General BSD Notes

The home directory for the user named 'root' is now called /root instead of /.

General Notes

The (networking) *route* command has a somewhat different syntax than older versions with which you may be familiar. To add a network, use the syntax:

```
# route add -net network host
```

See the new manual page for more details.

The compiler defines `__bsdi__` as a constant that indicates that you are using our compilers. The compiler also defines `__i386__` to indicate that code is being generated for the 386 architecture. These can be used to `#ifdef` code specific to our operating system or the 386 architecture.

A small number of sites have had trouble NFS mounting files back and forth between Suns and BSDI machines. This is due to Sun machines checking to insure that the port number is within a certain range. Use the *mount* option resvport either in /etc/fstab or use -o resvport when you invoke *mount*.

To serve DOS PC/NFS clients, use /usr/contrib/bin/pcnfsd and specify the '-n' option to mountd in /etc/rc.

The terminal type for the console terminal emulator is 'ibmpc3'.

The hardware clock is kept in GMT so DOS (if you use it) will see times and dates as GMT, not local time. You can reconfigure and recompile the kernel if you wish to keep the clock in some other timezone.

Filesystem Hierarchy

This section shows the layout of the entire BSD/386 system file hierarchy.

/	root directory of the filesystem
/bin/	user utilities fundamental to both single-user and multi-user environments
/bsd	the kernel (the operating system itself; loaded into memory at boot time)
/dev/	block and character device files
/dev/MAKEDEV	script for creating device files; see MAKEDEV(8)
/dev/fd/	file descriptor files; see fd(4)
/etc/	system configuration files and scripts
/etc/localtime	local timezone information; see ctime(3)
/etc/disklabels/	backup disklabels; see disklabel(8)
/etc/kerberosIV/	configuration files for the kerberos version IV; see kerberos(1)
/etc/mtree/	mtree configuration files; see mtree(1)
/etc/namedb/	named configuration files; see named(8)

/etc/uucp/	UUCP configuration files
/mnt/	empty directory commonly used by system administrators as a temporary mount point
/sbin/	system programs and administration utilities fundamental to both single-user and multi-user environments
/stand/	programs used in a standalone environment
/tmp/	temporary files, usually a mfs(8) memory-based filesystem (the contents of /tmp are usually NOT preserved across a system re- boot)
/usr/	contains the majority of user utilities and applications
/usr/bin/	common utilities, programming tools, and applications
/usr/contrib/	large packages maintained by outside parties
/usr/games/	useful and semi-frivolous programs
/usr/include/	standard C include files
/usr/lib/	programming libraries
/usr/libdata/	misc. utility data files
/usr/libexec/	system daemons & system utilities (executed by other programs)
/usr/local/	local executables, libraries, etc.
/usr/obj/	architecture-specific target tree produced by building the /usr/src tree
/usr/old/	programs from past lives of BSD which may disappear in future releases
/usr/sbin/	system daemons & system utilities (executed by users)
/usr/share/	architecture-independent ascii text files
/usr/share/calendar/	a variety of pre-fab calendar files; see calendar(1)
/usr/share/dict/	word lists & reference databases
/usr/share/doc/	misc documentation; src for most of the printed BSD manuals (printed copies available from the USENIX association)
/usr/share/games/	ascii text files used by various games
/usr/share/man/	manual pages
/usr/share/me/	macros for use with the me macro package
/usr/share/misc/	misc system-wide ascii text files
/usr/share/misc/termcap	terminal characteristics database; see termcap(5)
/usr/share/misc/mk/	templates for make; see make(1)
/usr/share/misc/ms/	macros for use with the ms macro package
/usr/share/misc/tmac/	text processing macros; see nroff(1) and troff(1)
/usr/share/misc/zoneinfo/	timezone configuration information; see tzfile(5)
/usr/src/	BSD and/or local source files
/usr/X11/	X window system
/var/	multi-purpose log, temporary, transient, and spool files

/var/account/	system accounting files
/var/at/	timed command scheduling files; see at(1)
/var/backups/	misc. backup files
/var/db/	misc. automatically generated system-specific database files
/var/games/	misc. game status and log files
/var/log/	misc. system log files
/var/log/messages	system messages
/var/log/wtmp	login/logout log; see wtmp(5)
/var/mail/	user mailbox files
/var/preserve/	temporary home of files preserved after an accidental death of an editor; see ex(1)
/var/quotas/	filesystem quota information files
/var/run/	system information files describing various info about system since it was booted
/var/run/utmp	database of current users; see utmp(5)
/var/rwho/	rwho data files; see rwhod(8), rwho(1), and ruptime(1)
/var/spool/	misc. printer and mail system spooling directories
/var/spool/ftp/	commonly ~ftp; the anonymous ftp root directory
/var/spool/mqueue/	undelivered mail queue; see sendmail(8)
/var/spool/output/	line printer spooling directories
/var/spool/uucp/	UUCP spool directory
/var/spool/uucppublic/	commonly ~uucp; public UUCP temporary directory
/var/tmp/	temporary files that are kept between system reboots

ISODE

ISODE is a freely redistributable implementation of various pieces of the OSI protocol suite. ISODE V8.0 is installed in the /usr/contrib/isode directory and its subdirectories bin, sbin, etc, man, log, include, lib, and lint. The ISODE documentation is on the source media in: /usr/src/contrib/isode/doc/*.

Configuring UUCP

UUCP is a suite of programs which use ordinary serial lines (potentially with modems) to send files and other information from the BSD/386 system another computer running UUCP. UUCP can be used to join the networked mail and news community.

UUCP to SCO requires settings of 8-N-1 to login.

Configuration

UUCP has several configuration files, which are found in the /etc/uucp directory:

- /etc/uucp/CONFIG
- /etc/uucp/L-devices


```
# information for accounts (used by login(1))
touch /var/log/lastlog
chown root.utmp /var/log/lastlog
chmod 664 /var/log/lastlog
```

Once you've customized, type:

```
# sync
# sync
```

then remove the floppy and reboot. You should come up into multi-user mode.

Multi-User Mode

Root has no password so you should be able to login as root.

→ You should set the root password now ←

Use *passwd* to set the root password.

A Note About Multi-User Booting

One note about multi-user boots. If you pop the floppy out and boot, the system will come up multi-user without pausing to ask which kernel to boot. If you need a boot prompt, just hit any key while the default kernel is loading and the boot program will abort the load and prompt you. A system booted by typing to the boot prompt will come up single user, so this is the standard way of getting a single user shell.

All: Loading the DES/Kerberos Floppy

Sites in US and Canada receive extra floppies with the DES-enabled Kerberos binaries and source. Unless you wish to run Kerberos, you need not load these disks. All software runs fine without Kerberos installed.

To install software from the floppy, insert the floppy into the floppy drive. You must have booted from the hard-disk. As root, mount the floppy on /a:

```
# mount -o ro /dev/fd0a /a
```

Run *installsw* and inform it that you are installing from a floppy and that the floppy is mounted in '/a'. The root of the hierarchy being installed is typically '/' when installing packages after a system is up and running on the hard-disk.

Choose the packages you wish to load. Load them.

Configuring Kerberos is beyond the scope of this installation guide.

```
# pwd_mkdb -p /etc/master.passwd
```

will rebuild the password database and create a backward-compatible /etc/passwd. The *vipw* program runs *pwd_mkdb* automatically. Use this *awk* script to convert an old format password file from some other system to the new format:

```
$ cat pwconvert
#!/usr/bin/awk -f
BEGIN { FS = ":" ; OFS = ":" }
{ print $1, $2, $3, $4, "", 0, 0, $5, $6, $7 }
```

IN GENERAL, USE THE vipw PROGRAM TO EDIT THE master.passwd FILE; it automatically runs pwd_mkdb.

/etc/group
: Nothing tricky here except adding your name to the wheel group so you can use the *su* command to become superuser. The 'dialers' group controls which users may use the dial-out modem.

/etc/hosts
: Any hosts you wish to access without having to to use named or other name resolution schemes should be here.

/etc/networks
: Not much customization is required here.

/etc/resolv.conf
: Only if you intend to use bind – this is an optional feature.

/etc/named.boot, /etc/namedb/*: BIND nameserver configuration
: Only if you intend to use bind – this is an optional feature.

/etc/sendmail.cf, /etc/aliases: mailer configuration
: These files have been engineered to work generically, for the most part. No particular customization is necessary. Be sure to kill the old *sendmail* and restart it.

/etc/ntp.conf, /etc/rc.local: clock synchronization
: If you intend to use *ntp* – this is an optional feature.

/etc/crontab: the crontab for root (see
: *cron*(8)) Any customization you need for root's crontab goes here.

/etc/exports
: Only for NFS. See the exports man page.

/etc/rc.local
: Uncomment *ntpd* if you're using it. Uncomment the 'stty -D' line if you wish the default teletype modes to be 8-bit clean.

Here are some other items to customize:

```
# wtmp -- login trace records (used by last(1))
touch /var/log/wtmp
chown root.utmp /var/log/wtmp
chmod 664 /var/log/wtmp
# lastlog -- last login time/date
```


- /etc/uucp/L.sys
- /etc/uucp/L.cmds
- /etc/uucp/USERFILE

The installed files contain sample data that is probably similar to what you'll want. The best reference to installation is the O'Reilly & Associates manual on UUCP "Managing UUCP and Usenet". Contact them in the U.S. at 800-338-6887 (from overseas at +1 707 829 0515) or via email at nuts@ora.com.

Notes on UUCP

Using 'make install' does not overwrite existing /etc/uucp/* files.

An experimental "hayesv" dialer is included in addition to "hayes" (try it if you have a Telebit modem).

Problem Isolation

Check the output from "uuparams -a" to make sure that all the necessary directories have been created (like those for LOGDIR, SPOOLDIR, and LOCKDIR) and that your NODENAME is correct. Use "/usr/libexec/uucico -Ssystemname -x9" to debug the connection phase. Use tip to dial the system manually to make sure that everything works.

Running UUCP

Once you have everything working you will probably want to set up /etc/crontab to run UUCP so that everything is fairly automatic. For example:

```
# once an hour check for any UUCP jobs
49 * * * * uucp /usr/sbin/uusched
```

The simple-minded example above will just get you started. It dials at 49 minutes after the hour, a time less likely to used by other sites than, say, exactly on the hour.

Bidirectional Modem Support

See the section on bidirectional modems elsewhere in this document.

Other Documentation

Many publishers produce books applicable to BSD/386. For example, many of the books in the Nutshell series by O'Reilly & Associates are useful. Contact them in the U.S. at 800-338-6887 (from overseas at +1 707 829 0515) or via email at nuts@ora.com.

Recent Research

The Winter 1993 USENIX conference proceedings contain several technical papers about BSD systems:

"A Tree-Based Packet Routing Table for Berkeley UNIX", Keith Sklower, UCB. p. 93.

"Lessons Learned Tuning the 4.3BSD Reno Implementation of the NFS

Protocol" Rick Macklem, U. of Guelph. p. 53.
"The BSD Packet Filter: A New Architecture for User-level Packet Capture", McCanne and Van Jacobson. p. 259.
"Hardware Profiling of Kernels", Andrew McRae, p. 375.

Selecting Tape Densities

Selecting SCSI tape density has been a mystery. Here's some information on it and how it applies to the SCSI tape driver on BSD/386.

At the most technical level, the SCSI-1 standard defines a MODE SELECT command which can be used to select tape density. The first 1-byte field in MODE SELECT's 'block descriptor' contains a density code. The SCSI-1 standard recognizes a number of density codes, and the SCSI-2 standard provides even more. 'Density code' was not a good choice of terminology, since it really describes tape formats as disparate as 1/2-inch NRZI tapes and 8mm videotape, but we're stuck with it. For quarter-inch cartridge (QIC) formats, there are density codes for QIC-11, QIC-24, QIC-120, QIC-150 and more.

But, there are some obstacles to handling multiple QIC tape formats. Many drives will read multiple formats but can write just one format. If your drive doesn't support MODE SELECT (an 'optional' SCSI command), then you can only write one format. Some drives don't support the corresponding status command ('MODE SENSE'); we currently don't know of any that support MODE SELECT for density selection but fail to support MODE SENSE. The driver assumes that no MODE SENSE implies no MODE SELECT. At least one drive appears to support MODE SENSE but not MODE SELECT; the BSD/386 driver has a special case for this drive (Wangtek).

If your drive supports MODE SELECT, you still may not be able to write some formats. For example, an Maynard/Archive Viper 150 will read QIC-11, QIC-24, QIC-120 and QIC-150, but will only write QIC-24, QIC-120 and QIC-150. This drive is sensitive to the cartridge type; it can write QIC-24 on DC300XLP, DC600A or DC600XTD, but it will write QIC-120 or QIC-150 only on the DC600A and DC600XTD tapes. Note that different tape manufacturers use different tape designations but the cartons will usually tell you which standard types they're 'compatible with'. You will need to check the programming manual for your particular drive to find out if it supports MODE SELECT and what tape formats it can read and write.

The st ('SCSI Tape') driver has a simple way of dealing with SCSI density codes. The high 5 bits of the minor device number encode the low 5 bits of the SCSI tape density. A density code of 0 is the default density, used by /dev/rst*. The system also provides other, easier to remember device names: /dev/rst0_f8mm, /dev/rst0_q11, /dev/rst0_q120, /dev/rst0_q150, /dev/rst0_q24, and /dev/rst0_q525.

depending on the number of packages requested and on the speed of the distribution device and disks.

installsw reports its actions in voluminous detail.

All: Customization

Before multi-user boot, some files must be edited. Without rebooting, change the root to the hard-disk:

\# ***/a/usr/sbin/chroot /a***
\# ***. /.profile*** *read correct paths for execution*
(the command is indeed the 'dot' command)

Most folks have a long list of things they like to customize; here are a few that we suggest:

/etc/fstab: set up your filesystems

\# ***cp /etc/fstab.sample /etc/fstab***

Use your editor to set up your disks in formats like the lines already in that file. YOU MUST UPDATE fstab BEFORE RE-BOOTING.

/etc/netstart: configure

The /etc/netstart file has several lines that may need configuration. Read it through, potentially changing or uncommenting lines as appropriate. Setting the hostname is required for many utilities (e.g., *sendmail*).

/etc/localtime, /usr/share/zoneinfo: set up your time zone

Use

\# ***ls -R /usr/share/zoneinfo***

to find your timezone name (probably in /usr/share/zoneinfo/US/*). Then copy your timezone file to /etc/localtime. Here's an example for MST:

\# ***cp /usr/share/zoneinfo/US/Mountain ***
/etc/localtime

NOTE: BSD/386 reads the hardware clock to set its initial idea of time; the kernel uses the timezone configured in the kernel only to interpret the hardware clock. The hardware clock is normally kept in GMT and is set when the *date*(1) command is used to set the computer's idea of the current time. This is a bit of a conflict with MS/DOS, which likes the hardware clock to be kept in local time. See the "Problem Resolution" section.

The zoneinfo directory also has the 'posix' and 'right' subdirectories which contain the 'standard' POSIX timezone info (without leap seconds) and the 'right way to do it' (with leap seconds) respectively. The 'zoneinfo' directory itself uses the 'posix' scheme.

/etc/master.passwd

Add accounts to the system by cloning other accounts in the /etc/master.passwd file. The format of master.passwd differs slightly from the original 4.3BSD /etc/passwd format of years ago. Executing

If a remote drive is specified, *installsw* asks:

```
Hostname of remote host?:
```

Enter the hostname exactly as it is specified in /etc/hosts. If the hostname does not appear in /etc/hosts, you will have to leave *installsw* to enter it.

If installing remotely or locally from CD-ROM or floppy, the next question is:

```
What directory is it mounted on? [/]:
```

Enter the remote or local directory name upon which the media is mounted (e.g., / or /cdrom).

For remote CD-ROM installation, you must tell *installsw* what name-mapping is required to transform the CD-ROM names (e.g., PKG/NAME.;1) to BSD/386 filenames (e.g., pkg/name). The first choice (direct mapping) is correct for systems which support the Rock Ridge filesystem extensions. The other choices are other possibilities for ISO9660 mappings; log into the remote system and look at the files mounted from the CD-ROM to get an idea of how it presents the file names. You may have to use trial and error to find the correct mapping scheme.

For tape install, *installsw* asks:

```
Name of the no-rewind tape device
        (e.g., /dev/nrst0 or /dev/nrwt0)?:
```

Enter a no-rewind tape device (i.e., first letter of the device name is n). The device '/dev/nrst0' is a popular one for SCSI tapes; likewise '/dev/nrwt0' for ISA-based tapes.

Once *installsw* verifies the device is available, it reads the manifest from the media and asks one more question:

```
What is the root of the install tree? [/a]:
```

Answer this one with the name of the filesystem which is the root of those files to be installed ('/a' during initial installation).

installsw then displays a full-screen display of all available packages, their sizes, and their status (i.e., already loaded, chosen to be loaded). Use letter keys to choose those packages which should be loaded (or not loaded). All keys on this screen are 'hot keys'; it is not necessary to type return. The screen legend explains each of the status indicators. Use the number keys to select all packages from a particular category or even all packages from the distribution medium.

The box in the lower right-hand corner tracks total disk space required for the selected packages. Because packages store data on both the root and on other filesystems, this number doesn't say exactly how much space is required on any specific filesystem. If '/usr' is the only other filesystem, it will require just a bit less space than shown in the box.

When all the desired packages have been selected, use the 'Q' key to signal *installsw* to proceed. The list of packages to be loaded is displayed; verify it to begin actual loading. Loading takes from 30-90 minutes,

Network Configuration

Basic Setup

Once configured, BSD/386 network operation is initiated from the /etc/netstart file. This file is just a shell script that is run from /etc/rc at system boot time.

Three basic things happen in /etc/netstart:

- The system hostname is set
- Network interfaces are configured
- Routing is started

Routing is probably the most complicated aspect. Most network hosts don't have to worry about routing much, typically you just point your "default" route to a local gateway.

You will need to get some information from your local site administrator before you can configure your network. Things you will need to know before you get started are:

1. Your machine's hostname. If possible this should be the Fully Qualified Domain Name (austin.bsdi.com). There have actually been whole documents written on just choosing a name. If you are at a large site you will be constrained by local policy.
2. Your IP Address (should look something like 137.39.95.2)
3. Your networks netmask (examples are 0xffffff00 or 255.255.255.0)
4. The IP Address of your local gateway system. This is also called the 'default' route. If you are not connected to a larger network then you will not have one of these.
5. The BSD/386 name of the interface device you are using (e.g., we0 for WD8003/8013 & 3COM 3C503). You can get this information at system boot time; /etc/netstart contains a list of devices in its comments.

If you are connecting to a network via some kind of point-to-point protocol (HDLC, PPP, SLIP), then you will also need to know the IP Address of your service provider (which will almost certainly be the answer to #4).

The first thing you'll see in /etc/netstart is a set of options like "timedflag=NO". To get started, you shouldn't need to worry about them, they are harmless as shipped.

The next thing to do is set the hostname. First, set the shell variable 'hostname' to your hostname (since it's used in several places this makes life easier if it must be changed). The script then runs the "hostname" command to set the system hostname (using $hostname).

Next examine the commented out "ifconfig" lines. These are examples to show you the various network device names and different styles of ifconfig commands. You need to create an ifconfig line for your device and insert it following the examples provided. For ethernet devices, the 'link0' flag almost always enables the BNC connector on devices so equipped. Read the section 4 man page on your device to find out more (e.g., man 4 ef).

The rn device (RISCom/N1 HDLC) is a point-to-point link and therefore requires two hostnames (or IP Addresses). The first hostname specifies your end of the network connection while the second is the remote end's name or IP Address. There is also a sample SLIP configuration for sl0 and it works the same way but has some additional link flags that might be used.

Next, there are a two standard lines:

```
ifconfig lo0 inet localhost
route add $hostname localhost
```

These configure the loopback interface and route packets destined for your machine through it. This just makes talking to yourself a little bit faster. Do not modify these. If you get errors during boot about "localhost: bad value" or "localhost: not found" then you haven't set your name resolver up correctly (contact your Domain administrator for help).

Finally, you must configure your routing. The sample /etc/netstart file contains a commented out line:

```
route add default __gateway__
```

Replace the ***__gateway__*** part with the IP Address of your site's default router. Default routing takes place when you have a packet destined for an unknown network. In that case, the packet is simply forwarded along to the default route for proper delivery.

Once you configure /etc/netstart, you can start the network by hand:

```
# sh /etc/netstart
```

Once the network is started, you'll need to make sure it works, try to "ping" some other host on the network. See "Debugging Network Problems" if things aren't working at this point. The system will automatically run /etc/netstart for you every time it boots.

Gateways

A gateway knows how to route packets from one network to another. If you are configuring your BSD/386 system as a gateway you will need to rebuild the kernel with the GATEWAY option enabled. Once you install the new kernel and reboot, you can make sure everything is ok by typing:

```
# bpatch ipforwarding
```

and make sure you get the result:

```
0x1 = 1
```

If you don't, then something went wrong. Maybe you ran "config" on the wrong file or copied the new kernel into the wrong place.

Routing is more complicated on gateways and is beyond the scope of this document.

would be called '/a/usr'). Then you must *mount* the filesystems. Directions follow.

Mount the hard-disk root on '/a':

```
# mount /dev/wd0a /a          sd0a for SCSI
```

Make the '/usr' directory on the hard-disk root and mount the other BSD/386 partition there (usually the 'h' partition, sd0h or wd0h):

```
# mkdir /a/usr
# mount /dev/wd0h /a/usr      sd0h for SCSI
```

If you plan on using a separate partition for /var, mount the /var partition onto /a/var:

```
# mkdir /a/var
# mount /dev/wdXX /a/var      sdXX for SCSI
```

where *XX* is the appropriate partition. Only some sites do this. If you do not do this, the installation program *installsw* enables you to install the /var directory either as part of the root or on some other BSD/386 partition (potentially combined with other parts of the hierarchy, e.g., /usr).

Be sure to mimic the above procedures for any other filesystems upon which you intend to load system software (a fairly rare event).

All: Load the Distribution

Run *installsw*:

```
# installsw
```

or

```
# installsw -l tapeuser     (dash-ell)
```

installsw asks a few questions about the environment and then moves to a full-screen interactive display of available packages. Select packages you wish to install. When finished, the 'Q' key instructs *installsw* to proceed with package installation.

All questions asked by *installsw* are shown as a question, a default (between square brackets), and then a colon. Replies are entered on the same line as the question. If you type a return (with no other data entered), the default is used.

The next question asks which medium will be used (some questions are shown on multiple lines to enhance the presentation):

```
Are you installing from tape, floppy,
                    or CD-ROM? [cdrom]:
```

Valid answers are: 'floppy', 'tape', and 'cdrom'. Type the correct answer.

Then:

```
Is the distribution volume local or remote? [local]:
```

Valid replies are *local* and *remote*. Enter *remote* if some other machine on the local area network hosts the tape drive or CD-ROM drive. Reply with the answer.

 # ping *192.0.2.87* *Substitute valid remote host address*

Remote Tape or Remote CD-ROM: Gateways

If using a gateway, you will need to add routing information if you need to reach a remote tape or remote CD-ROM host through a gateway; for example,

 # route add default *192.0.2.2*

will route packets to a gateway at 192.0.2.2 (your gateway's address is undoubtedly different).

Remote Tape or Remote CD-ROM: rsh Access

To use remote media, the remote system must support the rsh protocol for remote execution. You'll need root access on the remote machine or a remote "tape user" which allows access by root on the remote host. You must edit the remote machine's tapeuser's .rhosts file to provide access to the BSD/386 machine, at least for the duration of installation. (Note that BSD/386's root's .rhosts file is in /root/.rhosts .)

As superuser ON THE REMOTE MACHINE (csh example):

 # echo *myhost.mydomain* root **>>** *~tapeuser***/.rhosts**

substituting your BSD/386 machine's hostname for *myhost* and your complete domain name for *mydomain* (your complete domain name might contain several words separated by periods – ask your system or network administrator). Substitute the *tapeuser* name appropriately (e.g., root).

If you are accessing a remote CD-ROM, the CD-ROM must be mounted as a filesystem ON THE REMOTE MACHINE:

 # mkdir /cdrom *if necessary*

One of:

 # mount -t iso9660 /dev/mcd0a /cdrom *BSD/386 systems*

 # mount -t hsfs -o ro /dev/sr0 /cdrom *Sun-like systems*

The remote system must support the ISO9660 and/or Rock Ridge filesystems on the CD-ROM.

The floppy system needs a /etc/hosts file to reach the remote system using *rsh*. Create a /etc/hosts file on your BSD/386 floppy like this:

 # echo *192.0.2.27 remotehost* **>> /etc/hosts**

or you can edit the file with the *vi* editor. Be sure that you type the correct host id number and the correct host name.

All: Setup Filesystem Hierarchy

Now it's time to set up the hierarchy of filesystem mount points so that the package loading process will put all the software in just the right place. This task is accomplished by mounting the hard-disk root filesystem onto a directory called '/a' (the loading can't go to the traditional '/', since '/' is being used for the floppy or cdrom root!). Once the hard-disk root filesystem is mounted, then you must create directories for the other mount points (the most common situation comprising exactly one more mount for '/usr', which

Debugging Network Problems

If you suspect network problems, the simplest test is to use "ping" to try and send a packet to another host on the network. Start with a nearby machine (one on the same network) and work your way further out.

Until you get the network running and the 'named' name daemon set up, you will probably need to use the "-n" flag with networking utilities that print hostnames so they will not try and resolve IP Addresses to symbolic hostnames. If you don't, then the programs will hang trying to resolve the name.

You can use "netstat -rn" to view your routing table. Here is the routing table from BSDI's austin.bsdi.com machine:

default	137.39.95.1	UG	9	149782	rn0
127.0.0.1	127.0.0.1	UH	1	5308	lo0
137.39.95.1	137.39.95.2	UH	1	3	rn0
137.39.95.2	127.0.0.1	UGH	1	32201	lo0

The "default" entry points to the service provider to whom all non-local packets are forwarded. If you are on an ethernet, there are probably multiple hosts on your network and your machine will send packets to the local ethernet directly and will only forward packets to other networks when they are not destined for the local network. 127.0.0.1 is the loopback network and you'll note that 137.39.95.2 (the local machine) is routed to it so that packets generated on the local machine that are for the local machine are quickly routed without using the network.

Austin.bsdi.com is on a point-to-point link so there is only one other host on the "network" to 'ping'.

The 'ifconfig' command reports (and changes) the configuration of a network interface:

```
# ifconfig rn0
rn0: flags=51<UP,POINTOPOINT,RUNNING>
inet 137.39.95.2 --> 137.39.95.1 netmask fffffffc
# ping -c 1 137.39.95.1
PING 137.39.95.1 (137.39.95.1): 56 data bytes
64 bytes from 137.39.95.1: icmp_seq=0 ttl=255 time=37.061 ms
--- 137.39.95.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 37.061/37.060/37.061 ms
```

If you do not use the "-c 1" option, ping will keep trying forever. In that case, you can kill it by typing the interrupt character (often '^C').

You can also use the "traceroute" command to print out the path a packet takes on its long journey from here to there (over the Internet it can be a long journey, indeed).

Another tool that can be helpful for network debugging is ‘‘tcpdump’’. Tcpdump enables you see all the traffic passing through your machine or even all traffic on the network to which it is connected. The man pages explains it well. It can consume a vast amount of resources on your machine when running.

Network Configuration Problems

In a severe case, configuration problems will manifest themselves as the appearance of total network failure (i.e., nothing works). Typically, these only occur when you are initially setting up the system. After that, you may find that many services work while some others don’t.

It is difficult to debug configuration problems via documentation. You can examine man pages that seem related and usually glean some information there.

Watch out for having TCP header compression enabled on a SLIP or PPP line that doesn’t support it. Also watch out for "autocompression", since it can sometimes be enabled by line noise.

Check your DNS server, sometimes a dead name server can cause the appearance of a dead network; always keep a numerical IP address handy for using with ping(8).

Helpful commands are ifconfig, ‘‘netstat -i’’ and ‘‘netstat -s’’.

Routing Problems

These are usually seen as the ability to talk to some hosts but not others. Try ping’ing yourself, then a nearby host, then your local gateway. Use traceroute to find out how far the packet is going.

Helpful commands are ping, traceroute, and ‘‘netstat -rn’’.

Connectivity Problems

Connectivity problems are caused by packet filters, broken cables, noisy transmission lines, and crashed servers (amoung other things). Any server your system relies heavily on can cause you problems. NFS and DNS server problems are the most painful. If your DNS server isn’t reliable then you should consider running named locally.

Loading the Distribution

This section presumes that all the disks are set up correctly and that it is time to load the distribution.

Many of the section names below are tagged with a prefix which tells which type of user should perform those steps, e.g.:
Local Tape: Tape Setup
All: Customize

Remote Tape or Remote CD-ROM: Setup Network

If you intend to load the software from a remote tape drive or remote CD-ROM, you must perform a few steps to get the network running. If you are installing from a local device, skip to the next section.

First, you need to ‘configure’ your Ethernet interface (i.e., enable it). Set up your network with ifconfig:

```
# ifconfig XX0 inet hostaddr
```

Substitute we0 for XX0 for Western Digital/SMC style controllers; or substitute ne0, eo0, ex0, ef0, el0, pe0, xir0, or ep0 as reported by the configuration process earlier. `hostaddr` is your system’s host address in ‘dot’ notation, e.g., 192.0.2.1. (Don’t be too rash in choosing numbers; try to fit into your site’s scheme of things.)

Note: Append the phrase `link0` to the `ifconfig` command if you are using the BNC connector on a 3c503 (we interface), 3c507 (el interface) or 3c509/3c579 (ef interface). Append the phrase `link1` to the `ifconfig` command to specify the twisted pair connector of the 3c509/3c579 (ef0).

When using a newer Intel EtherExpress 16 card, the selected connector type will be printed during autoconfiguration. To changed it, specify link0 for the BNC connector, link1 for twisted pair, and both link0 and link1 for the AUI connector.

Include the final arguments ‘netmask 255.255.255.000’ or something similar in the event that you are on a subnet. If you do not know how to choose a host address in dot notation, please consult with your system administrator or network administrator. The numbers are not arbitrary and must conform to your site’s standards.

Verify the configuration with:

```
# ifconfig XX0                ne0, eo0, ex0, ef0, el0, ep0, we0, pe0, xir0
```

The result should include ‘UP’ and ‘RUNNING’ in the output, something like this:

```
we0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
     inet 192.0.2.1 netmask ffffff00
     broadcast 192.0.2.255
```

You should be able to ping an internet address (using dot notation, not by name) now as further verification that you are connected to the network correctly. Be sure to hook up the cable!

corrupted (i.e., *diskdefect* prints lots of errors) then you should zero the bad-sector table:

```
# diskdefect wd0 serial-number
```

Then reboot the system before running a *diskdefect* scan as above.

A 33 Mhz 486 took just over four minutes to scan a 200MB drive.

Build/Initialize Filesystems

Each BSD/386 filesystem partition must be initialized via *newfs*; do not use *newfs* on DOS partitions or you will destroy their data.

→ Initializing a BSD partition using newfs destroys all its files ←

Now, initialize each of the BSD/386 file system partitions (but not the swap partition or any partition that will be used by another operating system). Run:

```
# newfs /dev/rwd0a      rsd0a for SCSI
# newfs /dev/rwd0h      rsd0h for SCSI
and so on
```

to initialize the root, then do almost the same thing with the other BSD/386 filesystems (e.g., /dev/rwd0h). There is no need to initialize the swap area (partition 'b'), but you'll probably need to run newfs on /dev/rwd0h (rsd0h for SCSI) if that is where you intend to store /usr. You may want to run *fsck* on the filesystems to make sure that they work as expected. Note the use of the raw device name of the disks – with the 'r' in front.

Running X11R5

The 1.1 release includes two distinct X server distributions. The default server is X386-SGCS (version 1.4) from Snitily Graphics Consulting Services (SGCS). The XFree86 servers (version 2.0) are also all part of the binary release and are included in the source distribution as well.

Loading the X packages is just like loading the other OS packages. If you didn't load them when you loaded the rest of the release, you can load them at any time using *installsw*.

We recommend that you do not extract X source unless you will be doing active X development or debugging; an X tree with compiled objects and binaries can eat 200 MB of disk space or more.

You may not want to extract X binaries unless you either have a local display that supports X or have another machine that runs X and want to exec xterms on the BSD/386 machine while doing work there.

Motif

The Motif windowing system requires a licensed product that BSDI does not distribute. Contact Sequoia International, 600 W. Hillsboro Blvd., Suite 300; Deerfield Beach, FL 33441 at +1 305-480-6118.

Mice

The SGCS X server supports almost any serial mouse and the Microsoft and Logitech compatible bus-mice. The XFree86 server also supports serial and bus mice and additionally supports PS/2 style mice (which use the auxiliary port on the PS/2 style keyboard controller). Note that many bus-mouse controllers must be configured before they can be used. For example, to get the ATI bus mouse to work, you'll need to boot the ATI configuration utility (included with the card on a DOS diskette) and enable the mouse port at the primary address (0x23c) and IRQ 5.

NOTE: ATI bus mouses can be either of two different device types: Some are type bms and some are type lms.

Configuration

Once you have loaded the X binaries, you must generate a configuration file for the X server. The X configuration file for both the SGCS and XFree86 servers is /etc/Xconfig. The X386-SGCS man page describes the contents of the file for the X386-SGCS server and there is a simple program called `configX386` that may be used to generate the file. In general, all you should need to do to configure X is run `configX386` as root and answer its questions. There are also some sample X configuration files for the SGCS server in /usr/X11/lib/X11/etc/Xconfig.

If you wish to use the XFree86 X servers, you'll need to point the /usr/X11/bin/X symbolic link to the appropriate server for your display hardware and generate an appropriate configuration file. The XFree86 and

Xconfig man pages present information on the XFree86 servers and configuration files. There is a sample configuration file and some additional documentation on configuring XFree86 in /usr/X11/lib/X11/etc/XFree86. There are also man pages for the various XFree86 servers.

Hints for Running X

Once configured, make sure /usr/X11/bin is in your search path and start the X window system like this (example for csh):

```
% set path=($path /usr/X11/bin)
% startx                        starts an xterm window
```

In the xterm window, start the window manager:

```
% twm &
```

You can start a new xterm just by invoking /usr/X11/bin/xterm.

You can use control-alt-[plus] (where [plus] is the plus key on the far right side of the keyboard) to adjust resolution (it's a circular list) of resolutions listed on the 'Modes' line in the Xconfig file.

You can use control-alt-backspace to cause the X server to exit immediately.

If the server fails to reset the screen correctly, you can often recover by running the `vreset` program. You can find out what errors the X server generated (even if you were forced to reboot) by looking in the .xinit.log file in the home directory of the user who attempted to start the server.

Initialize Bad Block Table(s) for non-SCSI disks

This command does not need to be run on SCSI disks, they do bad-sector replacement internally. If you are initializing a SCSI disk, skip this section.

If you have a disk drive with perfect media, zero the bad block table:

```
# diskdefect wd0 serial-number
```

and move on to the next section.

The *diskdefect* utility will scan your disk(s) for bad sectors and set or update the BSD/386 bad-block table. Optionally, you can specify known bad sectors on the command line. Running *diskdefect* with no arguments will display a verbose usage message.

Choose one of the next two command sequences for running *diskdefect*. These commands should be run in single-user mode or with the partition umounted.

If there is no previously existing bad-sector table, use this command to create one:

```
# diskdefect -s [-w] [-v] [-a] wd0
```

[You may see the message "wd0: format error in bad sector file" – this is not unexpected; ignore it.] Note that if you already have a bad-sector table, the *diskdefect* program will not find the bad-sectors already in the list (since they have been "mapped away") – then it will overwrite the bad-sector table and omit those bad blocks! Be sure to specify '-a' to append more bad-sectors if you have an already existing bad-sector table.

To scan a disk and supply additional known bad sectors, you will need an arbitrary disk serial-number (e.g., the date, 030193) and a list of the known bad sectors.

```
# diskdefect -swv wd0 serial-number sector#1 [...]
or
# diskdefect -swv -a wd0 sector#1 [...]
```

As before, to append to an existing bad-sector table, specify the -a option and omit the serial number. The -w option specifies a read/write scan instead of just a read scan (your disk will remain intact as long as there is no other system disk activity). The -v option specifies verbose output (lots of warm fuzzies).

After the disk scan, diskdefect will prompt you to confirm that you want to write out the bad-sector table. It will also print out a warning if it thinks you are going to overwrite an existing bad-block table. If you have never run *diskdefect* on the drive it is safe to overwrite the table (since *diskdefect* simply checks for a certain value at a certain location on the disk it can be fooled by garbage data). If you are unsure if *diskdefect* has ever been run, then kill the first invocation of the program and type:

```
# diskdefect wd0
```

diskdefect will print any existing data. If the data looks sane and there are no errors, you should probably use the '-a' option. If the data looks sane and is

Use the lower case 'a' through 'h' keys to set lengths; use the upper case keys to set types. Heed the warnings about aggregate lengths being too long or too short. You can not proceed to the next phase if the BSD partitions will not fit in the FDISK partition. Note that *disksetup* presents a default size value if you change the size of a BSD partition and it is possible for that partition to fill out the FDISK partition exactly. This is handy for the 'h' partition in this example; less handy for the rest.

Set the sizes and types and type 'N' to move to boot block installation.

```
                                BSD Partition Sizing
FDISK
Part# Start   Length BSD Type   Length MB  Warnings
  2   22048  409600  a  Unused    0   0
                     b  Unused    0   0
                     h  Unused    0   0  BSD partitions 409600 too short

Commands:  [a-h] Set BSD partition length   [N]ext phase
           [A-H] Set BSD partition type     [P]revious page (sizing)

Command>

Directions:
   Use 'a' thru 'h' to set the BSD partition lengths
   Use 'A' thru 'H' to set the BSD partition types
   Use 'N' to move to the next phase: Installing boot blocks
   Use 'P' for previous pg  (FDISK sizing and BSD partition assignment)
```

Installing Boot Blocks (Both BSD-only and Mixed Systems)

The last phase of *disksetup* asks you which kind of boot blocks to install on your disk. Make the correct menu choice:

```
Which boot blocks do you wish to install?

     1: wd   (IDE, ESDI, ST506)
     2: aha  (ISA SCSI -- AHA154[02][BC[F]], BT542, BT445S)
     3: eaha (EISA SCSI -- AHA174[02])

Boot block type:
```

Finally, if you are sharing the disk among multiple operating systems, *disksetup* asks which master boot record you wish installed. If there was a pre-existing master boot record, a query asks if it should be left in place. If you wish to use *bootany*, you should choose to overwrite the existing master boot record.

The next query asks which master boot record should be installed. If you elect to install the default (i.e., *bootany*) you can choose it by using the return key. Alterntatively, you can enter the pathname of a different boot program (which you must supply).

If you elect to install *bootany*, you must tell which partitions are bootable and what identifying string should be displayed by the *bootany* program (*bootany* is a bootstrap loader that enables you to choose which operating system to bring up).

NFS Configuration

The Network File System (NFS) enables remote access to files as if they were local. Many man pages document the NFS filesystem.

nfsd(8) - NFS server (required for server)

Nfsd runs on a server machine to service NFS requests from client machines. At least one nfsd must be running for a machine to operate as a server

mountd(8) - service remote NFS mount requests (required for server)

Mountd is the server for NFS mount requests from other client machines.

nfsiod(8) - local NFS asynchronous I/O server (optional for client)

nfsiod runs on an NFS client machine to service asynchronous I/O requests to its server. It improves performance but is not required for correct operation.

exports(5) - define remote mount points for NFS mount requests

Each line in the /etc/exports file specifies one remote mount point. The first field is the mount point directory path followed optionally by export options and specific hosts separated by white space. Only the first entry for a given local file system may specify the export options, since these are handled on a per local file system basis. If no specific hosts are specified, the mount point is exported to all hosts. Unless you specify the -subdir or -alldirs option in the export file, remote machines will need to mount the directory exactly as specified.

You can list the export list of an NFS server using "showmount -e servername".

After editing /etc/exports you must notify mountd(8) for the changes to take effect. Find mountd's process id and send it the HUP signal (kill -HUP mountd_pid).

PC/NFS Configuration

When using PC/NFS, change /etc/rc to start mountd with the -n flag, then reboot for that to take effect (or you can kill and restart the mountd daemon).

Use the "-subdir" option in /etc/exports to export all of a directories subdirectories (e.g., if you export /var/spool without the -subdir option then a remote system cannot mount /var/spool/lpd).

You must also export the actual filesystem, use the "/bin/pwd" command to determine this information (just typing "pwd" isn't good enough as some shells have a built-in 'pwd' which symbolic links fool). Be careful with /var, which is often a symbolic link to /usr/var.

Add "echo -n pcnfsd; /usr/contrib/bin/pcnfsd" to /etc/rc.local at the appropriate place.

FDISK Sizing & BSD Partition Assignment

Figure out your disk's overall layout scheme (e.g., how much for MS/DOS; how much for BSD/386; how much for any other system). Enter the parameters using the 'L' key to enter the FDISK partition size (do this first). Next, set the FDISK partition types using the 'T' key. Then, use 'S' to enter the starting sectors. Adhering to this order will minimize distractions when partition starting sectors are moved due to constraints inherent in the various operating systems.

Adjust the lengths and starting sectors to eliminate all warnings.

Finally, assign the BSD partition names ('a' through 'h') that you wish to use to refer to the various FDISK partitions. A typical scheme for BSD partition layout is: 'a', 'b', and 'h' for the BSDI portion of the disk; 'd' for the first other operating system (e.g., MS/DOS), 'e' for the next, and 'f' for the next. Of course, it is common to have only two operating systems present.

You should only assign to FDISK partitions those BSD partitions that you plan to use. You need not assign every single BSD partition. It is very common to see some unused BSD partitions and unused FDISK partitions.

When you have set up the FDISK partition sizes and the BSD partition assignments, use 'N' to proceed to the next screen. You can not proceed in certain cases (e.g., no FDISK partitions assigned). Resolve the issue and then move on.

```
         FDISK Partition Sizing & BSD Partition Assignment
FDISK
Part#  type       Start Length  MB BSD Partitions  Warnings
  1  [0x01 DOS  ]    53  20480  10 d              Gap: 53  [OK]
  2  [0x9F BSDI ] 22048 409600 200 a b h          Gap: 1515
  3  [--unused--]
  4  [--unused--]
Last data sector:1736491                    Gap at end: 1304844

Unassigned BSD partitions:  e f g

Commands: [T]ype [S]tart [L]ength [A]ssign BSD part [N]ext phase

Command>

Directions:
   Use 'L' and 'S' to set the FDISK part. Lengths and Starting sectors
   Use 'T' to set the FDISK partition Types
   Use 'A' to assign the BSD partitions to cover the FDISK partitions
   -- One BSD partition for non-BSDI FDISK partitions
   -- One or more BSD partitions for BSDI FDISK partitions
   -- Assignment is to the end of the partition list
   Use 'N' to move to the next phase:  BSD partition sizing
```

BSD Partition Sizing

The second screen shows the FDISK partitions of type BSDI – the partitions which will be subdivided into smaller BSD partitions. The start and length of the FDISK partition are shown. Only the length of the BSD partitions is shown; the offsets are computed automatically so that the partitions are packed as tightly as possible (and in the order shown). Only the lengths and types of the BSD partitions can be set on the screen. If required, return the previous screen ('P') to adjust the FDISK starting sector or length.

memory based filesystem), and allocating the remainder to the 'h' partition for /usr directory and files owned by your system's users.

Enter 'N' when the parameters shown meet your needs to move to the next phase: boot block installation.

The screen reproduction shows the layout for BSD partition sizing.. The first column shows the partition name. Partitions that are not to be unused show a partition type (changed with 'T'). The third column shows the length of the partition in sectors; the fourth column shows the partition's length in megabytes. A megabyte is 1024x1024 bytes; the displayed number ignores partial megabytes. The fifth column shows the starting sector (set with 'S'). The sixth column shows the calculated last sector. Warnings like 'Overlap' cue you to potential trouble.

```
                      BSD Partition Sizing

    BSD Type       Size    MB     Start     End  Warnings
     a  4.2      16744 (   8)         0   16743
     b  swap     66768 (  32)     16744   83511
     c  Unused 1271400 ( 620)         0 1271399
     d   <unused>
     e   <unused>
     f   <unused>
     g  4.2     358384 ( 174)     83512  441895
     h  4.2     829296 ( 404)    441896 1271191
                    Last Data Sector:  1271191

    Commands:  [T]ype  [S]tart  [L]ength  [N]ext phase

Numbers can be relative (with + or -) or suffixed with c (cylinders),
            m (megabytes), or M (megabytes rounded to cylinders)
```

Type 'N' when you wish to proceed to the next phase.

FDISK & BSD Partition Setup

Skip this section if your disks are to contain only the BSD/386 operating system.

The *disksetup* program uses two interactive screens to complete FDISK partition & BSD partition setup.

The first interactive screen has two goals. It enables you to set the starting sector, length, and type for each FDISK partition to be used. It also enables you to assign the name of the BSD partition(s) to be associated with each FDISK partition.

The *disksetup* program does **not** allow arbitrary disk layouts; instead it enforces common sense rules. For instance, *disksetup* does not allow partitions to overlap in any final configuration. *Disksetup* also occasionally adjusts start addresses to various boundaries (a warning is shown). FDISK partitions can have more than one associated BSD partition only when the FDISK partition type is 'BSDI'.

Problem Guide

DOS Time is Incorrect

BSD/386 normally stores the computer's time time in GMT in the CMOS clock, which means that DOS time could show up several hours different from your local timezone. You can change this by reconfiguring the kernel. In the kernel config file change the 'timezone ***HH***' line ('timezone 0' by default) to specify the number of hours west of Greenwich (GMT). The keyword 'dst' may be appended to set the daylight savings correction. If you are not using DOS (or don't care about having DOS show GMT time) then you do not need to change this. BSD/386 does the translation from GMT to whatever timezone you happen to be in (/etc/localtime is used by default, users can override this by setting the TZ environment variable which points to a file in /usr/share/zoneinfo/).

DOS Time is Set to 12:00

You need to update your boot blocks to those in BSD/386 Release 1.1. Make sure you have the latest book blocks in /usr/bootstraps and execute "disksetup -B wd0" (or sd0 for SCSI).

My Boot Blocks Are Gone!

Problem: When I installed the DOS 5.0a update or DOS 6.0 it erased bootany!

No problem. Just boot from floppy and run "disksetup -B wd0" (or sd0 for SCSI). This will reinstall the boot blocks.

Automatic File System Check Failed... Help!

If the system cannot cleanly fsck all the filesystems in /etc/fstab it will print "Automatic file system check failed... help!" and enter single-user maintenance mode. Typically, all that is required at this point is for you to run a few passes of "fsck -y /dev/rwd0a" (where rwd0a is the partition that failed to fsck cleanly). If that doesn't clear things up then you will probably need to newfs the filesystem and reload its contents from your system backup tapes.

This should not be a common occurrence, if your filesystems are being corrupted often, then something is probably seriously wrong with your system or its configuration. For example, serious filesystem corruption can occur if you enable swapping or run newfs on the wrong disk partition or if you have configured your disk partitions to overlap.

The only other time corruption should occur is when the system crashes or is powered off without a proper shutdown. These cases usually result in very minor corruption that is automatically repaired by the system on reboot.

Booting from Floppy

If for some reason you need to boot single user (e.g., you forgot the root password, hard disk will not boot, filesystems are corrupted, or /etc/fstab is incorrect or missing) you can boot off the original boot floppies.

1 Boot system
  - Boot from floppy1, switch to floppy2 if tape or remote install
  - Select root device (mcd0, fd0, sd1). Use a CDROM drive if you have one. If not, use fd0 and switch to the second boot diskette when prompted to change diskettes.
  - Run "setup"
  - If still running 1.0: run "mount -t mfs /dev/fd0c /tmp"

2 Configure local disk. It's best to always run fsck before mounting something.
  - Run "fsck /dev/rwd0a" [rsd0a for SCSI disk]
  - Run "fsck /dev/rwd0h" [rsd0h for SCSI disk]
  - Run "mount /dev/wd0a /a" [sd0a for SCSI disk]
  - Run "mkdir /a/usr"
  - Run "mount /dev/wd0h /a/usr" [sd0h for SCSI disk]. Mount any other filesystems you created under the /a directory. For example, if you have a /var partition mount it on /a/var.

You are now in a semi-normal work environment. You could now run installsw or you could continue on with the customization section:

- Setup work environment
  - Run "/usr/sbin/chroot /a"
  - Run ". /.profile"

This will place you in a virtual working environment. You can use all the standard commands and they will operate as if you booted from the hard drive. Execute "reboot" when you are finished (be sure to eject the floppy if you intend to boot from hard disk).

BSD/386 disk numbers (sd0, sd1, ...) are mapped onto SCSI target/Logical Unit Number (LUN) addresses in a fairly straightforward way. The logical SCSI disk 0 (/dev/sd0[a-h]) is the first responding disk LUN on the first target with a disk.

SCSI tapes are assigned in a similar way: SCSI tape 0 (/dev/rst0) is the first responding tape LUN on the first target with a tape LUN, and so on.

BSD/386 disk numbers do NOT directly correspond to SCSI targets or LUNs. If a configuration has disks on targets 0, 2, 5 and 6, these correspond to devices /dev/sd0[a-h], /dev/sd1[a-h], /dev/sd2[a-h] and /dev/sd3[a-h]. A solitary tape on target 4 will be /dev/rst0, not /dev/rst4.

Interactive Sizing

Disksetup displays one of two different screen sets for sizing, depending on whether the drive is to be shared between operating systems (i.e., whether co-residency is required). If the drive is used exclusively for BSD/386, then *disksetup* displays the 'BSD Partition Sizing' screen. Otherwise, it displays the 'FDISK Partition Sizing & BSD Partition Assignment' screen. Separate sections that follow document these two different execution paths.

In general, the interactive screens use 'hot keys' for their commands and otherwise require a 'return' to be entered after the input. A 'return' with no input usually aborts the current operation; otherwise, it indicates acceptance of a default value which has been presented between square brackets. Use backspace to delete a character; use control-U to delete an input line.

Enter numbers in many different ways: absolute, relative, and with a unit suffix. Absolute numbers are entered without a sign and have a value identical to the number typed. Relative numbers are preceded by a + or - sign and specify a change to the current value (i.e., an increase or decrease).

Normally, numbers represent sectors, However, specifying a unit suffix can change the meaning of the numbered typed. Use the 'm' suffix to specify megabytes (1024*1024 bytes); 'c' for cylinders, or 'M' for cylinder-aligned-megabytes (the entered number multiplied by 1024*1024 and rounded up to the nearest cylinder boundary.

BSD Only Disks

Skip this section if your disks will contain multiple operating systems.

The goal of the interactive screen is to set the sizes, starting offsets (i.e., starting sector numbers), and types of the BSD partitions.

This screen enables you to set up overlapping partitions and will display a warning in the 'Warnings' column. Unless you have specific needs for overlapping partitions, they are best avoided.

Use the 'T', 'S', and 'L' keys to set the parameters you desire. As a starting idea, consider using an 10 MB 'a' (root) partition, 16 MB 'b' (swap) partition (32MB if you will be using the X window system or MFS, the

Abbr.	Description
MK2124FC	Toshiba MK2124FC 2.5 Inch 120 MB IDE
ST41200	Wren VII
bt9220	Brand Technologies 200Mb IDE
compaq120	120 MB drive of Compaq SLT 386s/20
cp30104	Conner 120 MB IDE
cp3204f	Conner 207 MB IDE
cp3301	Conner 340MB IDE, Compaq Type 63
cp3540	Conner Peripherals 540MB SCSI
cp3544	Conner Peripherals 540MB IDE
im94186-766E	imprimis 94186-766E 640 MB ESDI
maxtor213a	Maxtor LXT213A, Maxtor 200 MB IDE
maxtor4380	Maxtor XT4380E, Maxtor 380 MB ESDI
maxtor8760 Maxtor XT8760E, Maxtor 650 MB ESDI	
maxtor8800	Maxtor XT8800E, Maxtor 660 MB ESDI
mic1558, mic1558a	Micropolis 1558 ESDI
micropolis1518	Micropolis 1518-15 24MHz 1354 MB ESDI
miniscribe7080	MiniScribe 7080 IDE
miniscribe9380, compaq38	Miniscribe 9380 ESDI
nb60	Toshiba T2200SX Notebook 60 MB IDE
nb80	Toshiba T2200SX Notebook 80 MB IDE
nec300	NEC 300 MB ESDI
rz35	DEC rz35 SCSI
seagate211	Seagate ST1239A 211mb IDE
seagate300	Seagate 300 MB drive
type17	40 MB IDE Winchester
type38	400 Mb IDE Winchester
wd200	Western Digital generic 200 MB IDE
wd340	Western Digital generic 340 MB IDE
wdcav, wdCav	Western Digital Caviar 200 MB IDE

Naming Disks in Multiple Disk Systems

Be sure you know the correct name(s) of your disks. While a system with a single non-SCSI disk uses wd0 as its disk name, a multi-disk system might use wd1 for the second disk (on the same controller as wd0) or wd2 for the name (for the first disk on a second controller).

SCSI Disk Naming Conventions

If you have a SCSI disk, you need to know how BSD/386 assigns numbers to SCSI disks. SCSI targets are devices on a SCSI bus. Each has an address between 0 and 7 inclusive (usually set by switches on the hardware). A target may support one or more logical units (devices), but typically supports just one.

Attaching Serial and Parallel Devices

Attaching and configuring a serial device requires five short steps:

- Choose the name of the new device(s)
- Configure /etc/ttys
- Configure /etc/gettytab [dial-in only]
- Configure kernel (potentially)
- Signal the 'getty' process to recognize the new devices or reboot

Preliminary

You must decide if the device is to be used for modem dialins (with users logging in through it) or not. If dialins are allowed, you should also decide if the device is to be used for both dialins and dialouts (''bidirectional'' mode).

Assigning The Name For The New Device(s)

Each serial port has a name that is an entry in the /dev directory. The names are tty00-ttyzz. The second two letters identify the interface and 'unit number' of the serial port. Standard single serial device ports are numbered tty00-tty09 and tty10-tty19. Multiplexors use different ranges of tty numbers to denote the device to which they interface: tty[abefg][0-7] for the RISCom/8, /dev/ttyh[0-7] for the MAXpeed card, and so on.

Deduce the tty* name(s) of the serial port(s) you'll be using by peeking ahead at the hardware section for your device or using the names from the previous paragraph.

Configuration Files For Serial Device(s)

You will configure the /etc/ttys file and maybe the /etc/gettytab file.

The /etc/ttys file defines which devices should have a login process started on them by the init(8) process (i.e., those devices which will allow users to dialin or login to them). The fields are ordered as follows:

Name: the name of the tty device (e.g., tty00)

'Getty': the command init(8) should execute to initiate the login process (e.g., "/usr/libexec/getty std.9600", where std.9600 is a valid entry in the /etc/gettytab file)

Type: the default value of the TERM environment variable for logins on that device

Status: defines if the device is active ('on'/'off') and whether uid 0 logins should be allowed ('secure' means allow them).

By way of example, this /etc/ttys file segment configures tty00 as a dedicated, 9600-baud, vt100 terminal that allows root access; tty01 as a bidirectional, 57600-baud, dialin/dialout line; tty02 as a dialup 19200->9600->2400->1200 baud rotary; and tty03 as disabled.

```
#Name   Getty                          Type     Status
tty00   "/usr/libexec/getty std.9600"   vt100    on secure
tty01   "/usr/libexec/getty bidir.57600" dialup  on
```

```
tty02   "/usr/libexec/getty t19200"      dialup   on
tty03   "/usr/libexec/getty std.9600"    dialup   off
```

After changing /etc/ttys, execute "kill -HUP 1" to signal init that the file has changed and must be re-read. If you have disabled a line, then run "ps axww | grep getty" and make sure that getty is no longer running on that device (kill it with signal 9 if one is). Note that new teletype descriptions must appear at the end of /etc/ttys or you must reboot when you have finished editing the file.

The /etc/gettytab file is a collection of device specifications that getty uses to configure the baud rate, parity, and other device specific parameters before printing the login banner. The supplied configuration may contain all the entries you need.

The 'std.*BAUD*' entries can be used for fixed speed connections (such as dedicated terminals). You can also use the "rotary" tables (e.g., d1200, d2400, t19200, and t9600a) to enable variable speed connections which toggle through a set of speeds by pressing the break key (e.g., "/usr/libexec/getty t19200" selects the 19200->9600->2400->1200 baud rotary, changing baud rates each time a 'break' is received on the serial line).

See the gettytab(5) and ttys(5) man pages for more information.

Lines for 'Dialout' Only

Serial lines which are not used for dialins require only the simplest of configurations. Add an entry like this to the end of /etc/ttys for each new serial port:

```
tty__ "/usr/libexec/getty std.9600" unknown off
```

If you choose to add the entry other than at the end of the file, you must reboot the system instead of signalling process 1.

For non-dialin serial lines, you will also want to ensure that no getty process is running on any of the new ports:

```
# ps axww | grep tty__
```

You need to check each of the new serial lines.

Remember to signal the 'init' process as superuser superuser any time you change /etc/ttys:

```
# kill -HUP 1
```

or reboot (which is mandatory if you made a new kernel).

'Dialin' Lines

Serial lines used for dialins require you to choose an entry from /etc/gettytab that describes the serial port (the std.9600 entry may fit your bill). Of course, you may modify the file, if you need to.

See the section "Bidirectional Connections" for additional information on configuring lines that operate in both incoming and outgoing modes.

Other options are described below.

disksetup asks if the disk is to be used exclusively for BSD partitions or if it is to be shared with other operating systems. Answer the question appropriately.

disksetup learns the drive's geometry either by reading it from an existing BSD label on the drive, interrogating the hardware, or by querying the user. If no BSD label previously existed, no geometry information is available so *disksetup* asks the user how to proceed. Otherwise, *disksetup* proceeds to the sizing screens (see below).

When no geometry information is available, *disksetup* asks the user to choose from:

- Running the *scsicmd* utility (only for SCSI disks, of course) to query the drive for its geometry,
- Querying the IDE or ESDI disk controller for its geometry (works only on some controllers)
- Supplying the name of a 'disktab' entry that corresponds to the disk to be labeled (chosen from the file /etc/disktab which is summarized in the table below),
- Naming a file containing a saved geometry (usually from a previous invocation of disksetup (described later) or some other command)
- Answering prompts about the geometry,

When manually answering the prompts for a geometry, the user must supply the following data:

- Interface vendor and type (e.g., 'Maxtor P012S') which is only used to document the disk's vendor and type ID
- Bytes/sector (usually 512, always a multiple of 512)
- Number of cylinders (from the manual; must be the number of data cylinders on the drive – it excludes those cylinders for the drive's own internal use such as for bad blocks)
- Number of heads (also known as tracks/cylinder; from the manual)
- Number of sectors per track (from the manual); you may have to specify the number of sectors reduced by one for some controllers (those which reserve spare sectors per track)
- Total data sectors (which might not be the same as the default in the case that the last cylinder is not a complete one). For SCSI disks, the correct number for total data sectors is printed during autoconfiguration.

The user then verifies that the information is correct before proceeding.

Here are the types currently described in /etc/disktab (listed with their short name, aliases, and their long name):

course. You can always start out with /usr/local as part of /usr and, if needed, create a separate partition later (and point a symbolic link to it).

Usenet News Partitions

Usenet news tends to create a very large number of smaller-than-average files. A full newsfeed can consume an enormous amount of disk space (not to mention network bandwidth). You will probably need to alter the default newfs(8) parameters when creating the filesystem to get the best results.

By default, newfs(8) allocates one inode for every four disk fragments. A fragment is the smallest unit of disk allocation while a block is the largest unit of I/O. The default is 1K/8K, but for news filesystems you probably want to use 512/4K.

One inode is required per allocated file; when you run out of inodes, it's like running out of disk space (only it's more annoying because you have disk space left). For filesystems containing news the average file size is only about 3K so you will need to increase the total number of inodes available (by decreasing the bytes per inode value) or else you will probably run out of inodes.

The following command will create a filesystem with a fragment size of 512 bytes and allocates an inode for each 2048 bytes of data on the filesystem. This should be reasonable for most news partitions: filesystem. This should be reasonable for most news partitions:

```
# newfs -f 512 -b 4096 -i 2048 /dev/rsd__
```

Of course, you'll want to substitute the correct disk partition name.

Disksetup

The goal of the *disksetup* program is to define the FDISK table if needed (see below), to define the BSD partition tables (and geometry information), and to write the appropriate boot blocks to each disk on the system.

Disksetup installs, examines, or modifies both the FDISK and BSD/386 disk labels on a disk drive. *Disksetup* can set or change the disk geometry, the sizes, types and offsets for the FDISK and BSD/386 partitions. It can also write both BSD/386 boot blocks and, optionally, master boot record boot blocks.

Invoking Interactive *disksetup*

Disksetup is most often used in the interactive mode. Invoke it by specifying the **-i** flag and naming the disk to be set up. The example below shows invocations for SCSI and non-SCSI primary drives, respectively:

```
# disksetup -i sd0        SCSI
 or
# disksetup -i wd0        non-SCSI
```


Remember to signal the 'init' process as superuser superuser any time you change /etc/ttys:

```
# kill -HUP 1
```

or reboot (which is mandatory if you wish to start a new kernel).

Single Serial Port Configuration

After configuring the files listed above, create 'device entries' in the /dev directory to access various hardware devices. To create the /dev device entries for com ports, execute the following commands as root:

```
# cd /dev
# ./MAKEDEV com0 com1 com2 com3 com4 com5 com6 com7
```

This command creates /dev/tty00 through /dev/tty07 with the appropriate major and minor device numbers for single serial ports. Of course, you may wish to configure only those com ports you have installed. The MAKEDEV command will create the appropriate /dev/tty* devices.

See the com(4) man page for additional information.

Ensure that the physical serial device is configured in your kernel. The GENERIC kernel (the initial kernel you install) includes the standard PC serial ports called COM1: and COM2: (they print in the configuration list as com0 and com1 and are accessed as /dev/tty00 and /dev/tty01 since BSD/386 usually starts number at 0 instead of 1). See the section "Hardware Configuration" for adding more devices.

Bidirectional Connections

Bidirectional connections enable you to share a resource (typically a modem) between users dialing in and users dialing out. One problem with bidirectional modems is that a user with access to the dial-out modem can create a 'Trojan horse' on the modem and trap other dial-in user's passwords. Unless you trust all your dial-in and dial-out users, you should not setup a bidirectional modem connection. There doesn't appear to be any way to solve this problem with current modem hardware.

To configure a tty line to support bidirectional access (dial-in and dial-out on the same tty) change the appropriate line in /etc/ttys to use an /etc/gettytab entry with the 'bi' name, for example:

/usr/libexec/getty bidir.19200" dialup on /usr/libexec/getty bidir.57600" dialup on

Look at /etc/gettytab to see what is available; you can also add new speeds or options yourself.

See the gettytab(5) and ttys(5) man pages for more information.

Hardware configuration

The hardware for the standard com ports is expected to be at IO port 3F8 hex for COM1: and 2F8 hex for COM2: (called tty00 and tty01 by BSD/386). You can configure the serial ports at any Interrupt Request level (the IRQ) and

the kernel should still be able to configure and interface to them correctly (presuming the interrupts do not conflict with other devices, of course).

The system should save the initial kernel configuration messages in the file /var/db/dmesg.boot. If that file does not exist, you can try the "dmesg" command right after a system boot or look in /var/log/messages to see that the kernel has detected the devices correctly.

If your serial ports are not found and reported at boot time, then check their hardware configuration. If you need to, you can change your kernel configuration to find them wherever you have their hardware configured. Here are the configuration lines from the default kernel configuration file:

```
device          com0    at isa? port "IO_COM1"
device          com1    at isa? port "IO_COM2"
```

See the com(4) man page for information about configuring AST or Boca multiport serial cards.

When adding devices you must make sure that the I/O Memory, I/O Port, and IRQ settings do not conflict with any other device (sometimes this is a challenge because the devices may not be as configurable as you might need). Other device examples can be found in the kernel config file. See "Rebuilding Your Kernel" in the release notes and the config(doc) man page (which you can read by typing "man doc config") for more information about kernel configuration.

SDL Communications Multiport RISCom/8 Cards

To configure the SDL Communications RISCom/8 card you must add the 'rc' device in the kernel configuration file. For example:

```
device          rc0     at isa? port 0x220
```

To create the /dev device entries, execute the following commands as root:

```
# cd /dev
# ./MAKEDEV rc0
```

This will create the /dev/ttya[0-7] devices for the RISCom/8 card. Be sure to add any new devices to /etc/ttys (and reboot your system after editing /etc/ttys).

You can add additional RISCom/8 cards by adding additional kernel configuration lines for rc1, rc2, ... at different port numbers (with the hardware also configured to have unique IRQs). Be sure to MAKEDEV the devices and add them to /etc/ttys.

See the rc(4) man page for additional information.

Maxpeed Serial Multi-port Cards

The driver supports both the SS-4Plus and the SS-8/2 family of intelligent serial controllers.

To configure the Maxpeed serial multi-port card you must add the 'ms' device in the kernel configuration file. For example:

```
device          ms0 at isa? tty iomem 0xd4000 iosiz 8192
```


If you aren't using an MFS for /tmp, then you will need to put /tmp some place that has enough room that you will not run out of disk space. /tmp is one of those filesystems that tends to grow without bounds if you don't prune it back on occasion. Normally, one might imagine that /tmp could live on the root partition. It is best to keep the root partition small and point /tmp to some other filesystem, in that case.

Running out of disk space in /tmp is detrimental to normal system operation and can stop your system dead in its tracks. Be sure you have enough /tmp file space (i.e., at least 8MB or so).

Users' Home Directories

It's a good idea to allocate a separate partition for user directories because it's easy for users to accumulate many files (and hence consume disk space). By putting the users' files in a separate partition you will isolate this disk filling from system disks.

Some people like to name users' directories like '/home/username'; others prefer /u/username (and /u1, /u2, etc), still others use /home/hostname/username (though that usage is on the wane). Whatever you choose, make sure you have a good backup scheme in place (automate this task if at all possible; even simple scripts run from cron are effective).

Source Code

If you plan on loading the BSD/386 source code (or your own), you'll need some place to put it. When you consider disk layout policies, try to put the most often used data in the middle of the disk. When you have multiple disks, try to split often-used partitions onto separate disks. If you have the CDROM distribution, then you can mount the source code directly from the CDROM and save yourself a lot of disk space (until you wish to edit it).

If you plan on building BSD/386 source then you'll need to be aware that the data from the compiles is placed in either /usr/obj or /usr/cdrom_obj (the latter is where compiles from the CDROM place their data). These directories can grow very large.

Local files

There are always local customizations to any system. It makes life a lot easier when it's upgrade time if you can concentrate your changes into as few places as possible.

When you first install a system, execute "touch /.base" before changing anything. Then it is easy to generate a list of changed files simply by running "find -x / /usr -newer /.base". This is handy not only for doing upgrades but also for generating install procedures for large sites.

It's also nice to store the bulk of your local files in something like /usr/local. The required size of this filesystem will vary with your site, of

configuration files are on the root filesystem (in /etc) so you want to keep your root backups up to date.

The /usr Partition

The /usr partition contains the bulk of the system programs and data used in normal system operation. /usr is designed so that it can be shared among computers (via NFS).

If you have a single drive of 300MB or less, you may want to make /usr contain most of your data. If you do this, you will have less "wasted" space since all free space is pooled into one central location instead of spread out in several different partitions.

If you have more than 300MB of disk space or if you have multiple disks, you need to decide how to partition the data, not only for ease of use, but for maintenance, backups, disaster recovery, and performance.

Swap Space

Swap space is where virtual memory goes when it's not in physical memory. Generally, you will need at least 16MB of swap space, 32MB is that much better, if you need it. Generally, the sum of the megabytes of physical memory (RAM) and swap space should be at least 32MB, 64MB if you are running many users or many programs simultaneously. If you have very large physical memory (RAM), you can run efficiently with small swap space (32MB or even 16MB).

The /var Partition

/var contains volatile system data (print and other queues, accounting records, transaction logs, and electronic mail). BSD/386 requires a base of about 4MB for this partition. However, once you toss in /var/tmp (for editor files), /var/mail (email), and /var/spool (mail, printer, and uucp queue), you can easily push the requirements up to 50MB. Don't let /var run low on disk space; many system functions require disk space in /var to maintain normal operation. If you are not allocating a separate partition for /var, then you will probably want to tell installsw to install it into /usr/var when it asks you where you want it. Be sure that there is at least 10MB of 'breathing room' on the /var partition.

The /tmp Partition

Sites with at least 8MB of RAM may want to use a MFS (memory filesystem) for /tmp. Using MFS for /tmp can improve system performance with all the small files that get created and destroyed in normal system operation. Keep in mind that doing so means that /tmp is completely lost on reboot or crash. You should never store important data in /tmp in any case. You'll also want to allocate more swap space if using an MFS.

To create the /dev device entries, execute the following commands as root:

```
# cd /dev
# ./MAKEDEV ms0
```

This will create the /dev/ttyh[0-7] devices for the Maxpeed card. Specify ms1 (etc.) for /dev/ttyi[0-7] and so on. Be sure to add any new devices to /etc/ttys.

You can add additional Maxpeed cards by adding additional kernel configuration lines for ms1, ms2, etc. For example:

```
device          ms1 at isa? tty iomem 0xd4000 iosiz 8192
device          ms2 at isa? tty iomem 0xd6000 iosiz 8192
device          ms3 at isa? tty iomem 0xd8000 iosiz 8192
```

Be sure to MAKEDEV the new devices and add them to /etc/ttys.

See the ms(4) man page for more information.

Digiboard Serial Multi-port Cards

Newer Digiboards will not work with the driver as shipped in V1.0. Contact your BSDI support representative for an update.

The driver supports the PC/4e, PC/8e, and PC/16e intelligent serial controllers. It also supports the PC/8i controller when set to use 64K of memory.

To configure the Digiboard serial multi-port cards you must add the 'digi' device in the kernel configuration file. For example:

```
# DigiBoard PC/4e, PC/8e, PC/16e, PC/8i terminal multiplexor
# Use "flags 0x0001" to switch DSR and DCD for
# RJ-45 connectors.
device digi0 at isa? port 0x220 iomem 0xd0000
                         iosiz 65536 flags 0x0001
```

To create the /dev device entries, execute the following commands as root:

```
# cd /dev
# mknod /dev/ttyD0 c 26 0
# mknod /dev/ttyD1 c 26 1
# ...
# mknod /dev/ttyD7 c 26 7   # last for an 8-port card
# ...
# mknod /dev/ttyDf c 26 16  # last for a 16-port card
```

Be sure to add the new devices to /etc/ttys.

You can add additional cards by adding additional kernel configuration lines for 'digi1', and so on. Be sure to MAKEDEV the new devices and add them to /etc/ttys.

See the digi(4) man pages for more information.

Specialix SLXOS 8-32 Serial Multi-port Card

NOTE: As shipped in V1.1 this does not work. Contact your BSDI support representative for an update.

To configure the Specialix serial multi-port card you must add the 'si' device in the kernel configuration file. For example:

```
# Specialix SLXOS 8-32 port terminal multiplexor
device          si0 at isa? iomem 0x0d8000 iosiz 65536
```

To create the /dev device entries, execute the following commands as root:

```
# cd /dev
# ./MAKEDEV si0
```

This will create the following devices:

/dev/ttya[0-31]
/dev/si_control

Add the /dev/ttya[0-31] devices to /etc/ttys.

You also need to invoke siconfig(8) from /etc/rc.local.

See the si(4) and siconfig(8) man pages for more information.

RS-232 Connection Hardware

Probably the hardest part of installing new serial devices is correctly wiring the RS-232 connector. You must match male plugs to female sockets and match the DTE (Data Terminal Equipment) signal set to its DCE (Data Communications Equipment) counterpart (i.e., the computer's transmit line must be attached to the terminal or modem's receive line and so on for the other signals). The easiest way to debug RS-232 connections is to use a break-out box. A simple break-out box enables you to see which signals are active on the cable; fancy break-out boxes tell you exactly what you need to do to properly connect the devices. You can purchase these at computer electronics stores like Radio-Shack.

Standard PC serial ports are wired as 25-pin male DTE; if you have the 9-pin AT-style serial ports you can get cables that either convert them to the standard 25-pin RS-232 connector or that connect directly to certain devices (e.g., 9-pin to 25-pin modem cables are common). Most modem ports are wired as a 25-pin female DCE; use a straight-through cable to connect them to the PC serial port.

On the other hand, terminals are generally wired as 25-pin male DTE (just like the PC serial port) and require a "Null Modem" cable to connect them. "Null Modem" cables convert the DTE signal set to a DCE signal set by crossing (exchanging) the transmit and receive pairs: TD/RD (pins 2 and 3) and RTS/CTS (pins 6 and 8), DTR/DSR. As a general rule, a DTE to DCE connection requires a straight through cable and a DTE to DTE (or DCE to DCE) connection requires a "Null Modem" cable. Not all equipment strictly follows the DTE/DCE model so some devices require customized cables. A break-out box is probably required to fathom the correct connections in those cases.

Laying Out Hard Disk Partitions

BSD/386 Disk Partitioning

This discussion centers on dividing up the disk space you have assigned to BSD/386. It presumes that all issues of 'FDISK' partitioning and sharing a disk with other operating systems (e.g., DOS) have been addressed.

There are several issues to consider when deciding how to partition your disk space. The primary issue (discussed here) is how much disk space you have and how you intend to assign it. Other issues to consider (which are not covered here) include maintenance, backups, disaster recovery, and performance.

Having a greater number of partitions means you have less data to recover if something happens to one of the partitions. Having too many partitions, though, may mean you might have disk free space where you can't use it. Some of the other trade-offs are detailed below.

If you aren't sure what to make of all of this then your best bet is to allocate 10MB for root, 32MB for swap, and the rest for /usr (and then install the /var filesystem in /usr/var, installsw will prompt you for this after you start the installation process). For a fairly full system you'll need around 220MB in /usr (but that will not leave you much room). Add 13MB if you intend to install ISODE. Of course, you need not install, e.g., emacs, if you don't intend to use it. The installsw program displays disk usage data for each of its packages.

Types of data you might need to consider when defining a disk partitioning policy include:

/	The root partition
/usr	The usr partition
/var	The var partition (sometimes linked to /usr/var)
/tmp	The tmp partition (often an in-memory MFS)
swap	Swap space
/home, /u	User's home directories
/usr/src	Source code
/usr/local	Local system files
/var/spool/news	Usenet News partitions

Of course, not every system needs all of these, and by no means are you required or advised to allocate separate partitions for each of them.

The root Partition

It is a good idea to keep the root partition as small as possible (typically 10MB these days).

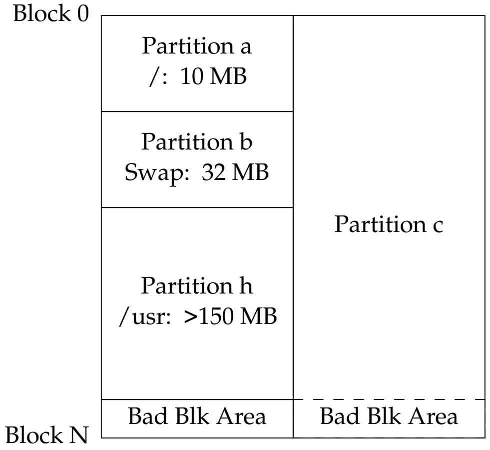
Keeping / fairly small and uncluttered makes it possible to do full backups of / every night; which in turn makes recovery from disk crashes less painful in the event that something goes wrong. Almost all of the system

Filesys	Partition	Size
/	a	10 MB
(swap)	b	16 MB+
-or- (swap)	b	32 MB (w/X and/or MFS)
/usr	h	[the rest of the disk]

Increase the swap partition to at least 32 MB if you intend to use the MFS (in-core filesystem) for /tmp or if you intend to run the X window system. If you think you'll need even more virtual memory, increase swap beyond 32 MB.

Note that partitions 'a', 'b' and 'c' are names reserved for root, swap and the whole disk, respectively. The 'h' partition name is used by convention.

By way of example, a typical set of BSD disk partitions might look like this:



SCSI disks do not have the 'Bad Blk Area'.

Some sites define BSD partitions that overlap (e.g., a 'g' partition that spans the BSD 'b' and 'h' partitions). You must not concurrently access (i.e., concurrently mount) partitions that overlap. The new *disksetup* program handles most of this for you by disallowing the creation of overlapping partitions (beyond 'c', of course). Note that it automatically configures partition 'c' and will not let that configuration be changed.

Some of the more expensive cables enable you to rewire them easily and therefore ease the connecting of equipment. Also available are "smart cables" that configure themselves and gender-benders for connecting two same 'sex' connectors together.

Carrier Detect

Serial interfaces, under BSD/386, require the carrier detect signal (CD) to be asserted on the serial port before the serial line can be used ("opened"). Setting the 'clocal' flag using stty(1) will enable you to open a serial device in the absence of the carrier detect signal. Note that system software (tip, uucico) will automatically set the clocal flag on outgoing connections so you rarely need to do this manually.

You need to set the clocal flag whenever you are connecting to a terminal or dial-out device that does not properly assert carrier detect or if you are attempting to connect to a dial-in device (perhaps to change its configuration). For example, to ignore the absence of carrier detect on tty00, type:

```
stty -f /dev/tty00 clocal
```

When you are finished with the device, you can restore the default behavior by clearing the clocal flag, e.g.:

```
stty -f /dev/tty00 -clocal
```

Note that you can cannot setup a dial-in connection on a device that does not properly assert carrier detect when a connection is made. With hardwired terminals this is not as big a problem, just use a getty entry that contains :hw: in /etc/gettytab (e.g., hw.9600). If you do this with a dial-in modem then the system will not be able to hang-up properly when the caller disconnects or should be disconnected.

Dial-out Security & Configuration

Dial-out access via tip(1) is granted only to those users in the group "dialer" (as defined by the file /etc/group). Add user's names to the 'dialer' line in the /etc/group file:

```
dialer:*:117:smith,jones,brown
```

The group number 117 is arbitrary but to change it you would need to find all existing data and programs in group 117 and change them (mainly /usr/bin/tip and /usr/bin/ppp).

Edit the /etc/remote file (read by the tip(1) program) and set up a simple entry for your device. Sample entries for tty00 and tty01 are included:

```
# file /etc/remote:
# Local access to devices on COM ports
tty00:dv=/dev/tty00:br#9600:
tty01:dv=/dev/tty01:br#9600:
```

Ensure that the devices are owned by the uucp user:

```
# chown uucp /dev/tty00 /dev/tty01
```

To test the connection (once configuration is complete), try:

```
# tip tty00
```

Note that tty00 is just the name of the /etc/remote entry, the choice is arbitrary, and does not have to be the name of the device (but, by convention, is often the same name).

Now you are ‘talking’ directly to the serial device on port tty00, make sure the device is talking to you. For example, on Hayes-compatible modems you can type “AT” and they should reply “OK” (you can watch the modem lights to see if it’s receiving anything). Try “ATE1Q0” to make the modem echo what you type and print result codes. If you are connected to a terminal whatever you type should appear on the display.

You can use this setup to talk to your modem (or other device that is configured via the serial port) to configure it. Things like terminals are usually configured directly from their keyboard. Modems should be configured so they restore the saved configuration setting when the system drops DTR (“AT&D” on some modems). If the device is going to be used for dial-in, then you must also configure it to ‘auto-answer’ (“ATS0=2” on some modems) and to be silent (“ATE0Q1” on some modems). Save the configuration such that the device stays configured properly even after reset (“AT&W” on some modems).

Exit tip(1) by typing tilde-dot-return at the beginning of a line.

You can now use tip(1) to dial-out to remote systems. You could also configure UUCP (see "UUCP Configuration") or other software like Seyon.

See the tip(1), remote(5), and group(5) man pages for more information.

Attaching Printers

The print queue utilities (e.g., lpr and lpq) use the file /etc/printcap to define the queues and capabilities (configuration) of your printers. The default /etc/printcap file contains sample entries for dumb printers (lp), PostScript (ps), and HP LaserJet II (lj) printers. These are probably sufficient to get you started. You can read the printcap(5) man page for all the details about what you can configure.

Be sure and adjust your /etc/printcap entries to reflect your hardware configuration. Common things that need adjusting are: “lp” (defines what device your printer is attached to), “br” (defines what baud rate to use, only for serial printers), “hw” (tells lpr that it’s a hardwired printer), and “ms” (the stty-style mode string for adjusting the stty settings of the device; e.g., to add hardware flow control).

If you are attaching a printer to a serial port, make sure its line is disabled in the /etc/ttys file. Otherwise getty(8) will send login messages to the printer and wait for it to login. See the section "Configuration Files for Serial Devices" to see examples of configuring a serial line described in /etc/ttys to ‘off’.

(e.g., in a typical BSD/386 installation). It is worth emphasizing that a typical BSD/386 installation will see three BSD partitions inside a single FDISK partition assigned to BSD/386.

In common usage, BSD/386 divides its part of the primary disk into three sections: one for the root filesystem (always the ‘a’ partition), one for swapping (and paging) (always the ‘b’ partition), and one for /usr and other files (usually the ‘h’ partition). If BSD/386 is to access other parts of the disk (e.g., for MS-DOS co-residency), another BSD partition is assigned to refer to that disk data (often the ‘d’ partition). Of course, there may be a third or fourth FDISK partition for yet other operating systems – they might be assigned the BSD partition names ‘e’ and ‘f’.

By the way, sophisticated users sometimes set up additional BSD partitions, e.g., for the /var directories. For load balancing purposes, these often appear on other hard drives, though.

The BSD/386 uses the *disksetup* program to specify both the FDISK and BSDI partitions on the disk. BSDI does not recommend attempting to use the DOS FDISK program to set up FDISK partitions.

VERY IMPORTANT NOTES

On ESDI and IDE disk drives, the CMOS disk setup geometries must match that supplied to the BSD/386 disksetup program.

When sharing a disk among operating systems, the geometry for that disk must be the geometry that DOS uses.

When installing co-residency on a drive with DOS already installed, don’t forget to backup your DOS partition, run DOS FORMAT on it, and re-install DOS.

The BSD ‘c’ Partition

BSD/386 always defines a partition that spans the entire physical drive: the ‘c’ partition. The ‘c’ partition refers to the entire physical disk (including the boot blocks and the bad sector table at the end of the disk). This partition overlaps all other BSD disk partitions and is created only for the use of some internal system software; it is not meant for access by user programs. To define a BSD partition that spans an entire logical disk, use a different, slightly shorter partition (e.g., define ‘h’ to be the entire usable portion of the disk). In the interactive (**-i**) mode, *disksetup* enforces the correct maximum size.

Your primary BSD/386 hard-disk probably will have three BSD/386 partitions (probably all contained inside one FDISK partition): one for the root filesystem, one for swapping, and one for the /usr partition. We recommend a set up like this:

Once the drives are hardware formatted, use the *disksetup* program to specify the BSD and FDISK partition tables (see the section on *disksetup* for distinctions between the two types of partitions). If you have an IDE or ESDI drive, use the *diskdefect* program to scan the disk for bad blocks and set up the bad block table. Finally, create the actual BSD filesystems on the disk itself.

Co-residency

BSDI uses the term "co-residency" when describing configurations in which two or more operating systems share a single hard drive. If you do not require this functionality, ignore sections about co-residency and refrain from using any special "FDISK" partitions. Configurations which do not require co-residency also do not require the *bootany* program for choosing which operating system to boot.

Co-residency: FDISK and BSD Partitions

BSD/386 was implemented many years after the original thoughts of disk layout in the DOS world. The ideas behind disk layout in UNIX-like systems pre-date DOS. These two facts combine to complicate discussion of disk layout because of almost-conflicting terminology and arguably baroque layout schemes.

BSD & FDISK Partitions

The large size of modern hard disks creates a powerful argument for dividing them into smaller pieces for management by various software systems. These pieces are called "partitions".

In the PC world, each hard disk is divided into 1, 2, 3, or 4 non-overlapping "FDISK partitions". In the BSD world, as many as eight partitions named 'a' through 'h' describe regions of each hard disk. The mapping among these two partition schemes is conceptually simple but sometimes complex to describe.

Generally, the set of FDISK partitions will exactly span a single disk drive. Likewise, the BSD partitions will exactly span the single disk drive.

BSD/386 and most other system software expect the FDISK partitions to describe the regions of the disk that contain system software and user files associated with a particular operating system. A prototypical system might divide a 150 MB hard drive into two pieces:

- FDISK partition 1: 30 MB for DOS
- FDISK partition 2: 120 MB for BSD/386

Of course, if only BSD/386 is to use the drive, FDISK partitions are a complexity that need not be used.

BSD partition assignments are independent of FDISK partition assignments but usually have a correspondence dictated by practicality. BSD partitions almost always either span a complete FDISK partition (e.g., for accessing DOS filesystems through mtools) or sub-divide it into smaller pieces

See the printcap(5) and ttys(5) man pages for more information.

Parallel Ports

BSD/386 should automatically detect the standard LPT1: parallel port configured at I/O Port 0x378 (it will configure as device lp0). If you have a second parallel port (LPT2:) then you will need to enable the lp1 device in your kernel configuration and rebuild the kernel. (LPT1: for DOS). Here is a segment of a kernel configuration file with both parallel devices enabled:

```
# parallel printer port
# set flags to 0 for CRLF expansion, 1 for raw
device          lp0     at isa? port 0x378 flags 1
device          lp1     at isa? port 0x278 flags 1
```

You should always use "flags 1" or else you will only be able to print plain text documents.

After configuration, follow the directions for configuring, recompiling, and installing your new kernel in "Rebuilding Your Kernel" in the release notes.

This version of BSD/386 allows the parallel port to be used either as a printer or for a Xircom ethernet adapter. To enable sharing, the kernel configuration line should list the port address on the "device lp0" line, plus it should have an additional line like:

```
device          pe0     at isa? port 0          # on lp0
```
or
```
device          xir0    at isa? port 0          # on lp0
```

When the system boots, the port will be configured for the printer. To start using the Xircom, issue an appropriate "ifconfig pe0 ... up" (or "ifconfig xir0..."). command. This will fail if the printer is currently in use or if no adapter is plugged in. If it succeeds, attempts to open the /dev/lp0 device will fail. To return to using the printer, run "ifconfig pe0 down" (or "ifconfig xir0 down").

See the lp(4) man page for more information.

Note on Telebit Modems

Telebit and other Hayes-like modems need to be told to be quiet before you start up a getty on them. Verify that an entry like this is in /etc/remote so you can use tip to talk to the modem:

```
tty00:dv=/dev/tty00:br#9600:pa=none:
```

Then execute these commands as root:

```
# chmod 666 /dev/tty00
# tip tty00
AT       (you say this, maybe several times)
OK       (modem says this)
ATZ&W    (Telebit specific; for others, see manual)
OK       (modem says this)
ATE0Q1&W (Telebit-specific; for others, see manual)
```

```
~.        (leave tip)
```

Edit /etc/ttys and "kill -HUP 1" to start the getty. Note that your actual modem port might not be /dev/tty00; replace this with whatever is right for your site/port/etc.

Also note that if you put E0Q1 into the modem's nonvolatile parameters, you need to undo this in your L-devices chat script, as in "A\dA\dA\dTE0Q0".

The system then displays (if you did not enter the device manually):

```
changing root device to specified_device
erase ^H, kill ^U, intr ^C

*** Don't forget to run setup!

#
```

The # is the superuser prompt. The kernel is in single-user mode and its shell is awaiting your command.

Any time the root filesystem is on floppy or CDROM media (e.g., during installation), you must run the *setup* command after booting. The *setup* command creates a memory-based filesystem into which it copies configuration files for (optional) editing.

```
# setup
Creating MFS filesystem...Done.
Copying files into place...Done.
#
```

Setting Up the Hard-Disks

At this point, local CD-ROM users are running off local CD-ROM root filesystem while others are running off the floppy root filesystem.

The floppy system is extremely small. Below is a *ls -R* of some of the more interesting directories on Boot Disk 2:

```
/bin:
cat        gzcat      mt         rsh        sync
chmod      gzip       mv         sed        tail
cp         ln         pax        sh         test
dd         ls         pwd        sleep      uncompress
gunzip     mkdir      rm         stty       vi

/etc:
disktab      fstab.rea    hosts.real    pwd.db   services
disktab.real group        passwd        rc       spwd.db
fstab        hosts        protocols     resolv.conf

/sbin:
diskdefect init      mount_iso9660 route   umount
disksetup  installsw mount_mfs     scsicmd
fsck       mknod     newfs         setup
ifconfig   mount     ping          slattach
```

Local CD-ROM users have access to the full complement of utilities, including the ones above and more.

Outline Of Hard-Disk Setup

The disk is presumed to be already formatted. If it is not already formatted, return to DOS and do so (in which case you'll be destroying any existing data on the disk). ESDI drives typically require formatting; IDE and SCSI drives are typically pre-formatted.

'target' 0 (BSD/386 internal notation)

sd0 at tg0 unit 0: MICROP 1578-15 rev 17U0 (CCS) 639449*512

Micropolis SCSI disk with 'CCS' SCSI conformance and 639449 sectors of 512 bytes; **either sd0 or wd0 is required for booting**

tg1 at eaha0 target 1

sd1 at tg1 unit 0: MAXTOR P0-12S rev JB21 (CCS) 2042264*512

hard disk

tg2 at eaha0 target 3

sd2 at tg2 unit 0: MaxOptix Tahiti 1 rev E6 (CCS) removable media, secsize 512

Optical disk

tg3 at eaha0 target 4

st0 at tg3 unit 0: ARCHIVE VIPER 1500 rev 2.2G (SCSI-2) streaming (512) tape

QIC tape drive; a QIC-150 or 8mm tape required for local tape installation; Non-SCSI tapes are device wt

tg4 at eaha0 target 5

st1 at tg4 unit 0: EXABYTE EXB-8200 rev 251K (SCSI-1)

8 mm tape drive

tg5 at eaha0 target 6

sd3 at tg5 unit 0: TOSHIBA CD-ROM DRIVE:XM rev 3232 (CCS) removable media, secsize 2048

SCSI CD-ROM drive

we0 at isa0 iobase 0x280 irq 9 maddr 0xd0000

we0: WD8013EB address 00:00:c0:cc:f7:28

Western Digital/SMC style Ethernet interface; other Ethernet controllers have device names ne, eo, ex, ef, el, ep, and pe

→ REMOTE LOADING ←

Note your Ethernet controller name (we0, ne0, eo0, ex0, ef0, el0, pe0, or ep0) for future reference when you configure your network.

→ IMPORTANT NOTE ←

If a critical device is not displayed during configuration, you should check for hardware problems. You can not proceed until the necessary devices configure correctly.

Specifying a Root Device

After configuration, BSD/386 pauses for input after printing:

```
root device?
```

Respond with the correct root device from this table:

- Local SCSI CD-ROM: ***sd**N* (insert correct digit for CDROM drive from configuration, e.g., sd1)
- Local Mitsumi CD-ROM: ***mcd0***
- Tape or Remote CD-ROM: ***fd0***

Dialup IP Services: PPP and SLIP

SLIP and PPP are two serial protocols that enable serial devices to communicate using standard IP, thus extending the paradigm of networked services (e.g., electronic mail, remote login, file transfer) from Ethernet links to inexpensive serial links which can form a wide-area network.

PPP Configuration

As shipped, the BSD/386 kernel is not configured for PPP. Add the following lines to the kernel's configuration file to enable the PPP drivers (see the chapter on re-building kernels to choose the correct configuration file):

```
options PPP
options PPP_DEBUG
pseudo-device  appp    4  # requires options PPP
```

Then rebuild and install your new kernel per the instructions elsewhere in the release notes. The PPP_DEBUG option helps you find problems when PPP is not performing correctly.

Verify that the PPP drivers are installed by running "netstat -i" and finding PPP's <Link> entries (while running the new kernel, of course). For example:

```
% netstat -i
Name  Mtu   Network Addr. Ipkts Ierrs Opkts Oerrs  Coll
ppp0* 1500  <Link>            0     0     2     0     0
ppp1* 1500  <Link>            0     0     0     0     0
ppp2* 1500  <Link>            0     0     0     0     0
ppp3* 1500  <Link>            0     0     0     0     0
```

Later, when you have PPP configured and working properly, you will want to switch off the trace information by adding "pppconfig ppp0 -trace" to the /etc/netstart file. You can disable it completely by removing "options PPP_DEBUG" from your kernel configuration but that makes it a little harder to debug if something ever goes wrong.

The manual pages you will want to read for PPP configuration are:

ppp(1)	The PPP daemon program (for automatic dialup)
pppattach(8)	Attach serial lines as PPP network interfaces
pppconfig(8)	Configure Point-to-Point Protocol parameters
ppp.sys(5)	PPP configuration file
ppp(4)	Point-to-Point Protocol (PPP)
appp(4)	asynchronous HDLC encapsulation for PPP

The configuration files you will need to edit are /etc/netstart, /etc/ppp.sys, and /etc/hosts (if not using DNS).

Below is a sample /etc/ppp.sys file, which you will need to customize for your site. It is a standard 'termcap' style file, with equal signs ('=') for alphabetic assignments and pound signs ('#') for numerical assignment. You'll need to change the the phone number, login, and password, of course:

```
# everything you need to define a modem connection
# the name 'modem0' is ref'd by 'dialout', below

modem0:\
:du:at=hayes:dv=/dev/tty00:br#38400:

# Example entry ``Pbsdi'' for dialup PPP to BSDI
# using modem0 on ppp0.  Choose an appropriate name
# and parameters for your file.
# Set :if: to your ifconfig flags (like /etc/netstart)

Pbsdi:\
:in#0:id#600:do:\
:if=inet MYHOST_IP RMTHOST_IP netmask 0xffffff00:\
:pn=REMOTEPHONENUMBER:s0=^M:\
:e0=ogin\::s1=YOURLOGIN^M:\
:e1=word\::s2=YOURPASSWD^M:\
:tc=modem0:
# tc means 'go read the rest of the parameters at
# another place'

# Define a  dial-in account from system hilltop;
# this name is referenced in the password file

Philltop:\
:in#1:di:if=inet MYHOST0_IP RMTHOST0_IP netmask 0xfffffffc:
```

Be sure and substitute the correct items above for your site. You will also need to fix the netmask to match your network.

If you aren't running a nameserver then make sure all hostnames you use are correctly configured in /etc/hosts.

To start the ppp daemon (to call and connect to a remote system) at boot time, add the following near the end of your /etc/netstart file:

```
# Start the PPP daemon
ppp -d REMOTE_SYSNAME &
```

Make sure the REMOTE_SYSNAME matches the name of a host listed in /etc/ppp.sys. You will need to start one PPP daemon for each outgoing host.

If your dialup PPP system connects to the Internet, then you will want to setup a default route to that address so that packets destined for systems no on your local network will be sent to the Internet:

```
route add default REMOTE_HOST
```

where REMOTE_HOST is the name of the host to which you are connecting. If you aren't running a nameserver then make sure that the name you use for REMOTE_HOST is in /etc/hosts.

Once you complete configuration and start the PPP daemon, try pinging the remote system:

```
% ping REMOTE_SYSNAME
```

If no replies are received, then take down PPP (use ps to find the daemon and

isa0 (root) *Autoconfiguration root, always printed (with extra 'eisa' notation on EISA machines)*

pccons0 at isa0 iobase 0x60 irq 1: color, 8 screens
Console device – **Required**

com0 at isa0 iobase 0x3f8 irq 4: buffered
com1 at isa0 iobase 0x2f8 irq 3: buffered
Serial ports (called COM1 and COM2 under DOS; 'buffered' indicates an NS16550 UART)

fdc0 at isa0 iobase 0x3f0 irq 6 drq 2
Floppy controller – **Required**

fd0 at fdc0 slave 0: 1.44M HD 3.5
Floppy drive 0 – **3.5" Required**

lp0 at isa0 iobase 0x378 irq7
Standard parallel printer port at LPT1 (LPT2 is not available in the GENERIC kernel)

pe0 at isa0
Xircom Pocket Ethernet-II on lp0. The adapter does not need to be plugged in until you run the ifconfig command.

xir0 at isa0 at lp0 (at 0x378)
Xircom Pocket Ethernet-III on lp0. The adapter does not need to be plugged in until you run the ifconfig command.

fd1 at fdc0 slave 1: 1.2M HD 5.25
Floppy drive 1

wdc0 at isa0 iobase 0x1f0 irq 14
ESDI/IDE/RLL/MFM hard disk controller

wd0 at wdc0 slave 0
Hard disk; **either sd0 or wd0 is required for booting**

wd1 at wdc0 slave 1
Hard disk

wt0 at isa0 iobase 0x300 irq 5 drq 1
Wangtek QIC-02 cartridge tape interface

npx0 at isa0 iobase 0xf0
Floating point co-processor (no IRQ for i486 internal FPU)

vga0 at isa0 iobase 0x3c0 maddr 0xa0000-0xaffff
Display (will always say vga no matter what is present)

eaha0 at isa0 iobase 0x5000 irq 15: ADAPTEC AHA1742A (ENHANCED) rev A
SCSI adaptor (EISA); ISA adaptors have device name 'ahaN'

mcd0 at isa0 iobase 0x334 irq 9 drq 3 firmware version M2
Mitsumi 8-bit ISA CDROM drive

mcd0 at isa0 iobase 0x334 irq 9 drq 6 firmware version M4
Mitsumi 16-bit ISA CDROM drive

eaha0: delaying to permit certain devices to finish self-test
This is a standard message. Each SCSI target yields two lines: target detection line and the device configuration line.

tg0 at eaha0 target 0

The boot program will print the size of the text, data and bss as it loads the kernel:

```
568480+39292+91064 Start 0x0
```

(the numbers may vary from this example) and then print:

```
File loaded-- change diskette if necessary, then hit return
```

The kernel has not yet committed to mounting the root directory. Unless you are installing from local CD-ROM, remove Disk 1 and insert Disk 2.

Type carriage return; the kernel will continue booting.

Boot Debugging

It is possible to boot a kernel with an option to obtain additional information about the autoconfiguration process, for example to find out why a device does not configure. It is also possible to control some aspects of the autoconfiguration process with this option, such as inhibiting some devices from being probed. To do this, boot the system, pressing a key to interrupt the loading of /bsd if necessary to get a Boot: prompt. Then, enter the -autodebug command, followed on a separate line by the kernel to be loaded, for example:

```
Boot: -autodebug 1
Boot: /bsd
```

A value of 1 prints detailed information about autoconfiguration, pausing after each screen. A value of 2 causes the kernel to wait for a y/n response before probing each device, skipping the probe if the response is n.

Configuration Report

As the kernel boots, it will print a banner and some configuration information.

The CPU speed printed is an estimate based on how fast the CPU can count. It is for informational purposes and will not affect operation of BSD/386. If the reported speed is way off (e.g., reports 8 MHz for a 33 MHz machine), you might check the 'Turbo' switch and/or the configuration of your caches and bus. Reporting 25MHz or 50MHz for a 66MHz is well within the expected variation for the speed reporting heuristic.

The numbers reported for real and available memory tell how much memory BSD/386 detected. The number for real memory includes the 384K I/O memory space, although the system does not use it as normal memory.

The subsequent lines report the devices found and their configurations. If you are missing a required critical device or have configured it incorrectly, you can find out by reading the configuration lines.

Note that some SCSI devices may take a while to reset (e.g., Exabyte 8mm tapes can take 20-30 seconds to become 'ready').

Here's a sample list of device names (the first part of each line is shown in exactly the format reported during boot; your configuration will probably be different):

kill it) and add the -x option to the ppp command as you reinvoke it (so you can see what it is doing). If the ppp daemon is connecting to the remote site and then failing, you'll need to look at the ppp trace output (which goes both to the console and is saved in /var/log/messages). Hopefully, between the two of them you will get enough PPP debug information to trace the source of the problem.

Note that you can observe the transmit and receive lights on modems to see when data is being sent. If, when you ping the IP address of the remote host that is your PPP connection and you see outgoing traffic on your end but nothing ever returns, then the two PPP implementations are not interoperating. Discuss it with your PPP provider, your PPP vendor, and ultimately with the BSDI service line.

Another program, pppattach(8), sets up hardwired PPP connections (the ppp(1) daemon can be used as well). Dialing by hand and using pppattach(8) can also be useful while debugging to make sure that the dialing process is not causing your problem. The pppattach(8) program simply enables PPP on a device, see the man page for details on usage.

To configure a dialin PPP account so others can connect to your system, set up an /etc/ppp.sys entry for the site (see the example above). Then add an account with the group id the same group as the /usr/bin/ppp command (118 in the 1.1 distribution). For example (broken across two lines for display purposes):

```
Psysname:xxx:2000:118::0:0:sysname PPP:
                           /var/ppp/sysname:/usr/bin/ppp
```

You will need to create the login directory (/var/ppp/sysname is just an example).

SLIP Configuration

The startslip(1) command is used to initiate an outgoing SLIP connection. You can use configsl(1) to generate a /usr/local/bin/slipup script with all the required information. Configsl prompts you for the information that it requires.

To setup a dialin SLIP connection see the sliplogin(8) man page. Here are some sample files to get you started:

Sample /etc/master.passwd fragment (broken for display):

```
Ssysname:*:100:1:sysname SL/IP:
                        /var/slip/sysname:/usr/sbin/sliplogin
```

You will need to create the login directory (/var/slip/sysname is just an example).

Sample /etc/slip.hosts:

```
# SLIP Hosts initialization file

# login   localaddr  remoteaddr  netmask       opt_args
#-------  ---------  ----------  -------       --------
Ssysname  myhost     sysname     255.255.255.0 autocomp
```

Sample /etc/slip.login:

```
#!/bin/sh
slipunit=$1; shift
speed=$1; shift
login=$1; shift
localaddr=$1; shift
remoteaddr=$1; shift
netmask=$1; shift
opt_args=$*;
/sbin/ifconfig sl${slipunit} inet $localaddr \
                    $remoteaddr netmask $netmask up
logger -s -t slip \
  "Attached unit $slipunit for $login \
(speed: $speed) (args: $opt_args)"
exit 0
```

Sample /etc/slip.logout:

```
#!/bin/sh
slipunit=$1; shift
speed=$1; shift
login=$1; shift
localaddr=$1; shift
remoteaddr=$1; shift
netmask=$1; shift
opt_args=$*;
logger -s -t slip "Detached unit $slipunit from $login"
exit 0
```


/etc/sendmail.fc	no more frozen config file or "sendmail -bz"
/etc/termcap	now a symlink to ../usr/share/misc/termcap
/dev/MAKEDEV	device changes for V1.1 (merge any local changes)
/dev/MAKEDEV.local	replace with current local version (no BSDI changes)
/usr/X11/lib/X11	app-defaults, etc.
/root	Shell startup files
/usr/share	Local changes?

These files in /var may require attention:

/var/account/*	New format; do not install V1.0 versions
/var/db/libc.tags	do not reinstall V1.0 version this information refelects the /usr/src tree
/var/log/cron	cron now uses syslogd (you can remove /var/log/cron if you no longer need the data)
/var/db/locate.database	execute "sh /etc/weekly.local" as root for udpate
/var/games/hackdir/data	reinstall these from your V1.0 saveset
/var/games/hackdir/help	
/var/games/hackdir/hh	
/var/games/hackdir/perm	
/var/games/hackdir/record	
/var/games/robots_roll	
/var/log/lpd-errs	
/var/log/maillog	
/var/log/messages	
/var/spool/at/at.counter	

Be sure you've saved any old data you might need (/etc/master.passwd, configuration files (including configuration files for system regeneration), /etc/netstart, /etc/fstab, /etc/disktab, /etc/resolv.conf, /etc/kerberosIV, Xconfig, that sort of thing).

Boot BSD/386 From Floppy

Start installation by inserting the BSD/386 floppy labeled 'Disk 1'. Insert the CD-ROM or tape in the drive from which it will be accessed.

Push reset (or cycle the power) and wait for diagnostics to finish. The floppy drive should spin for a short period of time. The display will print the following prompt:

```
loading /boot............
Boot:
```

At this point, you should enter a carriage return.

/etc/contrib/ircd/*	sample files
/etc/mtree/*	mtree files
/etc/uucp/*	UUCP Configuration Files
/etc/namedb/root.cache	latest root cache data
/etc/namedb/*	Other namedb files

Replace the 1.1 distribution version of these files with your current copy (no significant changes from 1.0 version, minor changes are noted):

/etc/aliases	
/etc/crontab	
/etc/disktab	changed comments
/etc/dm.conf	
/etc/exports	
/etc/fstab	
/etc/fstab.sample	new examples
/etc/hosts	made localhost.my.domain the canonical name
/etc/hosts.equiv	removed all lines
/etc/hosts.lpd	
/etc/kerberosIV/*	Kerberos IV configuration files
/etc/mail.rc	
/etc/master.passwd	added nonroot and changed demo shell to bash
/etc/monthly	
/etc/monthly.local	
/etc/motd	
/etc/mtoolsrc	fixed support for 720K and 360K diskettes
/etc/named.boot	should be unchanged
/etc/networks	
/etc/ntp.conf	
/etc/ntp.drift	
/etc/phones	sample phone numbers changed
/etc/ppp.sys	should be unchanged
/etc/protocols	
/etc/rpc	
/etc/security	
/etc/shells	
/etc/weekly.local	

These files require you to update or deal with them:

/etc/Xconfig	was /usr/X11/lib/X11/Xconfig
/etc/dumpdates	You will need to redo level 0 dumps for system related filesystems (/ /usr /var)
/etc/ftpaccess	new: look in /etc/ftpd_samples for starters
/etc/ftpconversions	new: look in /etc/ftpd_samples for starters
/etc/ftpdusers	new: look in /etc/ftpd_samples for starters
/etc/localtime	copy again from /usr/share/zoneinfo/
/etc/passwd*	run "pwd_mkdb -p /etc/master.passwd"
/etc/sendmail.cf	reconfigure /usr/share/sendmail/cf (see release notes)

Helpful Hints

How To Increase the Number of Pseudo-Teletypes (ptys)

If you have a large number of network users, you might wish to increase the number of pseudo-teletypes (ptys). To increase the number of available pseudo-teletypes from 32 (the default) to 64, change the following line in your kernel configuration file:

```
pseudo-device   pty
```

to read:

```
pseudo-device   pty 64
```

After reconfiguring, rebuilding, and reinstalling your new kernel execute the following commands as root to create the new devices.

```
# cd /dev
# ./MAKEDEV pty2 pty3
```

Add the new devices to the /etc/ttys file and reboot.

See the ttys(5) and pty(4) man pages for more information.

Shutting Down

Due to the nature of BSD/386's high-speed cached filesystems, simply switching off or rebooting the computer without flushing the cached data to disk can lead to filesystem data loss.

To shutdown from multi-user mode, use the reboot(8) or halt(8) commands as root. To reboot the system execute:

```
# reboot
```

To halt the system execute:

```
# halt
```

Use the shutdown(8) command to bring the system down to single user mode.

See the shutdown(8), halt(8), reboot(8), fasthalt(8), and fastboot(8) man pages for more information.

Virtual Consoles & Multiple Text Displays

There are several different ways to get multiple text displays in BSD/386. The most straightforward is to enable some of the ttyc* devices in /etc/ttys (which enable 'multiple consoles'). This will enable you to type Alt-Function Key to switch among different screens (a feature not supported on monochrome display adapters). Alt-Function-key-1 shows you the primary console, Alt-Function-key-2 gets you ttyc2, etc. The display switch is immediate.

BSD/386 also provides two programs that give you multiple text windows on any 'teletype' style display (including glass teletypes) by using the standard pseudo terminal drivers:

screen – screen manager with VT100/ANSI terminal emulation
window – Window implements a window environment on ASCII terminals

Screen is more like virtual consoles and window actually lets you see the

output from two different sessions at the same time (you are really running a simple windowing system that happens to be text based). The screen program is particularly flexible in that you can 'detach' a screen and later re-attach it to the same physical screen or to a different one.

You can choose whichever scheme fits your needs or mix and match (e.g., you can run a screen or window session on each virtual console).

See the ttys(5), pccons(4), cons(4), screen(1), and window(1) man pages for more information.

X Window System Configuration

The X Window System is a very large collection of code. Usually, the first thing you will need to get working is the X server, which is the daemon that controls the window display on your PC. Most sites should be able to get going by simply running configX386 and answering the questions.

If this fails, there are also some files in /usr/X11/lib/X11/etc that explain some of the details about the X configuration file and the man pages X386(1), X(1), and Xserver(1) are also good sources of information.

Once you get past configuring the X server you can look forward to many happy hours of adjusting 'app-defaults' files. One of BSDI's service people has collected about 140K of app-defaults files in his home directory! These enable you to change the "look and feel" of the X applications. Applications should run without any customization but it can be fun, on a rainy afternoon perhaps, to setup XTerm with a chartreuse background and lime green letters.

See the configX386(1), X386(1), X(1), Xserver(1), xinit(1), and startx(1) man pages for more information.

Multi-headed X displays

The X server supports multiple display devices. If you want to use a VGA adapter and something like an 8514/A together (e.g., an ET4000 with an ATI Ultra Pro), then you must be able to configure the cards so they do not overlap (e.g., disable the VGA portion of your 8514/A card). Then simply list both kinds of devices in the Xconfig file, for example:

```
vga
        Modes   "1024x768"
wga
        Modes   "1280x1024"
```

The first one in the Xconfig file becomes the primary display. You can include full graphics driver options in each section (chipset, clocks, etc).

See the X386(1) man page for more information.

Two more floppies contain DES and Kerberos. Due to US export regulations, they are only shipped to customers in the US and Canada. International distributors may ship a compatible substitute.

Check Manifest

Be sure you have everything you need. This includes:

- BSD/386 3.5" floppy labeled Disk 1
- BSD/386 3.5" floppy labeled Disk 2
- One of:
  - CD-ROM media
  - QIC-150 tape
  - 8mm (Exabyte) tape
- BSD/386 Floppies labeled ENCRYPTION and KERBEROS (US/Canada only)
- License agreement
- This document

Upgrading From Previous BSD/386 Releases

If loading the new release on top of the old one, you probably wish to backup all the files in the /etc, /var, and /usr/local directories. If you have made any changes in the / or /usr directories, you may wish to save them, as well. Likewise, be sure to save any Xconfig files you have modified. Of course, the prudent administrator would backup user directories, too.

Upgrading: Files to Save

BSDI has modifed several files that a system owner may also have modifed. Here's a list so that you can avoid losing local changes when installing software:

/etc/daily	cleanup scripts moved to daily.local
/etc/daily.local	cleanup scripts moved to here from daily
/etc/gettytab	pccons and other changes (new getty)
/etc/group	added "netdial:*:118:"
/etc/inetd.conf	changed rlogind -a, ftpd -l -A
/etc/magic	gzip compressed changed
/etc/man.conf	update for new man program
/etc/netstart	more examples, hpp/multicase/3C579
/etc/printcap	new examples, extended HP and PostScript support
/etc/rc	vi preserve, chkconfig, added rwhod flags
/etc/rc.local	xntp, motd perms, misc
/etc/remote	new examples
/etc/services	kpasswd changed from 761 to 751
/etc/syslog.conf	added "cron.info/var/log/cron"
/etc/ttys	use "pccons" for console getty
/etc/weekly	rotate cron log and other uucp changes
/etc/amd_samples/*	sample files
/etc/ftpd_samples/*	sample files

Software Installation

Typing on the BSD/386 System

Type on the BSD/386 console just as you would other computer consoles. Use backspace to erase a single character in the case of an error. Use control-U to erase the current input line in the case of a big typing error. Terminate programs by typing control-C.

Preparing for Install

The first step for installation is to settle in a for a couple hours (between 0.5 and 4 hours). Don't try to do this in a hurry. Most of the time is waiting for the tape or CD-ROM to spin.

Prepare your system. Any cards that you need to use for loading MUST be configured as described previously. If your disk requires a low level format (ESDI disks do; IDE disks don't; SCSI disks usually come preformatted but might require formatting), use DOS or whatever to format the disks. If you do not intend to share the disk with DOS, format with native mode (no DOS cylinder or head mapping) and with no interleave.

The hard-disk that BSD/386 will boot from should be unit 0 ('C:'). The 3.5″ floppy used during installation should be unit fd0 ('A:').

Outline of Installation

Installation proceeds from one of four devices:

- Via local tape
- Via remote tape
- Via local CD-ROM
- Via remote CD-ROM

The installation guide contains but a few 'forks in the road'. For the most part, installation from each the above devices proceeds along the same path. Differences are noted prominently below.

Basically, installation proceeds like this:

- Boot BSD/386 from a floppy
- Mount a root filesystem (either from floppy or CD-ROM)
- Set up the disks (may require some customization)
- Load the disks
- Customize configuration files
- Reboot from the hard drive

About the Floppies

BSD/386 uses one or two boot floppies in its boot process (another pair of floppies contains DES/Kerberos code in the U.S. and Canada).

The first boot floppy contains the operating system itself. The second boot floppy contains the utilities necessary to set up the disks and load the distribution.

About /etc/group

This table explains what the various groups in /etc/group are used for:

wheel	access to su
daemon	for system daemons and their files
kmem	for processes who read /dev/kmem
sys	read access to sources
tty	programs that access teletypes
operator	for those who perform dumps
uucp	for uucp and other processes that use serial lines
bin	owns binaries
news	for the news system
utmp	for processes that write the utmp file
games	security for games
dialer	outgoing modem access
netdial	provides security for PPP & SLIP
nogroup	should never have write permission

Additional Documentation

Additional system documentation is available online. The /usr/share/doc/bsdi directory holds BSDI documentation. The file config.n describes kernel building and configuration in detail. The file driver.n gives hints on writing device drivers. These files are also available via the 'man' command:

```
% man bsdi config
% man bsdi driver
```

Of course, the man pages yield a wealth of information. Use the 'man -k' command to find commands about various topics. Consider this example which shows how to find the floppy formatting program:

```
% man -k floppy
fd (4) - floppy disk driver
fdformat (1) - format floppy disk
```

Two books describe BSD/386 internals to an extent: *The Design and Implementation of the 4.3 BSD UNIX Operating System*, Leffler, McKusick, Karels, Quarterman, ISBN: 0-201-06196-1 and *The Design and Implementation of the 4.3 BSD UNIX Operating System Answer Book*, Leffler & McKusick, ISBN: 0-201-54729-9.

Installation Media Format

The first file on the tape, or the /cdrom/PACKAGES/PACKAGES file on the CD-ROM is the manifest. The file is ascii (it has comments at the top to explain the various fields). It is null extended to a size of 10k, but you should be able to use vi or emacs to make a copy more reasonable. In the tape version, the first column is the file number on the tape (0 origin), so you can pass that number directly to 'mt'. The directory field tells you what directory you should be in when you extract the archive.

The commands for extracting a normal archive are:

```
% dd if=file bs=10k | pax -rvp e
```

The commands for extracting a compressed (type z in the manifest) are:

```
dd if=file bs=10k | gunzip | pax -rvp e
```

Regularly Scheduled Maintenance

The default /etc/crontab file will execute the /etc/daily, /etc/weekly, and /etc/monthly shell scripts daily, weekly, and monthly. The default scripts do genernal maintenance, rotate logs, rotate accounting records, etc. They also invoke the files /etc/daily.local, /etc/weekly.local, and /etc/monthly.local for local procedures. As distributed, the local versions contain some general templates for operations many administrators like to perform (e.g., removing old core files, cleaning /tmp, etc.) but they must be explicitly enabled for this to occur. The script output is saved in /var/log/{daily,weekly,monthly} and is sent via email to root.

the standard port address (0x1f0). To switch over to auto-booting from a SCSI disk, you must enter the CMOS set up menu and set the primary hard disk configuration to “Not Installed”. This does NOT affect BSD/386’s ability to configure the primary hard-disk which will still be accessible through /dev/wd0[a-h].

Disk Space Requirements

BSD/386 loads its software in modules called 'packages'. As such, you can choose a small subset of software to load and use. For minimal operation, you can get away with a very small disk.

MB	Package	MB	Package
5.4	Core root	2.5	Doc
15.9	Core /usr	4.8	Games
0.2	Core /var	9.3	Mh
7.5	/usr	4.3	Ghostscript
4.2	Networking	9.5	Emacs
11.3	Development	9.1	Tex
8.8	Man	4.0	Flexfax
2.9	Groff	13.2	Isode
18.1	Contrib /usr	6.9	Sys Source
19.5	X11 Server	9.1	Man Source
27.4	X11 Clients	5.4	Sys Xtrasource
3.8	X11 Development	227.0	Utils Source
4.3	X11 Man	117.9	X11 Source
15.6	X11 Contrib		

Only available in US/Canada:

MB	Package
0.6	DES
3.0	Kerberos
1.2	Domestic Source (DES & Kerberos)

The tables show approximately how much space each of the distribution pieces requires (the final distribution may be slightly different). After a table of contents, each of these packages is stored in a separate file on the distribution media.

If you choose to load the sources, remember to reserve enough disk space to hold the .o files and the linked binaries (if you intend to perform software builds).

Booting SCSI Disks

To auto-boot from a SCSI disk, the disk must be set to target 0, LUN 0. It is possible to load and run programs on other targets, LUNs and partitions, given an already loaded standalone boot program; the four boot parameters are used as follows:

```
Boot: sd(adaptor-type,target,0,partition)program
  or
      aha(target,0,partition)program
  or
      eaha(target,0,partition)program
```

SCSI auto-booting is inhibited if a PC hard-disk is configured by the BIOS at

Kernel Debugging

If you have a system hang or panic, please gather the following information and send it to support@bsdi.com (or print it out and fax it to the fax support line if you do not have email access). Of course, if you can think of any other facts or data might be relevant, please include them.

The first thing we will need is a complete system configuration; the information contained in /var/db/dmesg.boot is a good start, but we also need cpu model and type (e.g., Intel 486DX2/66), etc. Please also list any cards (and their port and irq configuration) that are not found by BSD/386 at auto-configuration time (e.g., sound cards).

Preparation

You will need a kernel compiled with debugging symbols. To create it, you use the -g option when you configure your local kernel (e.g., "config -g LOCAL"). If you cannot easily generate a new kernel or cannot reproduce the problem, then go ahead and gather the following information using /bsd in place of bsd.gdb. If you can reproduce the problem please generate a new kernel using "config -g LOCAL" and use the resulting bsd.gdb kernel for debugging. If you are running the GENERIC kernel and have the CDROM handy there is already a bsd.gdb on the CDROM in /cdrom/usr/src/sys/compile/GENERIC/bsd.gdb.

When the system panics it will save the crash information in your swap area. To recover this information on reboot, you must have created the directory /var/crash (or made it a symbolic link to a directory where you have a lot of free space). The command "savecore /var/crash" is executed when the system reboots to save the crash dump information in /var/crash, e.g., (where # is 0, 1, 2, ...): /var/crash/bsdcore.# and /var/crash/bsd.#. You'll need at least as much free space in /var/crash as you have main system memory (plus room for the kernel bsd.#).

Debugging System Problems

Does the system panic or just hang? If it panics, then skip ahead to "Debugging Crash Dumps".

If the system is on a network, can you ping it after the hang [yes, no, n/a]?

Do characters echo when typed on the console [yes, no]? [If you cannot reproduce the hang while not running X, let us know.]

Run /bin/csh on the console and execute "stty status ^T", then cause a hang and try typing ^T (control-T). Do you get output like:

```
load: 0.16  cmd: cat 5935 [ttyin] 0.00u 0.02s 0% 0k
```

If so, include this information. If not, let us know.

Does ctrl-alt-del cause a system reboot [yes, no]? [If not then you'll need to contact BSDI support for further instructions.]

If you can reboot with ctrl-alt-del then you need to add "options DEBUG" to your kernel configuration file. Then when the system hangs, you will be able to use Ctrl-Alt-Del to generate a crash dump (it will ask do you want to reboot, you type 'y', then it will ask you if you want to create a crash dump).

Notes on Configuring and Debugging EISA Machines

Some 486/50 EISA machines have bad external caches, if you have problems try turning it off in the CMOS setup. You must run the EISA config utilities any time you change EISA hardware or add more memory.

Debugging Crash Dumps

The "script" command will leave the file "typescript" in your current directory that contains all the output from the commands you type. Please execute the following commands and then mail the resulting typescript file to support@bsdi.com along with the other information you have gathered.

You can get a kgdb.tricks.ps file via anonymous ftp to ftp.bsdi.com in bsdi/support/misc/kgdb.tricks.ps. If you do not have anonymous ftp access then just skip the "source kgdb.tricks.ps" and "ps" commands for now.

```
% script
Script started, file is typescript
% gdb -k /usr/src/sys/compile/LOCAL/bsd.gdb \
                                   /var/crash/bsdcore.0
(kgdb) source kgdb.tricks.ps
(kgdb) x/i 0xfe000000  <- where fe000000 is the eip value from the panic
(kgdb) print curproc
...                                           [gdb output here]
(kgdb) print *curproc
...                                           [gdb output here]
(kgdb) bt
...                                           [gdb output here]
(kgdb) ps
...                                           [gdb output here]
(kgdb) quit
% exit
Script done, file is typescript
```


failure occurs in almost all uses of the chip, including DOS. Contact Cyrix at 800-848-2979 for instructions on upgrading your co-processor.

BSDI has also been informed that there is a potential problem with the ULSI 387 coprocessor (though none of BSDI's customers has reported the problem). One user of a similar operating system reported that his system simply hung when using that 387 chip. He says, "ULSI...[is] aware of the problem". Their suggested fix: exchange the chip for a Cyrix equivalent.

BSDI has found that many 486/50 EISA systems have poor implementations of the external cache (no problems are seen, though, with other clock speeds or with the 486/50 ISA bus systems). The typical symptom is that a system runs perfectly well for a while (usually minutes) and then crashes (or processes terminate) for no apparent reason. Subsequent execution of the same process(es) results in success. If you are seeing these results, use your hardware set up menus to disable more and more caches. If your system works with the caches disabled, this is a strong indicator that the hardware has problems.

Configuration Conflicts

We've found that some PC hardware configurations are mutually exclusive and cannot be used in the same machine. In other cases, some hardware combinations may limit the address and interrupt configuration possibilities of a particular card. While we've made every effort to make our product, BSD/386, as flexible as possible in these situations, there are cases where technology fails and old fashioned trial and error succeeds. We urge you to be patient and persistent when configuring interface cards in your machine. We suggest configuring devices in the following order, keeping a map of used I/O ports and IRQs along the way:

1 Floppy drive controller
2 Fixed drive controller(s)
3 COM ports 1 & 2
4 Ethernet controller
5 Mouse (if applicable)
6 COM ports 3 & 4 (if applicable)
7 Other devices

Note that we've also experienced cards that play tricks on us: devices which have the option of reading their configuration from EEPROM, but sometimes this EEPROM information becomes corrupted by another card's power-on initialization sequence, reset due to a port/irq conflict, etc. If you think this is happening to you, verify that you can boot DOS (from power off) multiple times and have the card retain the same configuration as reported by its configuration utility. If not, you might try using the card at its default addresses or, on some cards, configure the card with jumpers rather than the EEPROM.

modes are very similar and easy to confuse. In ‘diagnostic’ mode, you can set the configuration but it doesn’t set the power up defaults. In ‘configuration’ mode, you do indeed save the configuration.

CMOS Setup

The BSD/386 kernel uses the CMOS settings to distinguish between 3.5″ and 5.25″ floppy drives. If the CMOS setting is wrong, the kernel will boot and then panic after the configuration messages. The configuration line from the kernel when it locates the floppy drives will show you what the CMOS entry says so you can verify that it is correct. If it is incorrect, change the CMOS settings to match the actual hardware configuration. For BSD/386, the 3.5″ floppy drive must be configured as drive ‘A:’ (fd0).

The 1740 Family SCSI Adaptors

The Adaptec 1740 family of SCSI host adapters can be configured in either ‘standard’ (1540B-compatible) or ‘enhanced’ (EISA) modes. BSD/386 supports both modes; enhanced mode should give better performance. Use the EISA Configuration Utility (via DOS) for your system and the configuration floppies provided with your 1740 to set your adapter to the desired mode. Sometimes the default transfer rate of 10 MB/s in enhanced mode won’t work for your particular SCSI bus; if at first you don’t succeed, try a lower rate.

Texel CDROM Drives

Early Texel CDROM drive firmware did not function with the Adaptec SCSI adaptors. Recent firmware has resolved this problem.

Special Chip Problems

A small number of Intel 80486DX-25 chips from particular production runs have internal problems that appear when running operating systems that use virtual memory. These chips appear to work perfectly under DOS, but with BSD/386, they cause occasional random core dumps or hangs. These chips are marked with part number 80486DX-25 and the shipping spec number of S X243, S X249, or S X250. They also have an Internal Lot Number that is a letter followed by 7 or 8 decimal digits.

If you have one of these chips, please remove it from the motherboard (carefully grounding yourself first) and photocopy the chip face. Then call Intel’s primary number 800-628-8686 and tell them that you have an architectural problem with your chip and that you need it replaced. Read this paragraph and explain to them, “This chip comes from a stepping that has virtual memory problems. Rob Kolstad discussed them at length with David Coleman.” If this doesn’t help, please call Rob at 719-593-9445.

BSDI has been informed that if you have Cyrix’s very old Cx-486-DLC CPU (old versions of the CPU) and very old Cx-87D87 (with serial number is A3xxxx, B3xxxx or C3xxxx) then you should upgrade the co-processor). The

Domain Name Service (DNS, named)

DNS is a hierarchical, distributed database for resolving hostnames to IP addresses (and vice versa). The daemon that performs this service is named(8). If you aren’t going to configure DNS, then you will need to explicitly list every hostname that you intend to use over the network in your machine’s /etc/hosts file (see hosts(5)).

Local Cache

The recommended way to configure DNS for non-primary systems is as a simple caching server (obviously the primary name server for your domain will need a more complex configuration). This will give you a local cache of commonly used IP addresses so that if other nameservers go down you can still resolve those hostnames to IP addresses.

```
; simple caching named doing its own lookups
directory  /etc/namedb
cache      .                     root.cache
primary    0.0.127.IN-ADDR.ARPA  localhost.rev
; forwarders 99.99.99.99
; slave
```

You will need to edit /etc/namedb/localhost.rev and change the domain names to your domain.

If you have a local, reliable server with a richer cache (mail server, etc.) then you might want to forward requests to it but still maintain your own local cache. To do so uncomment the following lines from the above sample:

```
forwarders 99.99.99.99
slave
```

And replace 99.99.99.99 with the IP address of your server. In practice, you should never have more than one forwarder listed (otherwise you can generate a lot of network traffic and make name lookups even slower).

The forwarding facility is useful to cause a large site-wide cache to be generated on a master, and to reduce traffic over links to outside servers. It can also be used to allow servers to run that do not have access directly to the Internet, but wish to act as though they do (in which case you must use "slave", however, it is recommended anyway).

One other thing you might want to add to your configuration is:

```
secondary YOURDOMAIN 99.99.99.99 SAVEFILE
```

Where YOURDOMAIN is your domain name (e.g., bsdi.com), 99.99.99.99 is the IP Address of the primary server for that domain, and SAVEFILE is the name of a file in /etc/namedb/ where you wish to cache the DNS information for that domain. Add secondary lines for domains you frequently use as it further protects you from nameserver outages. You should also add a secondary line for the relevant IN-ADDR.ARPA. domains (for gethostbyaddr(3)):

```
secondary 139.124.192.IN-ADDR.ARPA 99.99.99.99 SAVEFILE.rev
```

This example would make you a secondary for the 192.124.139 network (note that the network address is backwards on the secondary line).

You do not need to configure the /etc/namedb/ SAVEFILE files. These will automatically be downloaded from the primary nameserver and saved in /etc/namedb/ when named(8) is started.

The ''named'' daemon is automatically started at system boot time if the file /etc/named.boot exists. The man page for named(8) discusses server configuration in more detail.

See the named(8) man page for more information.

Remote Cache

The other way to configure DNS is to use the /etc/resolv.conf file and use a remote nameserver to resolve hostnames for you. This is only recommended if your site has a very reliable DNS server available via a fast network link (as expected this generates network traffic for every hostname lookup, no caching is done). Typically, you would use your fileserver or network gateway machine (a machine you already rely on for basic system operation).

Setting up DNS services using a remote server only requires the creation of a simple /etc/resolv.conf file (see resolver(5)). All you need to know is the IP address of your local server and your domain. For example:

```
nameserver 99.99.99.99
domain mycompany.com
```

This will cause all hostname lookup requests to be forwarded to the remote server (99.99.99.99 in this example) for resolution.

Primary Name Server

If you are setting up a primary server for a domain, then be sure and configure the .IN-ADDR.ARPA domain for those hosts (see the 0.0.127.IN-ADDR.ARPA example in the /etc/namedb/* files). Otherwise remote sites will not be able to resolve your IP Addresses to actual hostnames and your site will be denied access to all Internet services that require this.

See the named(8) man page for more information.

the user is necessary. The port number for the 3c509 is also set by the kernel. Note that the 3c509 also uses port 0x100 during autoconfiguration – this can conflict with another device there.

° BSD/386 supports WD/SMC Ethernet at 0x280, 0x2a0, 0x2e0, 0x300, 0x320, 0x340, 0x360, 0x380, 0x3a0, and 0x3e0.

‡ BSD/386 supports the 3C503 at 0x280, 0x2a0, 0x2e0, 0x300, 0x310, 0x330, and 0x350.

* Parallel printer ports and Pocket Ethernet adapters at port 0x3bc are known as unit 2 (lp2, pe2 or xir2). If your configuration doesn't conform to these specifications, please make it match. Note that **it is extremely important that you avoid conflicts in the Port addresses, IRQ, and 'iomem' addresses between and among the peripherals**. Conflicts are one of the two biggest sources of reported service problems. Please pay extra attention to the configuration.

The following are also available but not in generic kernel as distributed; they are in the DEMO kernel configuration. These devices may also be used at other addresses with localized kernel configurations; see the manual pages in section 4 of the manual.

Device	Example Port	IRQ	DRQ
PS/2 mouse	*	12	
DigiBoard PC/Xe	0x220	*	0xd0000
Specialix tty	–	*	0xd8000
RISCom/N1	0x220	12	
MPU-401 MIDI	0x300	*	
SDL RISCom/8	0x220	*	
MAXPEED	–	–	0xd4000

In the tables above, the asterisk ('*') denotes ''Automatically detected''.

The following are also available but not in the distributed kernels:

Device	Port	IRQ	iomem
AST-4	0x2a0	*	
HP EtherTwist	0x2c0,0x300	*	0xc8000
Allied Telesis Ether	*	*	

The BSD/386 kernel supports the floating point coprocessor or automatically emulates the hardware if none exists.

Important VGA/EGA Setup Comments

Many VGA/EGA cards have an option which will cause them to generate interrupts. When running BSD/386, this option should be disabled.

Requiring DOS for Configuration

Occasionally, you'll find you'll have to use the DOS software (supplied with your peripheral card) to set up WD/SMC Ethernet cards, ATI graphic controller cards, or maybe some others. Note that there appear to be two 'modes' of configuration provided by the WD/SMC DOS utilities. The two

Hardware Configuration

BSD/386 includes an 'autoconfiguration' facility that attempts to probe your system to determine which peripherals are present. Booting from the floppy uses a 'generic' kernel which has a large selection of devices – but not all of them.

If you have purchased a Mitsumi CDROM, note that the factory configuration might **not** conform to this table. You'll probably have to reconfigure it (see below).

Hardware Setup and Configuration

The generic kernel supports the configuration in the table.

Device	Port	IRQ	DRQ	iomem†
First com port	0x3f8	*		
Second com	0x2f8	*		
Floppy	0x3f0	*	2	
1st hard-disk ctrl	0x1f0	*		
Devices above have standard PC-compatible parameters				
2nd hard-disk ctrl	0x170	*		
Parallel Port*	0x378	*		
Xircom	0x378	7		
ISA QIC Tape	0x300	*	1	
Microsoft Busmouse	0x23c	5		
Logitech Busmouse	0x23c	5		
Adaptec 154x SCSI ctlr	0x330	*	5	
Adaptec 174x SCSI ctlr	*	*	*	
WD/SMC Ethernet°	*	*	*	0xd0000
NE Ethernet	0x340	*		
NE Ethernet (alternate)	0x320,0x360	*		
3c501 Ethernet	0x320	*		
3c503 Ethernet‡	*	*	*	0xdc000
3c505 Ethernet	0x240,0x320	*		
3c507 Ethernet	0x310	*	*	0xd0000
3c509 Ethernet†	0x250	*		
3c579 Ethernet	*	*		
Intel EtherExpress 16	0x260	*		
TNIC-1500 Ethernet	0x300,0x320, 0x340,0x360	*	3	
Xircom Pocket Ethernet/ Parallel port*	0x378,0x3bc	7		
Mitsumi LU002 CD-ROM	0x334,0x340	2/9	3	
Mitsumi LU005 CD-ROM	0x334,0x340	2/9	6	
Mitsumi FX001 CD-ROM	0x334,0x340	2/9	6	

† The iomem addresses for WD/SMC, 3c503, and 3c509 Ethernets are automatically configured by BSD/386 – no special hardware configuration by

Program Debugging

This section will cover what to do in the event that you are having a problem with a utility program.

Improper Behavior

If a program is doing something unexpected (e.g., Permission Denied errors), you can often determine the cause by doing a system call trace on it. To do this simply run the program using ktrace(1), e.g,:

```
% ktrace cat /dev/null
```

The ktrace(1) program will output the raw system call trace information in the file "ktrace.out". Then invoke the kdump(1) program to display this information in a human readable form.

Note that ktrace works even if you have not compiled the program specifically for debugging (i.e., cc -g).

Core Dumps

Core files are named core.program (where program is the name of the program, e.g., core.cat for the cat program). Core refers to the old-time use of core memory. Nowadays, the term may be dated, but the usefulness of a core file is not. It is a copy of all the memory allocated to a program at the time the program did something wrong to generate the core file. These are commonly caused by either a Segmentation Fault or a Bus Error, these terms are as dated as the core file but generally refer to the source of the fault. In general, they indicate a reference to memory outside the program's address space. The distinction isn't important for out discussion.

The other main way to generate a core dump is to request one. Unless you are running an interactive utility (like vi or emacs) you can usually type ^\ (control-backslash) to force a core dump to be generated. Try it. Run "cat", which will just sit around waiting for input. Then type ^\ (control-backslash) and you will see something like:

```
^\Quit (core dumped)
```

Debug core files using the gdb(1) program. For the example above, type

```
% gdb /usr/bin/cat core.cat
```

and to enter the gdb debugger. To debug a program with any ease, the program must have complete debugging symbols (i.e., be compiled with the -g option of cc(1)).

If you have a core dump from a system utility you will need to rebuild it from source using the -g flag to create the full symbol table. If you do not have a source license, you will need to make the core file available to BSDI's customer support folks for debugging.

Quota Configuration

First make sure your kernel was built with "options QUOTA" enabled (if not then rebuild it). This is the default in the distributed kernel.

Then add the options "userquota" and/or "groupquota" to the filesystems in /etc/fstab that you wish to run quotas on, e.g.,:

```
/dev/wd0d /home ufs rw,userquota 1 3
```

Run "quotacheck -a" to initialize the quota files on each filesystem and then run "quotaon -a" to enable the quota system itself. Both of these operations are done at system boot.

Once everything is initialized you can use edquota(8) to edit individual quotas. That will put in your editor with simple form to fill out. For example, "edquota sanders" gives:

```
Quotas for user sanders:
/u: blocks in use: 75616, limits (soft = 0, hard = 0)
        inodes in use: 3029, limits (soft = 0, hard = 0)
```

You don't have to edit each user either, edquota can be used to copy one users quota settings to another user (see the man page for details).

Users can use the quota(1) command to display their quota information. Use repquota(8) to get a summary of all users usage.

Driver	Graphics Chip	640x 480	800x 600	1024x 768i	1024x 768ni	1280x 1024i	1280x 1024
vga	STDVGA	4	-	-	-	-	-
	ET3000	4/8	4/8	-	-	-	-
	ET4000	4/8/16	4/8/16	4/8	4/8	4	-
	GD5402	4/8	4/8	4	4	-	-
	GD5420	4/8	4/8	4/8	4/8	-	-
	GD5422	4/8/16	4/8/16	4/8	4/8	4	-
	GD5424	4/8/16	4/8/16	4/8	4/8	4	-
	TVGA 8900C	4/8	4/8	4/8	4/8	-	-
	PVGA1	4/8	4/8	4	-	-	-
	WD90C00	4/8	4/8	4	4	-	-
	WD90C11	4/8	4/8	4	4	-	-
	WD90C30	4/8/16	4/8/16	4/8	4/8	4	-
agx	AGX14	8	8	8	8	-	-
	AGX15	8/16	8/16	8/16	8/16	8	8
cli	GD5426	8/16	8/16	8	8	8	-
	GD5428	8/16	8/16	8	8	8	-
w32	ET4000/ W32	8/16	4/8/16	4/8/16	4/8/16	4/8/16	
wdi	WD90C31	8	8	8	8	4	-
wga	VESA8514	4/8	4/8	4/8	-	-	-
	82C480	4/8	4/8	4/8	4/8	4/8	4/8
	82C481	4/8	4/8	4/8	4/8	4/8	4/8
	MACH8	4/8	4/8	4/8	4/8	4	-
	MACH32	4/8/16	4/8/16	4/8/16	4/8/16	4/8	4/8
	86C801	4/8/16	4/8/16	4/8	4/8	4/8	4/8
	86C805	4/8/16	4/8/16	4/8	4/8	4/8	4/8
	86C911	4/8	4/8	4/8	4/8	4	-
	86C924	4/8	4/8	4/8	4/8	4	-
	86C928	4/8/ 16/32	4/8/ 16/32	4/8/ 16/32	4/8/ 16/32	4/8/16	4/8/16
	WD9500	4/8	4/8	4/8	4/8	-	-
xga	IBM	XGA-NI	8	8	8	8	-

All drivers are accelerated with the exception of the VGA drivers in 4/8/16 bit modes.

1280x1024 in "depth 8" and 1024x768 in "depth 15". If you use depth 15, you have a DirectColor and a TrueColor visual. The first one is the default visual. Some applications won't work with that visual, hence it might be necessary to specify explicitly that you want TrueColor as default visual. No special strings for chipset or vendor are defined yet.

The vga driver is the general workhorse. It offers support for unaccelerated VGA boards. Depending on your equipment, you can use a wide range of depths (4, 8, 15 & 16) and resolutions. Depth 4 selects the standard 16 color mode which drives the VGA in planar mode. Up to 1280x1024 is possible with a virtual resolution of up to 2048x1024 on a board equipped with 1MB. The depths 8, 15 & 16 are the SVGA packed modes which are generally slower than the 16 color mode. The HighColor(tm) modes (depth 15 & 16) are only available on boards with ET4000, WD90C30, WD90C31, GD5422, GD5424 & GD5426 chips.

Currently following strings for chipset are defined (and hence ONLY FOLLOWING CHIPSETS ARE SUPPORTED):

ET3000	GD5422	WD90C00	STDVGA
ET4000	GD5424	WD90C11	TVGA8900C
GD5402	GD5426	WD90C30	
GD5420	PVGA1	WD90C31	

The wdi driver is used to do accelerated drawing only with "depth 8" on a WD90C31. Other depth can be used by specifying a vga driver instead. Resolutions up to 1024x768 are supported. As usual the remaining vidoe memory is use for server internal purposes. No special strings for chipset or vendor are defined yet.

The wga driver is the driver for 8514/A clones and S3 based accelerators. Depnding upon the graphics chip depths of 4, 8, 15, 16 & 24 are possible. Currently following strings for chipset are defined (and hence ONLY THE FOLLOWING CHIPSETS ARE SUPPORTED):

82C480	86C805	86C924	MACH32
82C481	86C911	86C928	VESA8514
86C801	WD9500	MACH8	

On the 86C801/805/911/924/928 & MACH8 there is a fast depth 4 mode available. The 86C801/805/928 & MACH32 support depth 15 and 16. On the 86C928 a depth 24 is supported. The following vendor strings are supported:

WINNER	ELSA WINNER 1280, WINNER 1000, WINNER 2000
WhirlWIN	PixelWorks WhirlWIN WW1280 & WL1280
Nth Engine/150	Nth Engine/150 for setting up 1280x1024
Nth Engine/250	Nth Engine/250 for all modes

The list of graphics chips supported in X386 1.4 is:

Rebuilding Your Kernel

Loading the Source

Use *installsw* to load the source if you do not intend to build from CDROM. Once you have the kernel (sys) hierarchy extracted, you can reconfigure and build new kernels.

Configuring the Kernel

The first step is to set up a configuration file. There are several example configuration files in /usr/src/sys/i386/conf. GENERIC is the distributed kernel and can boot from a floppy, cdrom or magnetic disk. The DEMO configuration includes more drivers and features. Usually, you will start with GENERIC or DEMO when making a custom configuration. The other configuration files are not normally used, and may be somewhat out of date; however, they will give you an idea of the kinds of changes you can make.

Once you have a configuration file tailored to your needs, run the config program to build the appropriate include and Makefiles. If you wanted to build a kernel for your configuration file called LOCAL, run the following command:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC LOCAL
# edit the local configuration file
# config LOCAL
```

The config program creates the directory /usr/src/sys/compile/LOCAL and populates it with all the necessary header files and a Makefile.

Building Kernels

→ IMPORTANT NOTE ←

A kernel configuration that uses "config bsd swap generic" must have GENERIC defined, usually by using "ident GENERIC", and must have all of the disk types configured. If you wish to make another generic kernel configuration (which configures root and swap at boot time), you can do so by copying the GENERIC configuration to another name and leaving the ident as GENERIC.

Alternatively, you can build a non-generic kernel using something like "config bsd root on wd0a". In this case, not all disk types need to be present, and GENERIC must not be defined (i.e., choose a different 'ident').

Below are the comments from the GENERIC configuration:

```
# Note: "swap generic" should be used
# only in a GENERIC kernel
# with all of the disk drivers included.
# Other systems should use something like:
# config bsd  root on wd0a swap on wd0b and wd2b
```

```
# config bsd  root on sd0a swap on sd0b and sd1b
config    bsd     swap generic
```

The kernel will not compile if the rules are not followed, and "swap generic" is included without defining GENERIC. At least that avoids creating a kernel that doesn't boot. Likewise, the kernel will not compile if the ident is GENERIC and the config line explicitly names the root and swap devices.

To build your new kernel, cd to the sys/compile/*kernel_name* directory and run make:

```
# cd /usr/src/sys/compile/LOCAL
# make depend
# make
```

The last make command should create an executable kernel called bsd in the current directory. It takes less than 10 minutes on 33 MHz 486 machines.

To install this kernel, move the current kernel out of the way, copy the new one to /bsd, and reboot:

```
# cp bsd /bsd.new
# reboot
```

Boot your new kernel explicitly. When it works, save the old one in bsdi.old and mv the new one into place. We recommend that you leave the original kernel in /bsd.generic so you always have a usable kernel to fall back on. See boot(8) for additional details on booting. To boot a kernel other than the default, you have to abort the load of the default (/bsd) by typing return while it is loading. The boot program should then respond with a Boot: prompt. At the Boot: prompt, you can load whatever kernel you wish:

```
Boot: wd()bsd.generic        sd() for SCSI
```

This will load the bsd.generic kernel from the filesystem on /dev/wd0a (or /dev/sd0a).

Building the Kernel from CD-ROM

Note that if you wish to modify kernel files (beyond the configuration files), you should just install the kernel source package and use the hard drive.

To set up the system to compile the kernel from the CD-ROM, execute the following steps:

1 Mount the CD-ROM on /cdrom

```
# mount -t iso9660 /dev/mcd0a /cdrom
```

or

```
# mount -t iso9660 /dev/sdNc /cdrom
```

(where X is the correct scsi disk identifier)

2 Create a build directory and cd there

```
# mkdir /usr/cdrom_sys
# cd /usr/cdrom_sys
```

3 Run the mksyscompile command

```
# /cdrom/cdrom_tools/mksyscompile \
                  /cdrom/usr/src/sys
```


CDROM	Mitsumi LU002S, LU005S, FX001

Ethernet	WD/SMC 8003/8013 EtherCard PLUS/Elite Series
	(8013 now supports 16-bit mode and 16K of memory)
	SMC Ultra
	Novell NE1000 & NE2000
	3COM 3C501 (EtherLink)
	3COM 3C503 (EtherLink II)
	3COM 3C505 (EtherLink Plus)
	3COM 3C507 (EtherLink 16)
	3COM 3C509 (EtherLink III)
	3COM 3C579 EISA 32-bit (EtherLink III)
	TNIC 1500 Transition Eng Fast ISA busmaster DMA NIC
	Allied Telesis RE2000/AT-1700 Series
	Intel EtherExpress 16
	HP EtherTwist PC LAN Adapter/16 Plus
	  HP 27247B PC LAN Adapter/16 TP Plus [AUI/UTP]
	  HP 27252A PC LAN Adapter/16 TL Plus [AUI/BNC]
	Xircom PocketEthernet II (plugs into parallel port)
	Xircom PocketEthernet III (plugs into parallel port)

HDLC	RISCom/N1 single port HDLC @ 56K (PPP or cisco HDLC) RISCom/H2 dual port HDLC @ 56K (PPP or cisco HDLC)
Miscellaneous	MPU-401/compatible MIDI controller SoundBlaster PRO

X Server Information

The X server drivers are classified as follows:

agx	IIT AGX014 based accelerators
cli	Cirrus Logic GD5426 based accelerators
dgx	Dell DGX/2 ("jaws")
vga	unaccelerated VGA
wdi	Western Digital WD90C31 based accelerators
wga	8514/A clones or S3 based accelerators

The agx driver only supports a "depth 8" on a AGX014. Resolutions up to 1024x768 are supported (real or virtual). The rest of the remaining video memory is used for server internal purposes. There are no special strings for chipset or vendor yet.

The cli driver is used for accelerated drawing only with "depth 8" on a GD5426. Other depths can be used by specifying a vga driver, instead. Resolutions up to 1024x768 are supported (even on a 2MB configuration). As usual, the remaining video memory is used for server internal purposes. No special strings for chipset or vendor are defined yet.

The dgx driver only applies to the Dell PowerLine computers equipped with a DGX/2 (jaws) board. Only a few resolutions are supported, due to the limited capabilities of the IMS332 chip used there: 1024x768, 1152x900 and

Supported Hardware

This list shows V1.1's supported hardware.

CPU	80386, 80486, and Pentium processors
FPU	387 or equivalent (486DX and Pentium has on-chip FPU)
	Automatic software emulation when FPU unavailable
RAM	4MB to 256MB; 8MB minimum for X
Display	Mono, CGA, EGA, VGA, SVGA
	(VGA, SVGA for X)
Keyboard	Standard PC/AT keyboard
Floppy	Standard IBM PC/AT Floppy Controller
	Dual-sided 3.5" floppy drive (720K, 1.44M)
	Dual-sided 5.25" floppy drive (360K, 1.2MB)
Parallel	Standard PC/AT parallel ports
Serial	Standard PC/AT serial com ports
Multiport	RISCom/8 8-port Async Mux
	MAXPEED Multiport Async SS-4/2, SS-4PLUS, SS-8/2, SS-16/2
	Digiboard Multiport Async PC/4e, PC/8e, PC/16e, PC/8i(w/64K)
	Specialix SLXOS 8-32 async mux
	AST-4 4-port async card
	MU-440 4-port async card
	Generic 'COM' multiport async cards
Mice	Microsoft Serial & Busmouse
	MouseSystems Serial Mouse
	MMSeries Serial Mouse
	Logitech Serial & Busmouse
	MouseMan Serial Mouse
	ATI Bus Mouse (both Ultra and Ultra Pro types)
	PS/2 Mouse
SCSI	Adaptec SCSI host adapters
	Models: 1540B, 1542B, 1540C, 1542C, 1540CF, 1542CF, 1740A, 1742A (174XA and 154XC* support Fast SCSI-II)
	BusLogic (BusTek) BT-542B SCSI adapter
	BusLogic (BusTek) BT-445S VLB SCSI adapter
Disk	Any SCSI hard disk
Tape	Any SCSI tape drive
CDROM	Any SCSI CDROM drive
Mag/opt	Any SCSI Magneto Optical drive
Disk	IDE/ESDI/ST-506/RLL/MFM controllers
	Any compatible hard disk
Tape	WangTek 5150PK QIC-02 tape
	Archive Viper QIC-02 tape
	Everex EV-811, EV-831, EV-833 Tape

4 `cd` to the i386/conf directory and edit your config file

```
# cd i386/conf
# cp GENERIC LOCAL
# vi LOCAL
```
make sure to fix the **config** *and* **ident** `lines`

5 Run `config` to create the build directory and cd to it

```
# config LOCAL
# cd ../../compile/LOCAL
```

6 Run the `fixmake` command to fix up the Makefile to compile from the CD-ROM

```
# /cdrom/cdrom_tools/fixmake /cdrom/usr/src/sys
```

7 Run 'make depend' and 'make'

```
# make depend
# make
```

You are now ready to move your old kernel to a backup name, install the new version, and reboot the system.

Compiling Directly from CD-ROM

The CDROM distribution was built with the ability to compile from the CDROM media. With the exception of the X and ISODE sources and some others, you can re-build from the sources without having to copy them to your hard drive.

The special Makefiles used for the libraries and utilities know how to make use of a sub-directory named 'obj' within each directory to hold the compiled sources. On the CD-ROM version, the source directories already contain 'obj' links which point into a hierarchy rooted at /usr/cdrom_obj. There is a special target in the Makefiles which automatically makes the directory pointed to by the 'obj' link, so you can compile everything directly from the CD-ROM.

The following example shows how you would compile the *cat* program from CD-ROM:

1 Make sure /usr/cdrom_obj exists and is on a filesystem with enough space.
Usually, the /usr filesystem will have plenty of free space to compile some programs and/or libraries. If you plan on compiling lots of stuff, you may want to make /usr/cdrom_obj a filesystem of its own, or you can make it a symbolic link to a filesystem with more space.

2 Mount the CD-ROM on /cdrom

```
# mount -t iso9660 /dev/mcd0a /cdrom
```
or
```
# mount -t iso9660 /dev/sdNc /cdrom
```
(where N is the correct scsi disk identifier)

3 **`cd`** to the source directory you wish to compile, e.g., for cat:

```
# cd /cdrom/usr/src/bin/cat
```

4 Make the directory the obj link references by issuing the 'make objdir' command:

```
# make objdir
```
5 Generate the dependencies, and compile
```
# make depend
# make
```
6 If desired, install the new version with 'make install'
```
# make install
```
If you want to modify a source file, you can copy it to the obj directory. The Makefiles will find the modified version in the obj directory first, so you don't have to copy the entire source directory to your hard drive. For example, if I wanted to make a modification to 'cat.c':
```
# cp cat.c obj
# vi obj/cat.c

# make depend
# make            to rebuild the new version
# make install    to install the new version
```
You should always do a 'make depend' before you do a the 'make' to build the programs. Some programs won't build unless a 'make depend' is done first.

Upload/Download Using ZMODEM

BSD/386 V1.1 now includes the public domain zmodem implementation. This provides the commands '/usr/contrib/bin/rz' for receiving files via zmodem, and '/usr/contrib/bin/sz' for sending files. See the man pages for details about these commands.

There are also two wrapper scripts ('zup' and 'zdown') included to support calling these programs while running tip(1). These let you easily upload and download files in the middle of a session. If you are connected to a remote system with tip, then the command:
```
~Czup filename
```
will upload the given file to the remote system using 'sz'. In the other direction:
```
~Czdown
```
will start 'rz' so that the other system can send a file. The remote system will specify the file name, and it will be removed if it presently exists.

You can pass along options, for example:
```
~Czdown -a
```
declares that the file is ascii, so \r\n sequences should be converted to \n, and ^Z should be stripped.

Name Resolver (DNS)

As of 1.1, the DNS resolver will no longer try partial subdomains of your default domain by default. For example, if your default domain is "home.vix.com" and you have a "localhost.vix.com" name defined but no "localhost.home.vix.com" name defined, you will not be able to use the unqualified name "localhost" or other names in that domain. The fix is either to define a "localhost" name in the domain you use as your default, or to use resolv.conf's "search" directive to explicitly generate the previously implicit behavior. The change was made for security reasons and to conform to RFC 1535.

Futures

- 4.4 BSD-Lite, including new filesystem features and performance
- SCO Binary Compatibility
- Shared Libraries
- SPARC port
- Even more and faster video cards (e.g., Matrox gives 250,000 Xstones)
- More device drivers

Configuration

- By default, 16-bit WD/SMC 8013 Ethernet cards now run in 16-bit mode with 16 KB or more of memory; in 1.0 they used 8-bit mode and 8 KB. If there are 8-bit memory devices in the same 128 KB area of I/O space, this can cause those devices to fail. This includes video BIOS ROMs used by rundos. Problems can normally be avoided by setting the link2 flag with "ifconfig we0 link2", leaving the card in 8-bit mode except when copying packets to or from the card. The card can be forced to use 8-bit mode at all times by reconfiguring with iosize set to 8192. See we(4) for details.
- Flags for selecting the medium connector type have changed for the 3Com 3C509. The link0 flag selects a BNC connector, link1 selects twisted pair, and the default remains as AUI.
- /etc/gettytab, /etc/printcap, /etc/remote have been updated to support the :ms: attribute (stty-style mode strings). The old way of specifying tty modes (fs/fc etc) is obsolete. This new method is much more flexible than the previous and the change was required to support the POSIX termios interfaces. This also lifts any baud rate restrictions so you can use fast baud rates on devices that support them. There are comments and examples in each file to help make conversion easier.
- Changes to the tty settings effected by getty now remain in effect – getty used to restore system defaults before exec'ing login. This means you can modify the settings on a per tty basis by putting the changes in its /etc/gettytab entry (made even easier with the :ms: capability). As a side effect you need to be sure to set the :np: flag in the console's /etc/gettytab entry; the sample /etc/gettytab and /etc/ttys shipped with 1.1 do this.
- The default /etc/rc and /etc/rc.local scripts have changed; be sure to merge any changes you have made into the new versions.
- Dump and restore now use the environment variable $TAPE for their default output device.

KERMIT

BSD/386 V1.1 does not contain kermit, due to a change in policy by the authors of kermit. If you are a user of kermit, here are some alternatives:

- Use the new zmodem support with tip (see below).
- Use the kermit distribution from BSD/386 1.0.
- Get the latest kermit sources from an FTP server and compile it.
  Recent versions compile on BSD/386 without change.

Known Problems and Surprises

Some of the main utilities, as distributed, exhibit more problems than we'd like to see. These include: `sh` (often `bash` will work correctly when `sh` fails), `ksh` (again, try `bash`), and `vi`.

Here's a list of surprises, known problems, and (sometimes) workarounds.

- The device mmap code now obeys the offset parameter, which previously was ignored. This may affect people writing either kernel- or user-level code to mmap devices.
- The Xircom PE-III does not always restart properly on laptop machines after entering and exiting "standby mode". Try "ifconfig xir0 down" followed by "ifconfig xir0 up"; if that doesn't work, reboot.
- The argument to uucp(1)'s "-n" option is currently limited to 8 characters. To reach longer addresses, use an alias in /etc/aliases.
- Pax/tar will not read tapes with blocksizes greater than 32256 (63 * 512) bytes. Such tapes must be re-blocked with dd and piped to pax/tar (e.g., 'dd bs=63k < /dev/rst0 | pax -r'). Also, pax/tar will not read data correctly from some fixed-record SCSI tapes. QIC tapes work fine, as do Exabyte tapes using the default (variable-record) selection. 4mm DAT with fixed 1024-byte records breaks, as does fixed 1024-byte record Exabyte. As it happens, pax/tar can read fixed-record tapes in variable-record mode, so that's a simple workaround for Exabytes; unfortunately we can't select fixed vs. variable mode for 4mm DAT tapes, so they must be reblocked (e.g. 'dd bs=1k < /dev/rst0 | pax -r').
- The system can hang if it runs out of swap space. The amount of swap space required depends on the amount of page-out activity, ranging from zero on systems with sufficient memory for all needs, up to enough space to contain the data plus stack segments used by all running programs on systems that page heavily.
- Long running programs that fork periodically under conditions that require paging can consume excessive swap space. The swap space is reclaimed when the parent process exits.
- Files' contents which are accessed both via the mmap() call and by read()/write() calls are not necessarily consistent after modifications. Programs using mmap() can force synchronization with the buffer cache by using the msync() call.
- As of 1.1, the DNS resolver will no longer try every partial subdomain of your default domain. this means, for example, that if your default domain is "home.vix.com" and you have a "localhost.vix.com" name defined but no "localhost.home.vix.com" name defined, you will not be able to use the unqualified name "localhost". the fix is either to define a "localhost" name in the domain you use as your default, or to use resolv.conf's "search" directive to explicitly generate the previously implicit behaviour. this is not a bug; the change was made for security reasons and to conform to RFC 1535. to restore the previous behaviour, you can use an explicit

"search" directive in /etc/resolv.conf; if your system's name is "gw.home.vix.com" you would use "search home.vix.com vix.com" in resolv.conf. for security reasons you should only list domains in your "search" list if you trust their domain administrators.

- On some laptops, our Xircom PE-3 driver is unable to correctly determine and/or use the default parallel port configuration. If you try the "xir" driver on a PE-3 and get a boot-time message to the effect that the selected hardware configuration is unavailable, you should use your SETUP floppy or your ROMBIOS "setup" mode to change the printer port configuration. On some machines you must change it to "bidirectional" and on others you must change it to "output-only". Please let <support@bsdi.com> know if you have trouble with this and what you finally did that solved it; this way we can share your experiences with other customers who have the same kind of laptop. Some Toshiba models (in particular the T4500) are known to require that the printer port be set to OUTPUT (rather than Bidirectional) with the TSETUP DOS utility in order to operate properly with the xir driver even though the PE3TEST program works in Bidirectional mode.
- If you invoke the X server and the Xconfig file has errors or if your mouse is connected improperly, then your display may function erroneously: won't start X, won't leave X, won't restore text mode, e.g. Sometimes the 'vreset' program will restore text mode correctly. Detected errors are logged in $HOME/.xinit.log if startx was used.
- The POSIX 1003.2 shell specification states that the Bourne shell (sh), like the historical ksh shell, now sources the file named by the ENV environment variable at startup. Many ksh users have an ENV variable already set – the new 'sh' may behave differently than before until the ENV environment variable is unset.
- Sendmail does not expand the '&' in gecos field for the user name. Be sure to fix your /etc/master.passwd file.

Notes On Writing SCSI Tape

Streaming cartridge tape has some restrictions on where writes may be initiated. A write may only start at the beginning of the tape (BOT) or after the last data block or file mark (end of data, EOD). Old data following the last written block or filemark is inaccessible; it is impossible to edit a tape, although you may append to it. The SCSI tape driver will automatically space to EOD if the tape is opened for writing and the tape head is not at BOT.

Earlier versions of BSD/386 used fixed-length 1024-byte records when reading or writing Exabyte 8mm cartridge tapes. The current version defaults to variable-length records, to be compatible with Sun Exabyte tapes. You can read fixed-style tapes in variable length mode by providing a blocksize of 1024, but this can be very slow; /dev/rf8mm*N* provides the old fixed-length record interface and is usually faster with such tapes.

- Compilers
  - GCC 1.42
  - GCC 2.5.8
  - libg++ 2.5.3
- Latest networking utilities: BIND 4.9.2, telnet, sendmail 8.6.5
- Sendmail test suite (checksendmail(8))
- Updated utilities:
  - groff 1.08, ghostscript 2.6.1, diff 2.6, grep 2.0, awk 2.15
  - byacc 1.9, flex 2.4.6, bash, elm, gzip, file, bc, screen, ...
  - Updated versions of pax, sort, ispell, ed, doscmd, cron, uucp, ...
  - New contrib code: dc, ac, lprps, irc, ups, gnuchess, seyon, ...
  - Completely new versions of pr, pstat
  - Updated db, regex, resolver, kvm routines; new rune, err routines
  - Significant improvements: getty, test, tip, startslip, disksetup
  - Many other fixes
- Improved shells: sh, csh, ksh
- cksum was fixed to conform to POSIX 1003.2
- FlexFax fax support
- X11 clients, including: NCSA Mosaic 2.1, XV 3.00
- Tcl 7.2 and Tk 3.5
- Zmodem support
- Latest MH (6.8.3)
- Secure data checksum programs:
  - shs – NIST proposed Secure Hash Standard
  - md5 – RSA Data Security, Inc. MD5 Message-Digest Algorithm
- BASIC Programming Language
- New 'man' – supports arbitrary formatting and (de)compression commands
- wuftpd FTP daemon (access control, logging, auto-tar, auto-compress)
- xntp3 NTP daemon and utilities
- bdes works

New Features

- Resolved hundreds of bugs; some still remain (but not too many!)
- Improved DOS support
- Mountable MSDOS Filesystem (enables use of standard utilities: grep, find, rm, etc.)
- Improved configuration and installation documentation
- Better locale support in libc, and Bell Labs Research style rune routines
- libnm(3) is a new freely redistributable ANSI/POSIX-compliant libm
- Improved printer filter support (HP and Postscript)
- Fast assembly versions of most C library string routines

Kernel

- Dramatic improvement in serial I/O performance
- High speed networking (95% of max speed on some cards) (better client-server!)
- Improved interrupt handling
- Implementation of missing/incorrect POSIX.1 features (VMIN/VTIME, getgroups)
- 32-bit user and group IDs, device numbers and process IDs (local filesystem allows only 16-bit user and group IDs)
- IP Multicast support including routing (great for multi-media) (supported on WD/SMC, Novell, 3C503, 3C507, 3C509, 3C579, TNIC-1500, PocketEthernet III, and point-to-point links)
- Hooks included for DEC's screend gateway-filtering facility (to obtain screend contact support@bsdi.com)
- Larger virtual memory limits
- Console display supports more ANSI features (mostly compatible with SCO console now)
- Better CPU identification (Including Pentium recognition)
- Large memory (up to 256MB)
- Support for larger buffer caches and other kernel virtual memory needs
- Easier configuration of other kernel resources for large systems
- Large kernels – removed 640KB limit on kernel size
- Dynamic allocation of memory for pseudo-devices reduces memory usage
- Compaction of kernel virtual memory when usage changes
- Support for serial console (although bootstrap uses display for now)
- Fix to argv[0] for interpreted scripts
- Fixes to revocation of terminals at logout
- ISO 9660 CD-ROM support for disks with small logical blocks
- Hardware floating point now works with DOS emulation
- Optional Async NFS
- Mountable MSDOS Filesystem

Software

- /bin/tar is now a link to pax; it supports the standard tar command line interface. See tar(1) for more information.
- /usr/bin/vi is now nvi (written by Keith Bostic). Elvis is still available as /usr/contrib/bin/elvis for the faithful.
- 'sh -r' creates a restricted shell, similar to System V. Sh now supports a csh-style 'limit' command (not 'ulimit'). Sh now handles POSIX.2 line editing (a bug or a feature, depending on your opinion).
- /bin/sh now uses $ENV. Users with an ENV environment variable may have to make changes to their ENV file if running shell scripts causes complaints.
- XFree86 2.0 distribution included
- New video card support (generic VGA, Diamond Stealth, monochrome)

Some quarter-inch cartridge tape drives can handle more than one tape format. Most such drives will read more than one density and write at their highest density by default. If you need to force a particular density, you can use one of the special tape devices named /dev/r*qXX*st*N*, where *qXX* is the density you want (e.g., q11 = QIC-11, q120 = QIC-120, etc.).

Acknowledgments

BSD/386 includes software distributed by the Computer Systems Research Group at the University of California, Berkeley. BSDI gratefully acknowledges the following organizations who have funded the CSRG since the 4.3BSD release, have provided people, or have provided hardware to support their research. BSDI apologizes for any name inadvertently omitted.

Compaq Computer Corp.
Cray Research Inc.
Dept. of Defense Advance Research Projects Agency (DARPA)
Digital Equipment Corp.
NASA Ames Research Center
The Hewlett-Packard Company
The Open Software Foundation
UUNET Technologies Inc.

The list below recognizes people and organizations that have provided a large subsystem to Berkeley. (Their contribution may not be part of the current BSD/386 release.)

Sparc I/II support	Computer Systems Engineering Group, Lawrence Berkeley Laboratory
386/486 support	Bill Jolitz and TeleMuse
ANSI C library	Chris Torek
ANSI C prototypes	Donn Seeley and John Kohl
Autoconfiguration	Robert Elz
C library documentation	American National Standards Committee X3
CCI 6/32 support	Computer Consoles Inc.
Disklabels	Symmetric Computer Systems
Documentation	Cynthia Livingston and The USENIX Assoc.
gcc, gdb	The Free Software Foundation
groff	James Clark (The FSF)
HP300 support	Jeff Forys, Mike Hibler, Jay Lepreau, Donn Seeley and the Systems Programming Group; University of Utah CS Dept.
ISODE	Marshall Rose
Job control	Jim Kulp
Kerberos	Project Athena and MIT
Math library (libm)	K.C. Ng, Z-S. Alex Liu, S. McDonald, P. Tang and W. Kahan
NFS	Rick Macklem
NFS automounter	Jan Pendry
Network device drivers	Micom-Interlan and Excelan
Omron Luna support	Akito Fujita
Quotas	Robert Elz
RPC support	Sun Microsystems Inc.

Update Notes from V1.0 to V1.1

Software

- Programs that use libkvm must be recompiled.
- Libraries and object files that call db routines need to be recompiled. Programs that use db files should be recompiled to take advantage of bug fixes; the new db routines will read the old format.
- The changes to user and group ID types have caused the stat structure to change. **Programs compiled for 1.0 will work with 1.1 in compatibility mode, but programs compiled for 1.1 will not work on 1.0 systems.**
- Libraries and object files that use ctype functions need to be recompiled.
- The format of the accounting files (/var/account/*) has changed and is incompatible with V1.0 files.
- The interface to process strings has changed – changes to argv are now reflected in ps listings. A new setproctitle() routine in -lutil encapsulates the new interface.
- The fgetline() routine has moved to -lcompat, to be replaced by the fgetln() routine, with correct (and not upward compatible) semantics. The -lcompat library now supports the venerable _doprnt(), _doscan(), ecvt(), fcvt(), gcvt(), insque() and remque() routines.

What's New in BSD/386 V1.1

Hardware

- Several new video display adaptors, including VGA
- Mitsumi FX001 support
- Digiboard PC/4e, PC/8e, PC/16e, PC/8i(w/64K)
- Specialix Multiport SLXOS 8-32 port serial card
- Much improved performance with RISCom/8 serial multiport card
- PS/2 Mouse kernel support with XFree86 2.0 (but not the other X server)
- BusLogic (BusTek) BT-542B SCSI adapter
- BusLogic (BusTek) BT-445S VESA LocalBus SCSI adapter (in 1542 mode)
- WD/SMC Driver (8013) Support for Ultra and 16-bit mode on all 8013 cards
- 3Com 3C579 EISA 32-bit (EtherLink III), and improved performance and reliability for 3Com 3C509 Ethernet cards
- TNIC 1500 Transition Eng Fast ISA bus master DMA Ethernet
- Allied Telesis RE2000/AT-1700 Series Ethernet
- Xircom PocketEthernet III (plugs into parallel port)
- HP EtherTwist PC LAN Adapter/16 Plus Ethernet
- SoundBlaster PRO driver plus sound utilities
- Keyboard mapping files for Belgium, Bulgaria, Norway, and Spain, and udpates for many others
- Better laptop support (disk timeouts adjusted, numlock initial setting)

Address Updates

To report change of paper mail address, please call the service desk at numbers above or send e-mail to info@bsdi.com.

Documentation Conventions

The examples of computer interaction use fixed-width type (`like this`) for computer output and bold fixed-width type for input that you, the user, type (***`like this`***). Italic fixed-width type (*`like this`*) indicates input that you type but that requires some special substitution (e.g., *`username`* requires you to change the *`username`* to the correct user's name).

Command sequences and computer output appear as left justified, fixed width text. A command line with options in square brackets like this:

```
# ls [-l]
```

means that the specifications in square brackets (e.g., -l in our example) are optional.

General Precautions

PLEASE READ THE DIRECTIONS. If you call for help, we will always go back to the first step and go through all the directions. Skipping any step can render the installation invalid.

BACKUP YOUR DATA – there is no reason to risk losing it.

Shared library support	Rob Gingell and Sun Microsystems Inc.
Sony News 3400 support	Kazumasa Utashiro
Stdio	Chris Torek
System documentation	The Institute of Electrical and Electronics Engineers, Inc.
Timezone support	Arthur David Olson
Transport/Network OSI layers	IBM Corp. and the University of Wisconsin
User level XNS	Cornell University
X25	University of British Columbia
VAX 3000 support	Mt. Xinu and Tom Ferrin
VAX BI support	Chris Torek
VAX device support	Digital Equipment Corp.
Virtual memory implementation	Carnegie-Mellon University Mach project

This list recognizes people and organizations that have provided a specific item, program, library routine, or program maintenance for the BSD system. (Their contribution may not be part of the current release.)

386 device drivers	Carnegie-Mellon University Mach project
386 device drivers	Don Ahn, Sean Fagan and Tim Tucker
HCX device drivers	Harris Corp.
Kernel enhancements	Robert Elz, Peter Ivanov, Ian Johnstone, Piers Lauder, John Lions, Tim Long, Chris Maltby, Greg Rose and John Wainwright
X25/LLC2	Dirk Husemann
adventure(6)	Don Woods, Jim Gillogly Will Crowther
ar(1)	Hugh A. Smith
arithmetic(6)	Eamonn McManus
arp(8)	Sun Microsystems Inc.
atc(6)	Ed James
backgammon(6)	Alan Char
battlestar(6)	David Riggle
bcd(6)	Steve Hayman
bdes(1)	Matt Bishop
bitstring(3)	Paul Vixie
bpf(4)	Steven McCanne
btree(3)	Mike Olson
byte-range locking	Scooter Morris
caesar(6)	Stan King and John Eldridge
cal(1)	Kim Letkeman
cat(1)	Kevin Fall
chess(6)	Stuart Cracraft (The FSF)
ching(6)	Guy Harris
cksum(1)	James W. Williams

clri(8)	Rich $alz
col(1)	Michael Rendell
comm(1)	Case Larsen
compress(1)	James A. Woods, Joseph Orost, Spencer Thomas
cp(1)	David Hitz
cron(1)	Paul Vixie
csh(1)	Christos Zoulas and Len Shar
curses(3)	Ken Arnold
cut(1)	Adam S. Moskowitz, Marciano Pitargue
dd(1)	Keith Muller and Lance Visser
des(1)	Jim Gillogly, Phil Karn, Richard Outerbridge
disklabel(8)	Symmetric Computer Systems
du(1)	Chris Newcomb
ed(1)	Rodney Ruddock
emacs(1)	Richard Stallman
erf(3)	Peter McIlroy, K.C. Ng
factor(6)	Landon Curt Noll
find(1)	Cimarron Taylor
finger(1)	Tony Nardo
fish(6)	Muffy Barkocy
fnmatch(3)	Guido van Rossum
fold(1)	Kevin Ruddy
fortune(6)	Ken Arnold
fpr(1)	Robert Corbett
fsdb(8)	Computer Consoles Inc.
fsplit(1)	Jerry Berkman and Asa Romberger
gcc/groff integration	UUNET Technologies, Inc.
glob(3)	Guido van Rossum
gprof(1)	Peter Kessler
hack(6)	Andries Brouwer (and a cast of thousands)
hangman(6)	Ken Arnold
hash(3)	Margo Seltzer
heapsort(3)	Elmer Yglesias, Kevin Lew, Ronnie Kon
hunt(6)	Greg Couch and Conrad Huang
indent(1)	David Willcox, Eric Schmidt, James Gosling, Sun Microsystems
j0(3)	Sun Microsystems, Inc.
j1(3)	Sun Microsystems, Inc.
jn(3)	Sun Microsystems, Inc.
join(1)	David Goodenough, Michiro Hikida, Steve Hayman
jove(1)	Jonathon Payne
kvm(3)	Peter Shipley
kvm(3)	Steve McCanne
larn(6)	Noah Morgan
lex(1)	Vern Paxson

Introduction

Welcome

Welcome to Release 1.1 of BSD/386. This system is based on the Net/2 release from the Computer Systems Research Group at the University of California (Berkeley). We have imported and tested many utilities and have commissioned others. We have fixed hundreds of bugs. We hope it meets your requirements.

Problem Resolution

BSD/386 includes 60 days of full telephone support from the day you receive the initial shipment. In the USA, call 800-ITS-BSD8 (+1 719 260 8114) to speak with a BSDI customer service agent when you have problems. The service desk is available five days a week from 9 am to 5 pm Mountain Standard Time and sometimes at other times (earlier, later, weekends). Please don't call in the middle of the night, though, as the phone is forwarded to the service agent's homes.

Customers can purchase additional support in the form of:

- Upgrades only
- Upgrades + USMail/email/fax support
- Upgrades + USMail/email/fax and phone support
- By-the-minute telephone support

BSDI does its best to respond to customer problems very quickly. Any user of BSD/386 who encounters a problem with the product is welcome to send an e-mail problem report to problems@bsdi.com. BSDI will attempt to repair any problem and incorporate the solution in its next release.

Mailing List and Support

Here are the USA contact numbers and addresses for problem reporting and support calls:

EMAIL:
support@bsdi.com (for fast response for supported customers)
problems@bsdi.com (for reporting any problems that should be fixed)
info@bsdi.com (for asking questions in general or about pricing)
bsdi-users-request@bsdi.com (to join the BSDI mailing list)

FAX: +1 719-598-4238

Voice: +1 800-ITS-BSD8; +1 719-260-8114

USMail: BSDI Bugs
7759 Delmonico Drive
Colorado Springs, CO 80919

If you would like to be placed on the general BSDI user e-mail discussion alias (bsdi-users@bsdi.com) run by Alan Clegg, please send a short e-mail request to bsdi-users-request@bsdi.com.

libm(3)	Peter McIlroy, UUNET Communications Services
locate(1)	James A. Woods
lock(1)	Bob Toxen
log(3)	Peter McIlroy
look(1)	David Hitz
ls(1)	Michael Fischbein
lsearch(3)	Roger L. Snyder
m4(1)	Ozan Yigit
mail(1)	Kurt Schoens
make(1)	Adam de Boor
mergesort(3)	Peter McIlroy
mh(1)	Rand Corporation
mille(6)	Ken Arnold
mknod(8)	Kevin Fall
monop(6)	Ken Arnold
mv(1)	Ken Smith
news(1)	Rick Adams (and a cast of thousands)
nm(1)	Hans Huebner
paste(1)	Adam S. Moskowitz
patch(1)	Larry Wall
phantasia(6)	Edward A. Estes and C. Robertson
ping(8)	Mike Muuss
pom(6)	Keith E. Brandt
primes(6)	Landon Curt Noll
qsort(3)	Jon Bentley, Doug McIlroy
quiz(6)	Jim R. Oldroyd and Keith Gabryelski
radixsort(3)	Dan Bernstein, Peter McIlroy
ranlib(1)	Hugh A. Smith
rcs(1)	Walter F. Tichy
regexp(3)	Henry Spencer
robots(6)	Ken Arnold
rogue(6)	Timothy C. Stoehr
sail(6)	David Riggle and Edward Wang
sed(1)	Diomidis Spinellis
sendmail(8)	Eric Allman
sh(1)	Kenneth Almquist
slattach(8)	Rick Adams
slip(8)	Rick Adams
tail(1)	Edward Sze-Tyan Wang
telnet(1)	Dave Borman
tn3270(1)	Gregory Minshall
traceroute(8)	Van Jacobson
trek(6)	Eric Allman
tsort(1)	Michael Rendell
unifdef(1)	Dave Yost

uniq(1)	Case Larsen
uucpd(8)	Rick Adams
uuq(1)	Lou Salkind and Rick Adams
uusnap(8)	Randy King and Rick Adams
vi(1)	Steve Kirkendall
who(1)	Michael Fischbein
window(1)	Edward Wang
worm(6)	Michael Toy
worms(6)	Eric P. Scott
write(1)	Craig Leres and Jef Poskanzer
wump(6)	Dave Taylor
xargs(1)	John B. Roll Jr.
yacc(1)	Robert Paul Corbett

This list recognizes people and organizations that have provided a specific item, program, library routine, or program maintenance for the BSD/386 system.

Network performance enhancements	Steve Nuchia, South Coast Computing Services
TNIC-1500 driver	Steve Nuchia, South Coast Computing Services
Booting larger kernels	Yuval Yarom
HP EtherTwist PC LAN Adapter	Kevin Fall
16 Plus Ethernet driver	Kevin Fall
Allied Telesis RE2000/AT-1700 series Ethernet driver	Sei Kigoshi
PS/2 mouse driver	Jan-Simon Pendry
SoundBlaster Pro driver	Steve Haehnichen
Keyboard mapping support	Dirk Huseman
Belgian keyboard map	Lode Vande Sande
Bulgarian keyboard map	Radka Kalcheva
IP Multicast integration	Steve McCanne
select.h	Chris Demetriou
NCSA Xmosaic for the X Window System	National Center for Supercomputing Applications

The following copyright notices acknowledge others' work that is included in BSD/386.

University of California Berkeley

PS/2 mouse driver

SoundBlaster Driver

Table of Contents

Distributors

Berkeley Software Design Int.
(Europe)
Katallin, Town Lane
Chartham Hatch, Canterbury
Kent CT4 7NN
United Kingdom
Phone: +44 227 781 675
Fax: +44 227 762 554
Email: bsdi@hillside.co.uk

Berkeley Software Design Int.
(Australia)
1st Floor, 2 King Street
Newtown NSW 2042
Australia
Phone: +61 2 550 5014
Fax: +61 2 519 2551
Email: enquiry@bsdi.oz.au

Nobuhiro Ohsuga
(Japan)
Foretune Co., Ltd.
2-11-3-8 Komazawa Setagaya-ku
Tokyo 154 Japan
Phone: +81 3 3667 2707
Fax: +81 3 5481 8432
Email: bsdi-info@foretune.co.jp

Daniel Kalchev
Digital Systems
BSDI Bulgaria
Neofit Bozveli 6
Varna - 9000
Bulgaria
Email: daniel@digsys.bg
Phone/Fax: +359 52 234540
Email: bsdi-info@digsys.bg

UUNET India Ltd.
(India)
Plot 270 N, Road No. 10, Jubilee Hills
Hyderabad, A.P., 500 033
INDIA
Phone: +91 842 247787
Fax: +91 842 247787
Email: bsdi@uunet.in

Yurii V. Gerasimov
(Russia)
PC-Center, Pervomaiskaya - 126
Moscow, 105203
Russia
Phone: +7 095 461 8775
Fax: +7 095 461 3514
Email: ywg@pczz.msk.su

Berkeley Software Design, Inc.
World Headquarters
7759 Delmonico Drive
Colorado Springs, CO 80919 USA
Phone: +1 800 800 4BSD
Phone: +1 719 593 9445
Fax: +1 719 598 4238
Email: bsdi-info@bsdi.com

BERKELEY SOFTWARE DESIGN, INC.

BSD/386 1.1
Release Notes
February, 1994

Written by Rob Kolstad (kolstad@bsdi.com), Tony Sanders, and Jeff Polk with comments by Donn Seeley, Trent Hein, Mike Karels, Pace Willisson, Peter Collinson, and others.

This is Version 1.1 of the release notes.

This product includes software developed by the University of California, Berkeley, Lawrence Berkeley Laboratory, and their contributors. Other credits are listed inside.