UNIX* Operating System
Release 2.0.0
System 150, 160, 200
March 30, 1983

NOTE: Read this release notice before attempting to install and use UNIX 2.0. Do not attempt to install the UNIX operating system on any systems that currently have MCS without checking with WICAT Customer Service.

*UNIX is a trademark of Bell Laboratories

1.0 OVERVIEW

UNIX 2.0 contains several new features, including drivers for SMD controllers, and Cipher and DEI tape drives. There is a boot/system load program to allow users to load a System 200 from tape, bad block handling, new utilities and several bug fixes.

A more detailed description of these improvements, instructions for loading from tape and diskette, and information on configuring and using the system is given in this release notice.

1.1 MEMORY REQUIREMENTS

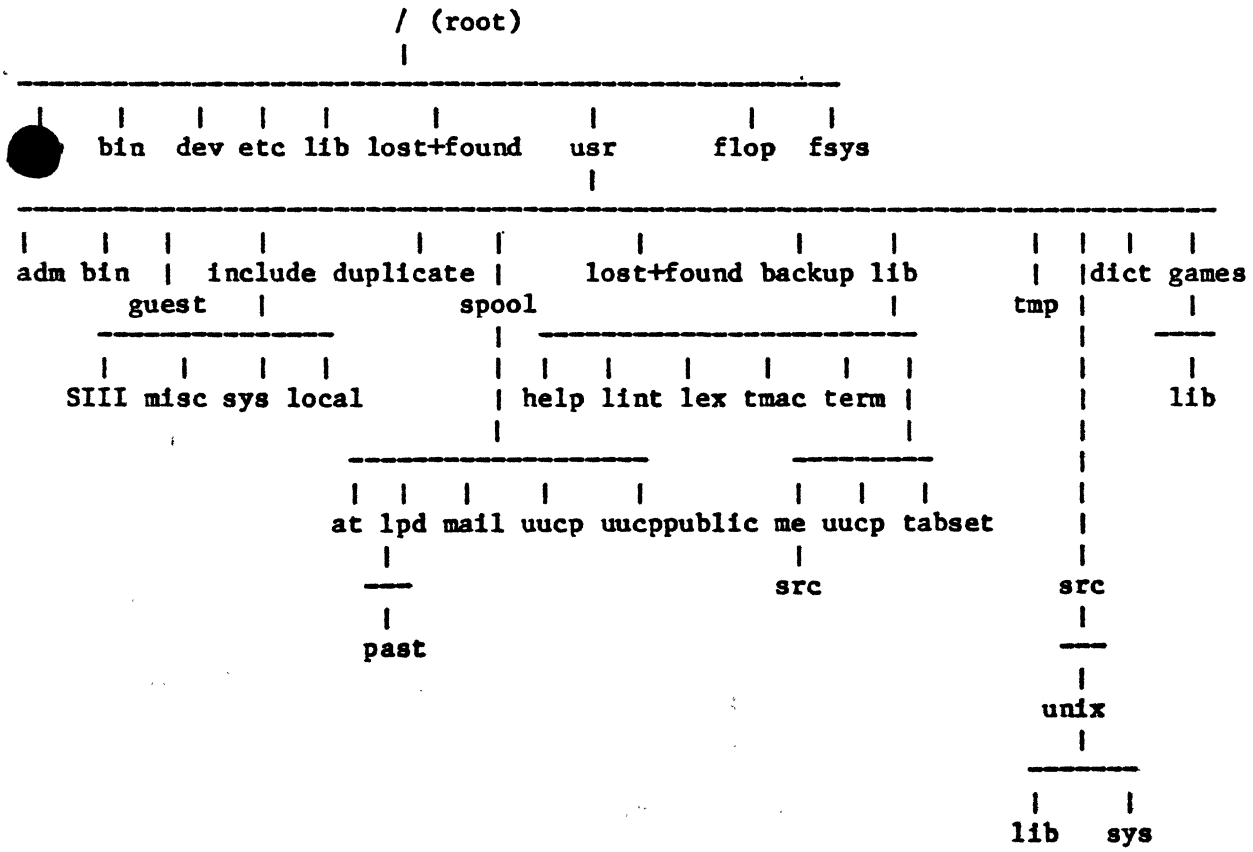
The memory requirements of the UNIX operating system and some often-used utilities are given below. /unix must always remain in core, but the other utilities may be swapped out to disk when there is not enough room in main memory.

process	1st user	each additional user
/unix	180Kb	0Kb
/bin/csh(C-shell)	84Kb	32Kb
/bin/sh(Bourne shell)	48Kb	24Kb
/usr/bin/vi	148Kb	36Kb
/usr/bin/ex	148Kb	36Kb
/usr/bin/nroff	78Kb	23Kb
/lib/ccom(largest part of C compiler)	156Kb	76Kb
/lib/f77pass1(largest part of fortram compiler)	115Kb	50Kb

The stack size for these utilities may vary, which may cause compiling or editing large files to increase the process size somewhat. The UNIX operating system will automatically have other processes, including /etc/update, /etc/cron, and /etc/init executing at most times. These processes will be swapped out to disk when too much main memory is consumed, so they will not significantly add to memory requirements.

2. RELEASE DESCRIPTION

The directory structure for the UNIX operating system is as follows:



Note: Some of these directories are used for floppy diskette backup and may not be on the system 200.

The files included in this release are as follows:

```
/ (root)
.cshrc      .login      bin         etc         fsys        lost+found unix
.exrc       .profile    dev         flop        lib         tmp        usr

./bin:
[       cc      csh     du      grep    ls      nice    ps      sed     sync    true
ar      chmod   date    echo    kill    mail    nm      pwd     sh      tar     wall
as      clear   dd      ed      ld      mkdir   od      rm      size    tee     who
awk     cmp     df      expr    ln      mt      passwd  rmail   stty    test    write
cat     cp      diff    false   login   mv      pr      rmdir   su      time
```

Note: The files in the directory /etc vary according to hardware configuration.

```
./etc:
ac              ddate           fback4          install                         restor
accton          dinit           fback5          install10                       restore
backup          diskformat      fback6          install11       mkboot          sa
badblk          dism            fback7          install12       mkfs            switch
boot            dump            fback8          install2        mklost+found    termcap
change          dumpdir         fback9          install3        mknod           test
checklist       fback1          fsck            install4        motd            ttys
chgrp           fback10         getty           install5        mount           ttytype
chown           fback11         group           install6        mtab            umount
clri            fback12         icheck          install7        ncheck          update
cron            fback2          ident           install8        passwd          utmp
dcheck          fback3          init            install9        rc

./fsys:

  lost+found:
```

Note: The files in the /dev directory vary according to hardware configuration.

```
./dev:
MAKE    null    rwdf0h  rwdw0d  rwdw0h  tty2    wdf0a   wdw0b   wdw0f
console plp     rwdw0a  rwdw0e  swap    tty3    wdf0c   wdw0c   wdw0g
kmem    rwdf0a  rwdw0b  rwdw0f  tty     tty4    wdf0h   wdw0d   wdw0h
mem     rwdf0c  rwdw0c  rwdw0g  tty1    tty5    wdw0a   wdw0e

./flop:

./lib:
c2      cpp     f77pass1 libc.a   libm.a   mcrt0.o  yaccpar
ccom    crt0.o  f77pass2 libl.a   makekey  nlibc.a

./tmp:

./usr:
adm         dict        include     lost+found spool       tmp
bin         guest       lib         src

./usr/adm:
acct        housekeeping wtmp
```

UNIX Filelist (continued)

./usr/dict:
hlista hlistb hstop spellhist words

./usr/include:

0.stdio.h	ctype.h	ident.h	pkh.h	signal.h	utmp.h
SIII	curses.h	lastlog.h	pkp.h	stdio.h	varargs.h
a.out.h	dumpresto.h	math.h	pwd.h	stdio.new	whoami.h
ar.h	dumprestor.h	misc	ranlib.h	sys	
assert.h	errno.h	nlist.h	setjmp.h	sysexits.h	
core.h	grp.h	pagsiz.h	sgtty.h	time.h	

./usr/lost+found:

./usr/spool:
at lpd mail uucp uucppublic

./usr/tmp:

./usr/bin:

adb	crypt	explain	look	p	sccsdiff	tail	uuname
admin	ctags	f77	lorder	page	script	tbl	uux
at	cu	fgrep	lpq	pc	see	tee	val
basename	dc	file	lpr	print	sleep	touch	vc
bc	delta	find	lprm	printenv	sort	tr	vi
bdiff	deroff	fmt	m4	prof	spell	troff	wc
cal	diction	fold	make	prs	spellin	tset	what
calendar	diff3	freq	man	ptx	spellout	tsort	whereis
cb	diffdir	get	mesg	ranlib	split	tty	whoami
cdc	edit	head	mkstr	reset	ssp	ul	xstr
chsh	egrep	help	more	rev	strings	unget	yacc
col	eqn	join	neqn	rewind	strip	uniq	yes
colrm	error	last	newgrp	rmchg	style	units	
comb	ex	lex	nroff	rmdel	sum	uucp	
comm	expand	lint	num	sact	swart	uulog	

./usr/guest:
.cshrc .login .profile

./usr/lib:

Mail.help	dprog	lib300.a	libmath.a	spell
Mail.help.~	eign	lib300s.a	libtermcap.a	stylel
Mail.rc	ex3.6preserve	lib4014.a	libtermlib.a	style2
aliases	ex3.6recover	lib450.a	lint	style3
atrun	ex3.6strings	libF77.a	lintl	tabset
calendar	explain.d	libI77.a	lint2	term
crontab	help	libPW.a	lpd	tmac
deroff	lex	libcurses.a	lpf	unittab
dict.d	lib.b	libl.a	me	uucp

UNIX Filelist (continued)

./usr/src:
unix

./usr/include/SIII:
fatal.h macros.h misc.h

./usr/include/misc:
uparm.h

./usr/include/sys:

acct.h	dei.h	imi.h	mmu.h	pci.h	reg.h	tty.h
altblk.h	dir.h	ino.h	mount.h	pibdefs.h	smd.h	types.h
boot.h	diskform.h	inode.h	mtio.h	pk.h	stat.h	user.h
buf.h	disktune.h	ioctl.h	mx.h	pk.p	systm.h	wdc.h
callo.h	fblk.h	ipe.h	newconf.h	proc.h	text.h	wdf.h
cdp.h	file.h	kprof.h	nproc.h	pwd.h	timeb.h	wdw.h
conf.h	filsys.h	map.h	param.h	px.h	times.h	

./usr/spool/at:
lasttimedone past

./usr/spool/lpd:
tfa15181

./usr/spool/mail:

./usr/spool/uucp:

./usr/spool/uucppublic:

./usr/lib/lint:
llib-lc llib-lc.ln

./usr/lib/tabset:

3101	beehive	std	teleray	xerox1720
aa	diablo	stdcrt	vt100	

./usr/lib/term:

tab300	tab300s	tab450-12	tab5520	tabx1700
tab300S	tab37	tab450-12-8	tab5520-12	
tab300X	tab450	tab450X	tabtn300	

./usr/lib/help:

ad	cb	cmds	de	ge	prs	un	vc
bd	cm	co	default	he	rc	ut	

./usr/lib/tmac:

tmac.a	tmac.an.new	tmac.e	tmac.scover	tmac.srefs
tmac.an	tmac.an6n	tmac.m	tmac.sdisp	tmac.vcat
tmac.an.V7	tmac.an6t	tmac.s	tmac.skeep	

UNIX Filelist (continued)

./usr/lib/lex:
ncform

./usr/lib/me:
acm.me eqn.me index.me revisions tbl.me
chars.me float.me local.me sh.me thesis.me
deltext.me footnote.me null.me src

./usr/lib/uucp:
.XQTDIR L-dialcodes SEQF UUCPLNAME uuclean
L-devices L.sys USERFILE uucico uuxqt

./usr/src/unix:
lib sys

./usr/spool/at/past:

./usr/lib/me/src:
acm.me eqn.me index.me local.me sh.me test.umlaut
chars.me float.me install null.me tbl.me thesis.me
deltext.me footnote.me install.csh revisions test.accent tmac.e

./usr/lib/uucp/.XQTDIR:

./usr/src/unix/lib:
libunix.a

./usr/src/unix/sys:
Makefile c.c unix100.a unix150.a

3.0 INSTALLATION/BACKUP INSTRUCTIONS

3.1 TAPE MEDIA

3.1.1 Installation

Note: Firmware changes and hardware modifications are necessary when the UNIX operating system is loaded on SMD devices that currently have MCS. If you wish to do so, contact WICAT Customer Service for assistance.

The UNIX operating system is factory installed on your WICAT system. Therefore, you will only need to use this installation procedure in the unlikely event that your system disk becomes unreadable. The installation procedure steps you through the load of your system backup tape.

3.1.1.1 SMD Disk / Cipher Tape

a. Insert the tape into the tape drive.
b. Press the reset button.
c. When the following message:

Booting...

appears, depress the space bar several times.
d. The system will display:

Boot failed. Enter boot id (P/S, drive id)

e. Type the following:

s0a0 <return>

f. The tape boot loader will load the boot monitor. When the boot monitor is loaded, it will display the prompt '='.

NOTE: The error 'Spurious interrupt at level 4' may appear on the screen during installation. Ignore this error. However, if any other errors occur, reboot the system and try this procedure again. If the error persists, contact WICAT Customer Service for assistance.

NOTE: The boot monitor will only recognize the following commands:

format	mkfs	icheck
restor	ls	
cat	boot	

These commands are created specifically to work with the boot monitor, and will not work exactly as described in the UNIX Users Manual when they are run from the boot monitor.

NOTE: The boot monitor accepts the backppace key as theddelete key. Use backspace to correct any typing errors.

NOTE: Step g (formatting the disk) is optional for disks that have already been formatted.

UNIX Tape Installation Procedure (continued)

g. Format the Disk by typing:

format <return>

The system will search for the format command on the tape. It will take approximately 30 seconds for the system to find format and load it into memory. When it is loaded, the system will display:

drive:

In response to this prompt, type:

smd0(0h) <return>

If format cannot find a boot block, it will print the message:

no boot block on device
drive type:

If this happens, enter the following:

f80 <return> (If the disk is a Fujitsu 80 Mb)
f160 <return> (If the disk is a Fujitsu 160 Mb)
f474 <return> (If the disk is a Fujitsu 474 Mb)

The system will initialize the disk. It takes approximately 15 minutes to initialize a 80 megabyte disk, 30 minutes to initialize a 160 megabyte disk, and 45 minutes to initialize a 474 megabyte disk. When this step is completed, it will display 'done' and the prompt '='.

h. Make the 'a' filesystem by typing:

mkfs <return>

The system will look for and load the mkfs utility into main memory. When mkfs is finished loading, it will print the message:

fs size:

In response to this prompt, type:

6332 <return>

Mkfs will now display:

fs:

In response to this prompt, type:

smd0(0a) <return>

It should take approximately 4 minutes to create the filesystem. When it is finished, it will print 'done.' and the prompt '='.

UNIX Tape Installation Procedure (continued)

i. Type:

restor <return>

The system will find and load the restor utility. When it is finished loading, restor will display:

input device:

In response to this prompt, type:

cdp0(0e) <return>

The system will search the tape for this file. When it has found it, it will display:

output device:

In response to this prompt, type:

smd0(0a) <return>

This will restore the files on the 'a' filesystem to the disk. It will take approximately 45 minutes. When it is finished, it will display 'input device:' again.

j. Reboot the system from the tape by repeating steps b through f.

k. When the prompt '=' appears, type:

boot <return>

When the boot program is loaded it will display the prompt:

:

In response to this prompt, type:

smd0(0a) <return>

This will load /unix from the disk. When it is finished loading, the standard UNIX prompt '#' will appear, and you will be running off of the disk.

Note: Use the <delete> key to delete characters from the command line when running from /unix.

l. type:

cd /dev <return>

m. Change permissions on the device MAKE files by typing:

```
chmod 755 MAKE <return>
```

This will allow you execute the command files which build nodes for the particular hardware configuration on your system. It should take only a few seconds for this to execute. When it is finished, the prompt '#' should appear.

n. Make the proper filesystems by typing:

```
MAKE 200 <disk_type> ct0 ipe0 std <return>
```

Where disk_type is:

sparrow0	if the disk is a Fujitsu 80 Mb
hawk0	if the disk is a Fujitsu 160 Mb
eagle0	if the disk is a Fujitsu 474 Mb

o. Type the following:

```
mkboot /dev/rsmd0h <drive_type> /etc/boot <return>
```

Where drive_type is:

f80	if the disk is a Fujitsu 80 Mb
f160	if the disk is a Fujitsu 160 Mb
f474	if the disk is a Fujitsu 474 Mb

This will build a boot program for the disk. It should take approximately 4 seconds to run. When it is finished, the prompt '#' will appear.

p. type:

```
sync <return>
```

This will flush the disk buffers. It should only take a few seconds to run. When it is finished the prompt '#' will appear.

q. Type:

```
fsck /dev/smd0a  <return>
```

Fsck checks the consistancy of the file system. If it finds any mistakes, it will prompt for permission to correct the errors. At this time there may be several mistakes that need correcting. Answer yes to all requests. It will take approximately 3 minutes to run fsck. When it is finished, the prompt '#' will appear.

r. If fsck did not make any corrections, type:

```
sync <return>
```

and continue with step q.

NOTE: Fsck may prompt you to reboot the system even if no changes were made. This is due to a bug in fsck. DO NOT reboot the system unless changes were made to the filesystem by fsck.

s. If fsck made corrections to the filesystem, reboot the system immediately by pressing the reset switch. This time the system will boot from the disk instead of the tape. Do not press the space bar while it is booting this time. When the system displays the prompt:

:

type:

<return>

and the system will finish booting. When the system is finished booting, it will display the prompt '#'.

t. Make the 'g' filesystem by typing:

/etc/mkfs /dev/smd0g <size> <return>

where:

size is:	
109504	if the disk is a Fujitsu 80 Mb
240928	if the disk is a Fujitsu 160 Mb
718528	if the disk is a Fujitsu 474 Mb

It should take approximately 4 minutes for mkfs to run. When it is finished, the prompt '#' will appear.

u. Rewind the tape by typing:

mt rew <return>

It may take approximately 20 seconds for the tape to rewind completely. When it is finished, the prompt '#' will appear.

v. The /usr filesystem is located in the sixth file on the tape. Skip the first five files by typing:

mt fsf 5 <return>

It should take approximately 5 minutes for the tape to skip to the correct file. When it is finished, the prompt '#' will appear.

w. Restore the /usr filesystem by typing:

restor rf /dev/rct4 /dev/smd0g <return>

It will take approximately 1 hour for /usr to be restored from the tape. When it is finished, the prompt '#' will appear.

UNIX Tape Installation (Continued)

x. Run fsck on the /usr filesystem by typing:

fsck /dev/smd0g <return>

Answer yes to any corrections fsck may have to make. It will take approximately 8 minutes to run fsck. When it is finished, the prompt '#' will appear.

y. The system is now built. Type:

<ctrl>d

After daemons are started, the login message should appear. log in as

root <return>

to be the superuser. Directions for adding users are found in passwd(5) of the UNIX User's Manual.

3.1.1.2 Winchester Disk / DEI Tape

a. Insert the tape into the tape drive.
b. Press the reset button.
c. When the following message:

```
Booting...
```

appears, depress the space bar several times.
d. The system will display:

```
Boot failed.  Enter boot id (P/S, drive id)
```

e. Type the following:

```
s0b0 <return>
```

f. The tape boot loader will load the boot monitor. When the boot monitor is loaded, it will display the prompt '='.

NOTE: The error 'Spurious interrupt at level 4' may appear on the screen during installation. Ignore this error. However, if any other errors occur, reboot the system and try this procedure again. If the error persists, contact WICAT Customer Service for assistance.

NOTE: The boot monitor will only recognize the following commands:

format	mkfs	icheck
restor	ls	
cat	boot	

These commands are created specifically to work with the boot monitor, and will not work exactly as described in the UNIX Users Manual when they are run from the boot monitor.

NOTE: The boot monitor accepts the backspace key as the delete key. Use backspace to correct any typing errors.

NOTE: Step g (formatting the disk) is optional for disks that have already been formatted.

UNIX Tape Installation Procedure (continued)

g. Format the Disk by typing:

format <return>

The system will search for the format command on the tape. It will take approximately 30 seconds for the system to find format and load it into memory. When it is loaded, the system will display:

drive:

In response to this prompt, type:

wdw0(0h) <return>

If format cannot find a boot block, it will print the message:

no boot block on device
drive type:

If this happens, enter the following:

cmi10	if the disk is a 10 megabyte
cmi15	if the disk is a 15 megabyte

The system will initialize the disk. When this step is completed, it will display 'done' and the prompt '='.

h. Make the 'a' filesystem by typing:

mkfs <return>

The system will look for and load the mkfs utility into main memory. When mkfs is finished loading, it will print the message:

fs size:

In response to this prompt, type:

5950 <return>

Mkfs will now display:

fs:

In response to this prompt, type:

wdw0(0a) <return>

It should take approximately 4 minutes to create the filesystem. When it is finished, it will print 'done.' and the prompt '='.

UNIX Tape Installation Procedure (continued)

i. Type:

restor <return>

The system will find and load the restor utility. When it is finished loading, restor will display:

input device:

In response to this prompt, type:

dei0(0d) <return>

The system will search the tape for this file. When it has found it, it will display:

output device:

In response to this prompt, type:

wdw0(0a) <return>

This will restore the files on the 'a' filesystem to the disk. It will take approximately 45 minutes. When it is finished, it will display 'input device:' again.

j. Reboot the system from the tape by repeating steps b through f.

k. When the prompt '=' appears, type:

boot <return>

When the boot program is loaded it will display the prompt:

:

In response to this prompt, type:

wdc0(0a) <return>

This will load /unix from the disk. When it is finished loading, the standard UNIX prompt '#' will appear, and you will be running off of the disk.

Note: Use the <delete> key to delete characters from the command line when running from /unix.

l. type:

cd /dev <return>

UNIX Tape Installation (continued)

m. Change permissions on the device MAKE files by typing:

chmod 755 MAKE <return>

This will allow you execute the command files which build nodes for the particular hardware configuration on your system. It should take only a few seconds for this to execute. When it is finished, the prompt '#' should appear.

n. Make the proper filesystems by typing:

MAKE 200 <disk_type> ct0 ipe0 std <return>

Where disk_type is:

wdw10.0	if the disk is a 10 megabyte
wdw15.0	if the disk is a 15 megabyte

o. Type the following:

mkboot /dev/rwdw0h <drive_type> /etc/boot <return>

Where drive_type is:

cmi10	if the disk is a 10 megabyte
cmi15	if the disk is a 15 megabyte

This will build a boot program for the disk. It should take approximately 4 seconds to run. When it is finished, the prompt '#' will appear.

p. type:

sync <return>

This will flush the disk buffers. It should only take a few seconds to run. When it is finished the prompt '#' will appear.

q. Type:

fsck /dev/wdw0a <return>

Fsck checks the consistancy of the file system. If it finds any mistakes, it will prompt for permission to correct the errors. At this time there may be several mistakes that need correcting. Answer yes to all requests. It will take approximately 3 minutes to run fsck. When it is finished, the prompt '#' will appear.

r. If fsck did not make any corrections, type:

sync <return>

and continue with step q.

NOTE: Fsck may prompt you to reboot the system even if no changes were made. This is due to a bug in fsck. DO NOT reboot the system unless changes were made to the filesystem by fsck.

s. If fsck made corrections to the filesystem, reboot the system immediately by pressing the reset switch. This time the system will boot from the disk instead of the tape. Do not press the space bar while it is booting this time. When the system displays the prompt:

:

type:

<return>

and the system will finish booting. When the system is finished booting, it will display the prompt '#'.

t. Make the 'g' filesystem by typing:

/etc/mkfs /dev/wdw0g <size> <return>

where:

size is:

10678	if the disk is a 10 megabyte
21082	if the disk is a 15 megabyte

It should take approximately 4 minutes for mkfs to run. When it is finished, the prompt '#' will appear.

u. Remove the boot tape and insert the user tape.

v. Restore the /usr filesystem by typing:

restor rf /dev/rct4 /dev/wdw0g <return>

It will take approximately 1 hour for /usr to be restored from the tape. When it is finished, the prompt '#' will appear.

w. Run fsck on the /usr filesystem by typing:

fsck /dev/wdw0g <return>

Answer yes to any corrections fsck may have to make. It will take approximately 8 minutes to run fsck. When it is finished, the prompt '#' will appear.

x. The system is now built. Type:

<ctrl>d

After daemons are started, the login message should appear. log in as

root <return>

to be the superuser. Directions for adding users are found in passwd(5)oof the UNIX User's Manual.

3.1.2 Backup

The following procedure will step you through backing up UNIX to tape. It will create a bootable tape similar to the tape shipped with the system, but will also include any user files that are on the disk. Since the operating system takes 700 to 900 feet of tape, a 1200 foot or larger tape should be used. User files will require additional space. If the user files take up enough space, an additional tape may be necessary.

a. Become the superuser by using the su(1) command or logging in as 'root'.
b. This will log you in as the super-user.
c. Make sure the write-enable ring is on the tape and load the tape into the drive.
d. Type:

```
cd /etc <return>
```

to change to the /etc directory.
e. Type:

```
MKDT <return>      for cipher tapes
MKCT <return>      for DEI tapes
```

3.1.2 Backup

The following procedure will step you through backing up UNIX to tape. It will create a bootable tape similar to the tape shipped with the system, but will also include any user files that are on the disk. Since the operating system takes 700 to 900 feet of tape, a 1200 foot or larger tape should be used. User files will require additional space. If the user files take up enough space, an additional tape may be necessary.

a. Become the superuser by using the su(1) command or logging in as 'root'.
b. This will log you in as the super-user.
c. Make sure the write-enable ring is on the tape and load the tape into the drive.
d. Type:

```
cd /etc <return>
```

to change to the /etc directory.
e. Type:

```
MKDT <return>      for cipher tapes
MKCT <return>      for DEI tapes
```

This will back up the the system to tape. It takes approximately 12 minutes to back up a system that has no user files.
f. When MKDT is finished executing, the UNIX prompt '#' should appear. UNIX backup is complete. Unload the tape.

Diskette load procedure (continued)

i. Press the reset button. The system will now boot from the winchester disk. When the prompt "#" appears, type:

```
# /etc/install <return>
```

The system will prompt you for the diskette number to install. When this happens type:

```
all <return>
```

and diskettes 2 through 13 will be loaded. When prompted to insert another diskette, insert the diskette and type "<return>". It should take approximately 55 minutes to load these 11 diskettes.

3.2.2 Backup

Backing up UNIX will take 12 diskettes. The following will step you through the backup procedure.

a. Become the superuser by using the su(1) command, or logging in as 'root'.
b. type:

```
/etc/backup <return>
```

to execute the backup command file.
c. The system will ask which diskettes should be backed up. If you want to back up the whole operating system type "all <cr>". When prompted, insert a new diskette and type "<cr>".

4.0 NEW FEATURES

4.1 NEW HARDWARE

UNIX now works on the System 200, and device drivers are now written for cypher and DEI tape drives. The utilities tar, dump, restor, and dd will copy data to and from tape. There is a standalone boot program that works from the tape.

4.1.1 Writing and reading tape

tar - Tar works as documented in the users reference manual. However, the f option and the file /dev/mt4 or /dev/mt0 must be used in reference to the tape. Tar will not default to the tape. It defaults to 512 byte blocks.

dump - Backup will work as documented in dump(8) of the users reference manual. Again, the destination device is /dev/mt4 or /dev/mt0. Use the raw device for block sizes greater than 512 bytes.

restor - Restor will work in the same manner as backup. Before restoring, the tape head must be advanced to the beginning of the restor container file. Mt may be used for this (see section 4.1.3 in this release notice).

4.3.2 Boot tape

The boot tape is divided into a number of physical files. Tapes contain 2 types of records: data blocks and file marks, with 2 consecutive file marks at the end of the logical tape. The following files are necessary to the boot and load procedure:

file	blocksize	description
a	1024	An any OS boot block
b	1024	A pc relative a.out loader (Cipher only)
c	1024	The standalone boot monitor
d	512	A tar image of standalone utilities
e	512	A dump of /
f	512	A dump of /usr

The any OS boot block is a 1k area at the beginning of the tape that contains information the boot ROMs use.

The boot loader is a small (under 4k) relocatable executable file that loads the boot monitor. There are several constraints that the boot ROMs put on this file, including size, sectoring, and format.

The boot monitor allows users to load and execute several standalone utilities which are in tar format and located right after the monitor on the tape.

The stand alone utility programs for the boot monitor are loaded into physical address space above the boot monitor and unix. (The boot monitor and UNIX live in the same space but never at the same time.) The ordinary utilities don't need a start up routine because their _main routines are called as ordinary subroutines from the boot monitor. (And they inherit the boot monitors stack) The utilities are expected to return to the boot monitor just like any other subroutine. The standalone utilities are:

```
format  - for formatting disks
mkfs    - creates filesystems
restor  - restors filesystems from tape container files
boot    - boots UNIX from disk (boot never returns to
          the boot monitor)
ls      - lists  contents of a directory on a filesystemm
cat     - displays contents of ascii files
```

4.3.3 MT utilities

Mt will perform several functions on a tape, including:

```
mt rew <return>
```

to rewind the tape;

```
mt fsf <file_number> <return>
```

to skip file_number files.

4.2 BAD BLOCK HANDLING

There is a utility called badblk which does bad block handling in 2.0 UNIX.

The syntax is:

```
badblk <raw_device_name> [<physical_sector_number>]
```

where <raw_device_name> is raw h filesystem for the disk.
Badblk works in 2 ways:

1. If the name of a raw device is specified, badblk will read every sector on the device and will map good sectors located in a bad block pool to any bad sectors that it finds.

2. If the name of a physical sector on the raw device is specified, the sector will be marked bad and and a good sector will be mapped to it. Physical sector numbers, along with the block numbers, are listed when a read or write error occurs. Block numbers are the offset into the logical disk and can be used to find the file that occupies the sector by typing:

   ```
   icheck -b block_number(s) raw_disk_name <return>
   ```

   This file should then be recopied from backup.

After running badblk, dismount the filesystem and run fsck.

4.2.0 PCI DRIVER

The pci driver has been rewritten to work in tandem mode and at 134.5 baud. The problem with losing characters has been diminished considerably.

4.3.0 NEW/ENHANCED UTILITIES

The following utilities have been added since the last release:

spell	diction	ac	uux
tbl	style	calendar	mt
neqn	explain	fold	
bc	expand	lorder	
chsh	fmt	m4	
colrm	lpr	man	
script	lint	sa	
sccs	uucp	tsort	

The following utilities have been fixed since the last release:

csh	at	fsck
awk	crypt	dd
adb	cu	ps

The following libraries have been added since the last release:

lib.b	libcurses	libl

5.0 NOTES/WARNINGS

5.1 ADDING USERS

1. Become the superuser by logging in as 'root' or using su(1).
2. Edit the file /etc/passwd using the field definitions given in passwd(5) in the UNIX Users Manual.
3. Create the user directory using mkdir. The user directory should be a subdirectory of /usr. A description of mkdir is found in mkdir(1) of the UNIX Users Manual.
4. Change the owner and group of the directory to the username by using the chown and chgrp commands.
5. A login shell file may be created. It must be in the users directory and be named .login if the user is using /bin/csh for the shell and .profile if the user is using /bin/sh for the shell. There are copies of .login and .profile in / and these can be copied to the user directory if desired.
6. Change the owner of the login file using the chown command.

5.2 FILESYSTEMS

Following is a brief description of the 8 logical filesystems on a UNIX disk:

a - root file system.
b - swap area.
c - entire disk minus boot, forwarded bad blocks
d - combination of a + b.
e - possible alternative to user filesystem. The first 30000 bytes of the user filesystem. (only on smd's)
f - possible alternative to user filesystem. The user filesystem minus filesystem e.
g - default user filesystem. Size of e + f.
h - entire physical disk

Note: You cannot have filesystem g on the same disk that has filesystems e and f, since g overlaps e and f.

Note: The address and the size of the filesystems cannot be changed at this time. This means that users creating filesystems on other disks must use the same sizes as already set up on the system disk.

5.3 STARTUP AND SHUTDOWN

5.3.1 Startup

Note: Refer to boot(8) in the Users Reference Manual for a more complete description.

Following is the correct procedure to power up the system.
a. Turn the power on. The mesage:

```
Booting..
Memory test
```

should appear almost immediately.

System Initialization (continued)

b. The system will do some hardware tests and read the boot program from the disk. The boot program resides in sectors 1 to 64 on the primary disk (the any OS boot block is in sector 0). When the boot program is loaded, it displays the message:

```
System loader loaded.
Standalone boot
:
```

Pressing return at this time will cause the system to continue booting normally by loading /unix as the operating system kernel. Another file may be specified as the system kernel. This would be useful for debugging versions on /unix with new or altered device drivers. The kernel file should be in the root (/) directory.

c. The system will display the loading address, the text size and the initialized and uninitialized data sizes of the unix kernel.

d. The UNIX kernel will now begin executing, first figuring and displaying the available amount of memory, then initializing the scheduler (process 0) and starting init (process 1), which starts the superuser shell.

e. In response to the superuser shell prompt, '#', run fsck to check the consistancy of the filesystem. If fsck made any corrections to the filesystem, reboot immediately. Fsck will only change the root filesystem on the disk, not in any buffers, and if the buffers get flushed to the disk, the inconsistancies would appear again. If no corrections were made, rebooting is unnecessary.

f. Set the date at this time. If the date is set when the system is in multiuser mode, the cron program may behave abnormally.

g. Type <ctrl> d. This will execute the shell file /etc/rc. In the standard release version, /etc/rc mounts the user file system and starts cron. Also at this time, init will spawn getty, the login process, for each login port on the system.

h. The system will display:

```
login:
```

at each valid login port. Users may now log in and work normally.

5.3.2 Shutdown

The following procedure should be used by the superuser in shutting down the machine.

a. Kill all of the processes but the single-user shell by typing:

```
kill 1 <return>
```

b. Dismount all mount filesystems.

c. Type:

```
sync <return>
```

This will flush all of the disk buffers.

d. When all disk activity has stopped, power off the machine.

Note: This procedure should be done from the system console after warning all users.

5.4 USING DISKETTES

On the system 150, the standard backup media is 5 1/4 inch diskette. There are several methods in UNIX for accessing diskettes.

5.4.1 Creating Filesystems

This method will allow a floppy to be written to and read as if it were another directory. To use this method, do the following:

1. Type:

   <ctrl>d

   and login as the superuser by typing:

   root <return>

   When the prompt '#' appears, you are logged in.
2. Insert the floppy into the floppy drive. Make sure it is not write-protected.
3. Type:

   diskformat /dev/rwdf0h <return>

   This will format the diskette. It should take approximately 2 minutes to execute. When it is finished, the prompt '#' should return.
4. A filesystem must be made on the diskette. To to this, type:

   mkfs /dev/wdf0a 1116 1 16 <return>

   It should take approximately 30 seconds to run mkfs. When it is finished executing, the prompt '#' should appear.
5. Mount the filesystem by typing:

   mount /dev/wdf0a /flop

The root of the floppy will now be the directory /flop Directories can be created and deleted and files can be read, copied to and from, deleted, and executed on the floppy just by using /flop. Dismount the floppy by typing:

```
cd / <return>
umount /dev/wdf0a <return>
```

Accessing a diskette that has already been created can be done by repeating step 1 to log in as the superuser and step 5 to mount the floppy.

5.4.2 Tar

Tar is a quick and easy method for backing up specific files and directories to the floppy. To use tar, do the following:

1. Insert the floppy into the floppy drive. Make sure it is not write-protected.
2. Type:

   diskformat /dev/rwdf0h <return>

   This will format the diskette. It should take approximately 2 minutes to execute. When it is finished, the prompt '#' should return.
3. Type:

   tar cvf /dev/wdf0c <filelist> <return>

   Where filelist is the list of files to be backed up. Wildcards may be used.

To recopy tar files to the disk, type:

   tar xvf /dev/wdf0c <return>

5.4.3 Dump and Restor

Dump will back up all of the files on a filesystem that have not already been backed up. It is very slow and will take up several diskettes, especially when it is run for the first time. However, it is very thorough and fairly easy to use. There are several options in dump that are not described here. A more complete description is found in dump(1) in the UNIX Users Manual. To use dump, do the following:

1. Type:

   <ctrl>d

   and login as the superuser by typing:

   root <return>

   When the prompt '#' appears, you are logged in.
2. Insert the floppy into the floppy drive. Make sure it is not write-protected.

Dump and Restor (continued)

3. Type:

```
format /dev/rwdf0h <return>
```

This will format the diskette. It should take approximately 2 minutes to execute. When it is finished, the prompt '#' should return.

4. Type:

```
dump fb /dev/wdf0c 1132 <device_name>
```

where device_name is the name of the device to be dumped. It must be either the 'a' or the 'g' device on the winchester disk. This will dump the contents of the filesystem to diskette. It will prompt you from time to time to insert diskettes. When this happens, insert a formatted floppy and press <return>.

If it necessary to use restor, use the directions given in restor(1) of the user's manual.

5.5 CHANGING TIMEZONES

WICAT UNIX systems sent from the factory are set for the Mountain Time Zone. To reconfigure the system for other timezones, do the following:

1. Boot the system normally and login as 'root'.
2. Type the following:

```
adb -w /unix <return>
```

3. The system will return with the message:

```
a.out file = /unix
core not found
ready
```

When this message appears, type the following:

```
timezone?W 0d<minutes> <return>
```

Where minutes is:

300	for Eastern time
360	for Central time
420	for Mountain time
480	for Pacific time

Note: 'timezone' determines the time zone for the system by giving the minutes west of Greenwich Mean Time. It is set at the factory to the value 420.

Note: Make sure that 'timezone' is in lower case. Make sure there are no spaces between 'timezone' and '?', and no spaces between '?' and 'W'. 'W' must be upper case. Made sure there is a space between the 'W' and the '0', and no spaces between '0' and 'd' and the 'd' and the minutes value.

4. Type:

```
<ctrl>d
```

to exit adb.

5. Shut down the system and reboot. When the system comes it should be in the new time zone.

Note: Patching the kernel is dangerous to the system's integerity. Be extremely careful in doing so.

5.6 MFR UTILITY

mfr is a utility provided on the system 150 and 160 which allows UNIX systems to read MCS 5 1/4 diskettes. It will read MCS version 2.3 or later formats. To use mfr, do the following:

a. Log in as 'root', then insert the MCS diskette into the drive.
b. Type the following:

```
mfr /dev/wdf0h <return>
```

mfr will respond with the mfr prompt:

```
next?
```

c. The following commands work with mfr:

<return>	lists the contents of ROOTDIR.DIR.
<directory> <return>	lists the contents of a main directory.
<directory>/<subdir.> <return>	lists the contents of a subdirectory
<directory>/<filename> <return>	causes mfr to display: output file? Type in the name of the destination file and press <return>. The file will be copied from the MCS diskette to the UNIX disk.

Note: On the MCS diskette, upper and lower case letters are mapped to upper case. However, there is no mapping in UNIX.

<ctrl>d	will exit mfr.

Note: There is no entry for mfr in the UNIX documentation. If you have any questions on its use, contact WICAT Customer Service for assistance.

5.7 DEVICE NAMES

The disk device driver names have been changed to a more standard format. The following is the current device name format:

[<r>]<disk_type><drive_number><logical_file_system>

where r (optional) is the raw device.

where disk_type is:
wdw for winchester disks
wdf for 5 1/4 inch floppies
smd for smd disks

where drive_number is an integer in the range 0 - 3 denoting which physical disk drive is to be selected.

where logical_file_system is a letter in the range 'a' - 'h' denoting which logical filesystem on the disk is to be selected.

Examples:

wdw0a	the 'a' filesystem on winchester drive #0;
smd0h	the entire smd disk
rwdf0a	the raw 'a' filesystem on the floppy.

The device names do not take disk size into account. The minor device number is what the system uses to determine device size. MAKE will create device nodes tailored for your system. A complete description of MAKE is found in MAKE(8) in the UNIX Users Manual.

FILE: _DC0A0/USRS.AL.PORT/UNX.RLS.2

3 OTHER NOTES/PROBLEMS

1. The sccs command "val" is not included. As a consequence, the command:

   admin -h <filelist> <return>

   does not work correctly.

2. Graph, plot and -lplot are not available yet.

3. System III include files were included to compile sccs.

4. Lpr sometimes core dumps if the passwd file is not proper.

5. UNIX sometimes gets spurious interrupts on level 4 when using the tape drive. This is a hardware problem and does not affect the data being transferred.

6. A process will occasionally go to sleep for about 30 seconds due to a bug in the scheduler. It does not occur often and does not affect anything else.

7. There is a problem in the tape controller board that causes the system to respond incorrectly when writing to a write-protected tape.

8. 512Kb boards do not work with UNIX on the System 200. At the present time, only 1Mb boards will work on the System 200.

9. There are still several problems with pcis. On the system 200, since there are ports that use the IPE (intelligent port expander) board, these pci ports should generally not be used. These ports are /dev/tty1, /dev/tty2, and /dev/tty3.

10. A new version of the user's manual should be delivered with this release. It should more closly match the UNIX operating system on WICAT hardware.

11. Because of the design of the pci interrupt handler, the integerity of higher-numbered ports is considerably less than lower-numbered ports. This problem is amplified on the systems which have 12 pci's. If you have such a system, do not use ports 6 - 12 for input, since a significant portion of the data read by these ports will be lost. On all systems using pcis, increased efficiency will result from using lower-numbered ports as login ports, and higher-numbered ports for printers and other output-only devices. This problem does not exist on system 200's and 160's that use the IPE board, since IPE boards have on-board processors and memory.

3.2 DISKETTE MEDIA

UNIX is factory installed on your WICAT system. Therefore, you will only need to use this installation procedure in the unlikely event that your system disk becomes unreadable. The installation procedure steps you through the load of your system backup diskettes (see section 3.2.2)

3.2.1 Installation

a. Make sure disk 1 is not write-protected, then insert it into the floppy drive.
b. Press the reset button (behind the brightness control).
c. When the following message:

```
Booting...
```

appears, depress the space bar several times.
d. The system will display:

```
Boot failed.  Enter boot id (P/S, drive id)
```

e. Type the following:

```
s0b0 <return>
```

The system will now boot from the diskette drive rather than the winchester drive.
f. The system has finished booting when the prompt '#' appears.

Note: Step g will destroy all user files. If you do not want to do this, skip step g and go on to step h.

Note: If you are updating from version 1.0 to 2.0, or converting from MCS, or loading an uninitialized system, you must do step g because of filesystem changes due to bad block mapping in the disk driver. Make sure to back up all user files to floppy before starting this step.

g. To initialize the winchester, type:

```
/etc/dinit <return>
```

This will format the winchester and create the necessary filesystems. When the dinit command file is finished, the prompt '#' will appear. It should take approximately 2 minutes to run dinit.
h. Enter the following:

```
# /etc/restore <return>
```

Enough of UNIX to boot from the winchester will be loaded from diskette 1. When the UNIX prompt '#' appears on the screen, installation of disk 1 is complete.