

15f edition

TABLE OF CONTENTS

Nov 3 1971

I. COMMANDS

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

sum	sum file
tap	manipulate DEctape
tm	get time information
tty	find name of terminal
type	print file on IBM 2741
umount	dismount removable file system
un	find undefined symbols
wc	get (English) word count
who	who is on the system
write	write to another user

II. SYSTEM CALLS

break	set program break
cent	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
gtty	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
rele	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

fptrap	floating-point simulator
ftoa	convert floating to ASCII
get	get character
itoa	convert integer to ASCII
log	logarithm base e
mesg	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
uids	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
dli	load DEC binary paper tapes
dpt	read DEC ASCII paper tapes
mo0	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

bilib B interpreter library
bproc boot procedure
brt1,brt2 B start and finish
f1,f2,f3,f4 Fortran compiler passes
glob argument expander
init initializer process
kbd map of TTY 37 keyboard
liba standard assembly-language library
libb standard B library
libf standard Fortran library
login, logout logging on and logging off the system
msh mini Shell
suftab roff's suffix table
tabs set tab stops on typewriter

INDEX

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
 a.out(V): assembler and loader output
 ar(I): archive (combine) files
 archive(V): archive file
 archive(V): archive file
 glob(VII): argument expander
 ar(I): archive (combine) files
 dpt(VI): read DEC ASCII paper tapes
 bcd(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...ftoa(III):
 itoa(III): convert integer to ASCII
 ascii(VII): map of ASCII
 as(I): assembler
 a.out(V): assembler and loader output
 as(I): assembler
 ba(VII): B assembler
 as2(VII): assembler's pass 2
 liba(VII): standard assembly-language library
 as2(VII): assembler's pass 2
 atof(III): convert ASCII to floating
 atoi(III): convert ASCII to integer
 ba(VII): B assembler
 bc(VII): B compiler
 bilib(VII): B interpreter library
 libb(VII): standard B library
 b(I): compile B program
 brt1,brt2(VII): B start and finish
 log(III): logarithm base e
 bas(I): BASIC dialect
 bas(I): BASIC dialect
 basic(VI): DEC supplied BASIC
 ba(VII): B assembler
 bcd(I): convert ASCII to BCD
 bcd(I): convert ASCII to BCD
 bc(VII): B compiler
 su(I): become super-user
 b(I): compile B program
 bilib(VII): B interpreter library
 bppt(V): binary paper tape format
 dbppt(I): write binary paper tape
 lbppt(I): read binary paper tape
 dli(VI): load DEC binary paper tapes
 remove symbols, relocation bits...strip(I):
 bj(VI): the game of black jack

bj(VI):	the game of	black jack
bproc(VII):		boot procedure
		boot(I): reboot system
		bppt(V): binary paper tape format
		bproc(VII): boot procedure
break(II):	set program	break
		break(II): set program break
		brt1,brt2(VII): B start and finish
		calculator
dc(I):	desk	calendar
cal(VI):	print	cal(VI): print calendar
		cemt(II): catch EMT traps
		ilgins(II): catch illegal instruction trap
		intr(II): catch or inhibit interrupts
		quit(II): catch or inhibit quits
		cat(I): concatenate (or print) files
		cemt(II): catch EMT traps
		chmod(I): change access mode of files
		chmod(II): change mode of file
		chown(II): change owner of file
		chown(I): change owner of files
		chdir(I): change working directory
		chdir(II): change working directory
putc(III):	write	character or word
get(III):	get	character
		chdir(I): change working directory
		chdir(II): 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(II): change mode of file
		chown(I): change owner of files
		chown(II): change owner of file
		close(II): close open file
		close(II): close open file
		cmp(I): compare file contents
		(combine) files
ar(I):	archive	command interpreter
		sh(I):
		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	consistency of file system
		tty(IV): console typewriter
		ls(I): list contents of directory
cmp(I):	compare file	contents
		bcd(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
		ctime(III): convert time to ASCII

cp(I):	copy file
core(V):	core image file
mem(IV):	core memory as file
sin(III):	sine, cosine
wc(I):	get (English) word count
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
sdate(I):	adjust date and time
smdate(II):	set date modified of file
date(I):	get date and time of day
date(I):	get date and time of day
	db(I): symbolic debugger
	dbppt(I): write binary paper tape
	dc(I): desk calculator
db(I):	symbolic debugger
dpt(VI):	read DEC ASCII paper tapes
dli(VI):	load DEC binary paper tapes
basic(VI):	DEC supplied BASIC
tap0,...,tap7(IV):	DECTape file
dtf(I):	format DECTape
rew(I):	rewind DECTape
tap(I):	manipulate DECTape
rmdir(I):	remove (delete) directory
rm(I):	remove (delete) file
dsw(I):	delete files interactively
unlink(II):	remove (delete) file
msg(I):	permit or deny messages
switch(III):	transfer depending on value
dc(I):	desk calculator
mount(I):	mount detachable file system
	df(I): find free disk space
bas(I):	BASIC dialect
directory(V):	directory format
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(V): directory format
das(VI):	disassembler
rf0(IV):	RF disk file
rk0(IV):	RK disk file
rkl(I):	load disk from tape
df(I):	find free disk space
rkd(I):	dump disk to tape
du(I):	find disk usage
rkf(I):	format RK disk

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
ld(I):	link editor (loader)
ed(I):	text editor
log(III):	logarithm base e
cemt(II):	catch EMT traps
wc(I):	get (English) word count
exec(II):	execute program file
exit(II):	terminate execution
glob(VII):	argument expander
exp(III):	exponential function
exp(III):	exponential function
cmp(I):	compare file contents
type(I):	print file on IBM 2741
stat(I):	get file status
stat(II):	get file status
file system(V):	file system format
check consistency of	file system...check(I):
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
file system(V):	file system format
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

dsw(I): delete	files interactively
ar(I): archive (combine)	files
concatenate (or print)	files...cat(I):
change access mode of	files...chmod(I):
chown(I): change owner of	files
set date modified of	file...smdate(II):
sort(VI): sort a	file
sum(I): sum	file
tap0,...,tap7(IV): DECTape	file
unlink(II): remove (delete)	file
du(I):	find disk usage
find(I):	find file with given name
df(I):	find free disk space
tty(I):	find name of terminal
tell(II):	find read or write pointer
un(I):	find undefined symbols
	find(I): find file with given name
brt1,brt2(VII): B start and	finish
ftoa(III): convert	floating to ASCII
atof(III): convert ASCII to	floating
fptrap(III):	floating-point simulator
	for(I): compile Fortran program
	fork(II): create new process
form(I): generate	form letter
dtf(I):	format DECTape
rkf(I):	format RK disk
roff(I): run off	(format) text
bppt(V): binary paper tape	format
directory(V): directory	format
file system(V): file system	format
	form(I): generate form letter
f1,f2,f3,f4(VII):	Fortran compiler passes
libf(VII): standard	Fortran library
for(I): compile	Fortran program
	fptrap(III): floating-point simulator
df(I): find	free disk space
rkl(I): load disk	from tape
	fstat(II): status of open file
	ftoa(III): convert floating to ASCII
exp(III): exponential	function
	f1,f2,f3,f4(VII): Fortran compiler passes
bj(VI): the	game of black jack
chess(VI): the	game of chess
moo(VI): the	game of MOO
ttt(VI): the	game of tic-tac-toe
form(I):	generate form letter
get(III):	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
getuid(II):	get user ID
	get(III): get character

find(I): find file with	getuid(II): get user ID
	given name
	glob(VII): argument expander
	gtty(II): get typewriter mode
hup(I): hang up typewriter	hang up typewriter
pr(I): print file with	headings
	hup(I): hang up typewriter
type(I): print file on	IBM 2741
getuid(II): get user	ID
setuid(II): set user	ID
uids(V): map names to user	ID's
	ilgins(II): catch illegal instruction trap
ilgins(II): catch	illegal instruction trap
core(V): core	image file
tm(I): get time	information
utmp(V): logged-in user	information
intr(II): catch or	inhibit interrupts
quit(II): catch or	inhibit quits
mkfs(I): initialize file system	initialize file system
init(VII): initializer process	initializer process
	init(VII): initializer process
ilgins(II): catch illegal	instruction trap
itoa(III): convert	integer to ASCII
atoi(III): convert ASCII to	integer
dsw(I): delete files	interactively
bilib(VII): B	interpreter library
sh(I): command	interpreter
intr(II): catch or inhibit	interrupts
	intr(II): catch or inhibit interrupts
	itoa(III): convert integer to ASCII
bj(VI): the game of black	jack
	kbd(VII): map of TTY 37 keyboard
kbd(VII): map of TTY 37	keyboard
	lbppt(I): read binary paper tape
	ld(I): link editor (loader)
form(I): generate form	letter
library...	liba(VII): standard assembly-language
	libb(VII): standard B library
	libf(VII): standard Fortran library
bilib(VII): B interpreter	library
standard assembly-language	library...liba(VII):
libb(VII): standard B	library
libf(VII): standard Fortran	library
	ld(I): link editor (loader)
	link(II): link to file
	ln(I): link to file
	link(II): link to file
ls(I): list contents of directory	link(II): link to file
	ln(I): link to file
dli(VI): load DEC binary paper tapes	load DEC binary paper tapes
rkl(I): load disk from tape	load disk from tape
a.out(V): assembler and	loader output
ld(I): link editor	(loader)
log(III): logarithm base e	logarithm base e
utmp(V): logged-in user information	logged-in user information
logout(VII): logging on and	logging off the system...login,

login, logout(VII):	logging on and logging off the system
logging off the system...	log(III): logarithm base e
the system...login,	login, logout(VII): logging on and
	logout(VII): logging on and logging off
mail(I): send	ls(I): list contents of directory
	mail to another user
tap(I):	mail(I): send mail to another user
uids(V):	manipulate DECTape
ascii(VII):	map names to user ID's
kbd(VII):	map of ASCII
	map of TTY 37 keyboard
mem(IV): core	mem(IV): core memory as file
	memory as file
mesg(I): permit or deny	mesg(I): permit or deny messages
msh(VII):	mesg(III): print string on typewriter
	messages
	mini Shell
chmod(II): change	mkdir(I): create directory
chmod(I): change access	mkdir(II): create directory
stty(II): set	mkfs(I): initialize file system
gtty(II): get typewriter	mode of file
smdate(II): set date	mode of files
moo(VI): the game of	mode of typewriter
	mode
	modified of file
	MOO
mount(I):	moo(VI): the game of MOO
mount(II):	mount detachable file system
	mount file system
	mount(I): mount detachable file system
	mount(II): mount file system
mv(I):	move or rename file
seek(II):	move read or write pointer
	msh(VII): mini Shell
	mv(I): move or rename file
tty(I): find	name of terminal
find(I): find file with given	name
nm(I): print	namelist
uids(V): map	names to user ID's
fork(II): create	new process
	nm(I): print namelist.
od(I):	octal dump of file
	od(I): octal dump of file
roff(I): run	off (format) text
close(II): close	open file
fstat(II): status of	open file
open(II):	open file
	open(II): open file
cat(I): concatenate	(or print) files
assembler and loader	output...a.out(V):
chown(II): change	owner of file
chown(I): change	owner of files
bppt(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	paper tapes
dpt(VI): read DEC ASCII	paper tapes
as2(VII): assembler's	pass 2
Fortran compiler	passes...f1,f2,f3,f4(VII):
	passwd(V): password file
passwd(V):	password file
mesg(I):	permit or deny messages
seek(II): move read or write	pointer
tell(II): find read or write	pointer
	ppt(IV): punched paper tape
	pr(I): print file with headings
	print calendar
cal(VI):	print file on IBM 2741
type(I):	print file with headings
pr(I):	print) files
cat(I): concatenate (or	print) files
nm(I):	print namelist
mesg(III):	print string on typewriter
ptime(III):	print time
bproc(VII): boot	procedure
fork(II): create new	process
init(VII): initializer	process
rele(II): release	processor
wait(II): wait for	process
break(II): set	program break
exec(II): execute	program file
b(I): compile B	program
for(I): compile Fortran	program
	ptime(III): print time
	ppt(IV): punched paper tape
	putc(III): write character or word
	quit(II): catch or inhibit quits
quit(II): catch or inhibit	quits
lbppt(I):	read binary paper tape
dpt(VI):	read DEC ASCII paper tapes
read(II):	read file
seek(II): move	read or write pointer
tell(II): find	read or write pointer
	read(II): read file
	reboot system
boot(I):	release processor
rele(II):	rele(II): release processor
	relocation bits
strip(I): remove symbols,	remote typewriter
tty0,...,tty5(IV):	removable file system
umount(I): dismount	remove (delete) directory
rmdir(I):	remove (delete) file
rm(I):	remove (delete) file
unlink(II):	remove symbols, relocation bits
strip(I):	rename file
mv(I): move or	rew(I): rewind DEctape
	rewind DEctape
rew(I):	RF disk file
rf0(IV):	rf0(IV): RF disk file
	RK disk file
rk0(IV):	RK disk
rkf(I): format	rkd(I): dump disk to tape

	rkf(I):	format RK disk
	rk1(I):	load disk from tape
	rk0(IV):	RK disk file
	rmdir(I):	remove (delete) directory
	rm(I):	remove (delete) file
	roff(I):	run off (format) text
suftab(VII):		roff's suffix table
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(VII):		set tab stops on typewriter
setuid(II):		set user ID
	setuid(II):	set user ID
msh(VII):	mini	Shell
fptrap(III):	floating-point	simulator
sin(III):		sine, cosine
	sin(III):	sine, cosine
	smdate(II):	set date modified of file
sort(VI):		sort a file
	sort(VI):	sort a file
df(I):	find free disk	space
liba(VII):		standard assembly-language library
libb(VII):		standard B library
libf(VII):		standard Fortran library
brt1,brt2(VII):	B	start and finish
	stat(I):	get file status
	stat(II):	get file status
fstat(II):		status of open file
stat(I):	get file	status
stat(II):	get file	status
tabs(VII):	set tab	stops on typewriter
mesg(III):	print	string on typewriter
suftab(VII):	roff's	suffix table
	suftab(VII):	roff's suffix table
	su(I):	become super-user
sum(I):		sum file
	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	system...login, logout(VII): logging on
mkfs(I): initialize file	system
mount detachable file	system...mount(I):
mount(II): mount file	system
dismount removable file	system...umount(I):
umount(II): dismount file	system
. file	system(V): file system format
who(I): who is on the	system
tabs(VII): set	tab stops on typewriter
suftab(VII): roff's suffix	table
	tabs(VII): set tab stops on typewriter
bppt(V): binary paper	tape format
dbppt(I): write binary paper	tape
lbppt(I): read binary paper	tape
ppt(IV): punched paper	tape
rkd(I): dump disk to	tape
rkl(I): load disk from	tape
load DEC binary paper	tapes...dli(VI):
dpt(VI): read DEC ASCII paper	tapes
	tap(I): manipulate DECTape
	tap0,...,tap7(IV): DECTape file
	tell(II): find read or write pointer
	terminal
tty(I): find name of	terminate execution
exit(II):	text editor
ed(I):	text
roff(I): run off (format)	text
ttt(VI): the game of	tic-tac-toe
tm(I): get	time information
date(I): get date and	time of day
time(II): get	time of year
ctime(III): convert	time to ASCII
	time(II): get time of year
ptime(III): print	time
sdate(I): adjust date and	time
stime(II): set system	time
	tm(I): get time information
switch(III):	transfer depending on value
catch illegal instruction	trap...ilgins(II):
cemt(II): catch EMT	traps
	ttt(VI): the game of tic-tac-toe
kbd(VII): map of	TTY 37 keyboard
	tty(I): find name of terminal
	tty(IV): console typewriter
	tty0,...,tty5(IV): remote typewriter
	type(I): print file on IBM 2741
	typewriter mode
gtty(II): get	typewriter
hup(I): hang up	typewriter
mesg(III): print string on	typewriter
stty(II): set mode of	typewriter
tabs(VII): set tab stops on	typewriter
tty(IV): console	typewriter
tty0,...,tty5(IV): remote	typewriter
	uids(V): map names to user ID's
	umount(I): dismount removable file system
	umount(II): dismount file system
un(I): find	undefined symbols

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