remove (delete) directory
remove (delete) file
remove symbols, relocation bits
rename file
rew(I): rewind DECTape
RF disk file
RK disk file
RK disk
dump disk to tape

- xvi -
rkl(I): load disk from tape
rk0(IV): RK disk file
rm(I): remove (delete) file
rm(IV): remove (delete) directory
roff(I): run off (format) text
sdate(I): adjust date and time
seek(II): move read or write pointer
mail(I): send mail to another user
smdate(II): set date modified of file
stty(II): set mode of typewriter
break(II): set program break
stime(II): set system time
tabs(II): set tab stops on typewriter
setuid(II): set user ID
msh(II): mini Shell
sh(I): command interpreter
fptrap(III): floating-point simulator
sin(III): sine, cosine
smdate(II): set date modified of file
sort(II): sort a file
df(I): find free disk space
liba(II): standard assembly-language library
libb(II): standard B library
libf(II): standard Fortran library
brl1,brl2(II): B start and finish
fstat(II): get file status
stat(I): get file status
stat(II): get file status
tabs(II): set tab stops on typewriter
mesg(III): print string on typewriter
strip(I): remove symbols, relocation bits
stty(II): set mode of typewriter
suftab(II): roff’s suffix table
su(I): become super-user
sum(I): sum file
su(I): become super-user
basic(VI): DEC supplied BASIC
switch(III): transfer depending on value
db(I): symbolic debugger
strip(I): remove symbols, relocation bits
un(I): find undefined symbols
file system(V): file system format
stime(II): set system time
boot(I): reboot system
check consistency of file system...check(I):
and logging off the
mkfs(I): initialize file
mount I: mount file
dismount removable file
umount(I): dismount file

who(I): who is on the
tabs(VII): set
suftab(VII): roff's suffix

bppt(V): binary paper
dbppt(I): write binary paper
lbppt(I): read binary paper
ppt(IV): punched paper
rkd(I): dump disk to
rkl(I): load disk from
load DEC binary paper
dpt(VI): read DEC ASCII paper

tty(I): find name of
exit(I): exit
ed(I): text editor
roff(I): run off (format)
ttt(VI): the game of
tm(I): get
date(I): get date and
time(I): get
ctime(III): convert
ptime(III): print
sdate(I): adjust date and
stime(II): set system

switch(III):
catch illegal instruction
cemt(I): catch EMT

kbd(VII): map of
gtty(I): get
hup(I): hang up
mesg(III): print string on
stty(I): set mode of
tabs(VII): set tab stops on
tty(I): console
tty0,...,tty5(IV): remote

un(I): find

- xviii -
du(I): find disk
getuid(II): get
setuid(II): set
uids(V): map names to
utmp(V): logged-in
mail(I): send mail to another
write(I): write to another
transfer depending on
    wait(II):
    who(I):
find(I): find file
pr(I): print file
wc(I): get (English)
putc(III): write character or
    chdir(I): change
    chdir(II): change
dbppt(I):
putc(III):
seek(II): move read or
    tell(II): find read or
    write(I):
time(II): get time of
as2(VII): assembler's pass
2
    type(I): print file on IBM
    kbd(VII): map of TTY
    un(I): find undefined symbols
    unlink(II): remove (delete) file
usage
user ID
user ID
user ID's
user information
user
utmp(V): logged-in user information
time...switch(III):
wait for process
wait(II): wait for process
wc(I): get (English) word count
who is on the system
who(I): who is on the system
with given name
with headings
word count
word
working directory
working directory
write binary paper tape
write character or word
write pointer
write pointer
write to another user
write(I): write to another user
year
2
2741
37 keyboard