The Linux SCSI HOWTO

Drew Eckhardt,<drew@PoohSticks.ORG> (transformed to linuxdoc-sgml format by Dieter Faulbaum),
<faulbaum@bii.bessy.de>
v2.30, 30 August 1996

Contents

1 Introduction

That said, I'd appreciate it if people would ask me <drew@PoohSticks.ORG> if there's a newer version available before they publish it. When people publish outdated versions, I get questions from users that are answered in newer versions, and it reflects poorly on the publisher. I'd also prefer that all references to free distribution sites, and possibly competing distributions/products be left intact.

IMPORTANT :

BUG REPORTS OR OTHER REQUESTS FOR HELP WHICH FAIL TO FOLLOW THE PROCEDURES OUTLINED IN THE 3 SECTION WILL BE IGNORED.

This HOWTO covers the Linux SCSI subsystem, as implemented in Linux kernel revision 1.2.10 and newer alpha code. Earlier revisions of the SCSI code are _unsupported_, and may differ significantly in terms of the drivers implemented, performance, and options available.

For additional information, you may wish to join the linux-scsi mailing list by mailing majordomo@vger.rutgers.edu with the line

```
subscribe linux-scsi
```

in the text. You can unsubscribe by sending mail to the same address and including

```
unsubscribe linux-scsi
```

in the text.

Once you're subscribed, you can send mail to the list at

```
linux-scsi@vger.rutgers.edu
```

I'm aware that this document isn't the most user-friendly, and that there may be inaccuracies and oversights. If you have constructive comments on how to rectify the situation you're free to mail me about it.

2 Common Problems

This section lists some of the common problems that people have. If there is not anything here that answers your questions, you should also consult the sections for your host adapter and the devices in that are giving you problems.

2.1 General Flakiness

If you experience random errors, the most likely causes are cabling and termination problems.

Some products, such as those built around the newer NCR chips, feature digital filtering and active signal negation, and aren't very sensitive to cabling problems.

Others, such as the Adaptec 154xC, 154xCF, and 274x, are _extremely_ sensitive and may fail with cables that work with other systems.

I reiterate : some host adapters are _extremely_ sensitive to cabling and termination problems and therefore, cabling and termination should be the first things checked when there are problems.

To minimize your problems, you should use cables which

1. Claim SCSI-II compliance

2. Have a characteristic impedance of 132 ohms

3. All come from the same source to avoid impedance mismatches
4. Come from a reputable vendor such as Amphenol

Termination power should be provided by _all_ devices on the SCSI bus, through a diode to prevent current backflow, so that sufficient power is available at the ends of the cable where it is needed. To prevent damage if the bus is shorted, TERMPWR should be driven through a fuse or other current limiting device.

If multiple devices, external cables, or FAST SCSI 2 are used, active or forced perfect termination should be used on both ends of the SCSI bus.

See the Comp.Periphs.Scsi FAQ (available on tsx-11 in pub/linux/ALPHA/scsi) for more information about active termination.

2.2 The kernel command line

Other parts of the documentation refer to a "kernel command line".

The kernel command line is a set of options you may specify from either the LILO : prompt after an image name, or in the append field in your LILO configuration file (LILO .14 and newer use /etc/lilo.conf, older versions use /etc/lilo/config).

Boot your system with LILO, and hit one of the alt, control, or shift keys when it first comes up to get a prompt. LILO should respond with

```
:
```

At this prompt, you can select a kernel image to boot, or list them with ?. Ie

```
:?

ramdisk floppy harddisk
```

To boot that kernel with the command line options you have selected, simply enter the name followed by a white space delimited list of options, terminating with a return.

Options take the form of

```
variable=valuelist
```

Where valuelist may be a single value or comma delimited list of values with no whitespace. With the exception of root device, individual values are numbers, and may be specified in either decimal or hexadecimal.

Ie, to boot linux with an Adaptec 1520 clone not recognized at bootup, you might type

```
:floppy aha152x=0x340,11,7,1
```

If you don't care to type all of this at boot time, it is also possible to use the LILO configuration file "append" option with LILO .13 and newer.

Ie,

```
append="aha152x=0x340,11,7,1"
```

2.3 A SCSI device shows up at all possible IDs

If this is the case, you have strapped the device at the same address as the controller (typically 7, although some boards use other addresses, with 6 being used by some Future Domain boards).

Please change the jumper settings.

2.4 A SCSI device shows up at all possible LUNs

The device has buggy firmware.

As an interim solution, you should try using the kernel command line option

```
max_scsi_luns=1
```

If that works, there is a list of buggy devices in the kernel sources in drivers/scsi/scsi.c in the variable blacklist. Add your device to this list and mail the patch to Linus Torvalds <Linus.Torvalds@cs.Helsinki.FI>.

2.5 You get sense errors when you know the devices are error free

Sometimes this is caused by bad cables or improper termination.

See section 2.1

2.6 A kernel configured with networking does not work

The auto-probe routines for many of the network drivers are not passive, and will interfere with operation with some of the SCSI drivers.

2.7 Device detected, but unable to access

A SCSI device is detected by the kernel, but you are unable to access it - ie mkfs /dev/sdc, tar xvf /dev/rst2, etc fails.

You don't have a special file in /dev for the device.

Unix devices are identified as either block or character (block devices go through the buffer cache, character devices do not) devices, a major number (ie which driver is used - block major 8 corresponds to

SCSI disks) and a minor number (ie which unit is being accessed through a given driver - ie character major 4, minor 0 is the first virtual console, minor 1 the next, etc). However, accessing devices through this separate namespace would break the unix/Linux metaphor of "everything is a file," so character and block device special files are created under /dev. This lets you access the raw third SCSI disk device as /dev/sdc, the first serial port as /dev/ttyS0, etc.

The preferred method for creating a file is using the MAKEDEV script - cd /dev

and run MAKEDEV (as root) for the devices you want to create - ie

```
./MAKEDEV sdc
```

wildcards "should" work - ie

```
./MAKEDEV sd\*
```

"should" create entries for all SCSI disk devices (doing this should create /dev/sda through /dev/sdp, with fifteen partition entries for each)

```
./MAKEDEV sdc\*
```

"should" create entries for /dev/sdc and all fifteen permissible partitions on /dev/sdc, etc.

I say "should" because this is the standard unix behavior - the MAKEDEV script in your installation may not conform to this behavior, or may have restricted the number of devices it will create.

If MAKEDEV won't do the right magic for you, you'll have to create the device entries by hand with the mknod command.

The block/character type, major, and minor numbers are specified for the various SCSI devices in section 6.3 in the appropriate section.

Take those numbers, and use (as root)

```
mknod /dev/device b|c major minor
```

ie -

```
mknod /dev/sdc b 8 32
mknod /dev/rst0 c 9 0
```

2.8 SCSI System Lockups

This could be one of a number of things. Also see the section for your specific host adapter for possible further solutions.

There are cases where the lockups seem to occur when multiple devices are in use at the same time. In this case, you can try contacting the manufacturer of the devices and see if firmware upgrades are available which would correct the problem. If possible, try a different scsi cable, or try on another system. This can also be caused by bad blocks on disks, or by bad handling of DMA by the motherboard (for host adapters that do DMA). There are probably many other possible conditions that could lead to this type of event.

Sometimes these problems occur when there are multiple devices in use on the bus at the same time. In this case, if your host adapter driver supports more than one outstanding command on the bus at one time, try reducing this to 1 and see if this helps. If you have tape drives or slow cdrom drives on the bus, this might not be a practical solution.

2.9 Configuring and building the kernel

Unused SCSI drivers eat up valuable memory, aggravating memory shortage problems on small systems because kernel memory is unpagable.

So, you will want to build a kernel tuned for your system, with only the drivers you need installed.

```
cd to /usr/src/linux
```

If you are using a root device other than the current one, or something other than 80x25 VGA, and you are writing a boot floppy, you should edit the makefile, and make sure the

```
ROOT_DEV =
```

and

```
SVGA_MODE =
```

lines are the way you want them.

If you've installed any patches, you may wish to guarantee that all files are rebuilt. If this is the case, you should type

```
make mrproper
```

Irregardless of weather you ran make mrproper, type

```
make config
```

and answer the configuration questions. Then run

```
make depend
```

and finally

```
make
```

Once the build completes, you may wish to update the lilo configuration, or write a boot floppy. A boot floppy may be made by running

```
make zdisk
```

2.10 LUNS other than 0 don't work

Many SCSI devices are horrendously broken, lock the SCSI bus up solid, and do other bad things when you attempt to talk to them at a logical unit someplace other than zero.

So, by default recent versions of the Linux kernel will not probe luns other than 0. To work around this, you need to the max_scsi_luns command line option, or recompile the kernel with the CONFIG_SCSI_MULTI_LUN option.

Usually, you'll put

```
max_scsi_luns=8
```

on your LILO command line.

If your multi-LUN devices still aren't detected correctly after trying one of these fixes (as the case will be with many old SCSI->MFM, RLL, ESDI, SMD, and similar bridge boards), you'll be thwarted by this piece of code

```
/* Some scsi-1 peripherals do not handle lun != 0.
   I am assuming that scsi-2 peripherals do better */
if((scsi_result[2] & 0x07) == 1 &&
   (scsi_result[3] & 0x0f) == 0) break;
```

in scan_scsis() in drivers/scsi/scsi.c. Delete this code, and you should be fine.

3 Reporting Bugs

The Linux SCSI developers don't necessarily maintain old revisions of the code due to space constraints. So, if you are not running the latest publically released Linux kernel (note that many of the Linux distributions, such as MCC, SLS, Yggdrasil, etc. often lag one or even twenty patches behind this) chances are we will be unable to solve your problem. So, before reporting a bug, please check to see if it exists with the latest publically available kernel.

If after upgrading, and reading this document thoroughly, you still believe that you have a bug, please mail a bug report to the SCSI channel of the mailing list where it will be seen by many of the people who've contributed to the Linux SCSI drivers.

In your bug report, please provide as much information as possible regarding your hardware configuration, the exact text of

all of the messages that Linux prints when it boots, when the error condition occurs, and where in the source code the error is. Use the procedures outlined in 3.1 and 3.2.

Failure to provide the maximum possible amount of information may result in misdiagnosis of your problem, or developers deciding that there are other more interesting problems to fix.

The bottom line is that if we can't reproduce your bug, and you can't point at us what's broken, it won't get fixed.

3.1 Capturing messages

If you are not running a kernel message logging system :

Insure that the /proc filesystem is mounted.

```
grep proc /etc/mtab
```

If the /proc filesystem is not mounted, mount it

```
mkdir /proc
chmod 755 /proc
mount -t proc /proc /proc
```

Copy the kernel revision and messages into a log file

```
cat /proc/version > /tmp/log
cat /proc/kmsg >> /tmp/log
```

Type CNTRL-C after a second or two.

If you are running some logger, you'll have to poke through the appropriate log files (/etc/syslog.conf should be of some use in locating them), or use dmesg.

If Linux is not yet bootstrapped, format a floppy diskette under DOS. Note that if you have a distribution which mounts the root diskette off of floppy rather than RAM drive, you'll have to format a diskette readable in the drive not being used to mount root or use their ramdisk boot option.

Boot Linux off your distribution boot floppy, preferably in single user mode using a RAM disk as root.

```
mkdir /tmp/dos
```

Insert the diskette in a drive not being used to mount root, and mount it. Ie

```
mount -t msdos /dev/fd0 /tmp/dos
```

or

```
mount -t msdos /dev/fd1 /tmp/dos
```

Copy your log to it

```
cp /tmp/log /tmp/dos/log
```

Unmount the DOS floppy

```
umount /tmp/dos
```

And shutdown Linux

```
shutdown
```

Reboot into DOS, and using your favorite communications software include the log file in your trouble mail.

3.2 Locating the source of a panic()

Like other unices, when a fatal error is encountered, Linux calls the kernel panic() function. Unlike other unices, Linux doesn't dump core to the swap or dump device and reboot automatically. Instead, a useful summary of state information is printed for the user to manually copy down. Ie :

```
Unable to handle kernel NULL pointer dereference at virtual address c0000004
current->tss,cr3 = 00101000, %cr3 = 00101000
*pde = 00102027
*pte = 00000027
Oops: 0000
EIP:    0010:0019c905
EFLAGS: 00010002
eax: 0000000a   ebx: 001cd0e8   ecx: 00000006   edx: 000003d5
esi: 001cd0a8   edi: 00000000   ebp: 00000000   esp: 001a18c0
ds: 0018   es: 0018   fs: 002b   gs: 002b   ss: 0018
Process swapper (pid: 0, process nr: 0, stackpage=001a09c8)
Stack: 0019c5c6 00000000 0019c5b2 00000000 0019c5a5 001cd0a8 00000002 00000000
       001cd0e8 001cd0a8 00000000 001cdb38 001cdb00 00000000 001ce284 0019d001
       001cd004 0000e800 fbfff000 0019d051 001cd0a8 00000000 001a29f4 00800000
Call Trace: 0019c5c6 0019c5b2 0018c5a5 0019d001 0019d051 00111508 00111502
            0011e800 0011154d 00110f63 0010e2b3 0010ef55 0010ddb7
Code: 8b 57 04 52 68 d2 c5 19 00 e8 cd a0 f7 ff 83 c4 20 8b 4f 04
Aiee, killing interrupt handler
kfree of non-kmalloced memory: 001a29c0, next= 00000000, order=0
task[0] (swapper) killed: unable to recover
Kernel panic: Trying to free up swapper memory space
In swapper task - not syncing
```

Take the hexadecimal number on the EIP: line, in this case 19c905, and search through /usr/src/linux/zSystem.map for the highest number not larger than that address. Ie,

```
0019a000 T _fix_pointers
0019c700 t _intr_scsi
0019d000 t _NCR53c7x0_intr
```

That tells you what function its in. Recompile the source file which defines that function file with debugging enabled, or the whole kernel if you prefer by editing /usr/src/linux/Makefile and adding a "-g" to the CFLAGS definition.

```
#
# standard CFLAGS
#
```

Ie,

```
CFLAGS = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -pipe
```

becomes

```
CFLAGS = -g -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -pipe
```

Rebuild the kernel, incrementally or by doing a

```
make clean
make
```

Make the kernel bootable by creating an entry in your /etc/lilo.conf for it

```
image = /usr/src/linux/zImage
label = experimental
```

and re-running LILO as root, or by creating a boot floppy

```
make zImage
```

Reboot and record the new EIP for the error.

If you have script installed, you may want to start it, as it will log your debugging session to the typescript file.

Now, run

```
gdb /usr/src/linux/tools/zSystem
```

and enter

```
info line *<your EIP>
```

Ie,

```
info line *0x19c905
```

To which GDB will respond something like

```
(gdb) info line *0x19c905
Line 2855 of "53c7,8xx.c" starts at address 0x19c905 <intr_scsi+641&>
   and ends at 0x19c913 <intr_scsi+655>.
```

Record this information. Then, enter

```
list <line number>
```

Ie,

```
(gdb) list 2855
2850    /*      printk("scsi\%d : target \%d lun \%d unexpected disconnect\n",
2851                host->host_no, cmd->cmd->target, cmd->cmd->lun); */
2852            printk("host : 0x\%x\n", (unsigned) host);
2853            printk("host->host_no : %d\n", host->host_no);
2854            printk("cmd : 0x%x\n", (unsigned) cmd);
2855            printk("cmd->cmd : 0x%x\n", (unsigned) cmd->cmd);
2856            printk("cmd->cmd->target : %d\n", cmd->cmd->target);
2857            if (cmd) {;
2858                abnormal_finished(cmd, DID_ERROR << 16);
2859            }
2860            hostdata->dsp = hostdata->script + hostdata->E_schedule /
2861                sizeof(long);
2862            hostdata->dsp_changed = 1;
2863        /* SCSI PARITY error */
2864        }
2865
2866        if (sstat0_sist0 & SSTAT0_PAR) {
2867            fatal = 1;
2868            if (cmd && cmd->cmd) {
2869                printk("scsi%d : target %d lun %d parity error.\n",
```

Obviously, quit will take you out of GDB.

Record this information too, as it will provide a context in case the developers' kernels differ from yours.

4 Modules

This section gives specific details regarding the support for loadable kernel modules and how it relates to SCSI.

4.1 General Information

Loadable modules are a means by which the user or system administrator can load files into the kernel's memory in such a way that the kernel's capabilities are expanded. The most common usages of modules are for drivers to support hardware, or to load filesytems.

There are several advantages of modules for SCSI. One is that a system administrator trying to maintain a large number of machines can use a single kernel image for all of the machines, and then load kernel modules to support hardware that is only present on some machines.

It is also possible for someone trying to create a distribution to use a script on the bootable floppy to query for which modules to be loaded. This saves memory that would otherwise be wasted on unused drivers, and it would also reduce the possibility that a probe for a non-existent card would screw up some other card on the system.

Modules also work out nicely on laptops, which tend to have less memory than desktop machines, and people tend to want to keep the kernel image as small as possible and load modules as required. Also, modules makes supporting PCMCIA SCSI cards on laptops somewhat easier, since you can load and unload the driver as the card is inserted/removed. [Note: currently the qlogic and 152x drivers support PCMCIA].

Finally, there is the advantage that kernel developers can more easily debug and test their drivers, since testing a new driver does not require rebooting the machine (provided of course that the machine has not completely crashed as a result of some bug in the driver).

Although modules are very nice, there is one limitation. If your root disk partition is on a scsi device, you will not be able to use modularized versions of scsi code required to access the disk. This is because the system must be able to mount the root partition before it can load any modules from disk. There are people thinking about ways of fixing the loader and the kernel so that the kernel can self-load modules prior to attempting to mount the root filesystem, so at some point in the future this limitation may be lifted.

4.2 Module support in the 1.2.N kernel

In the 1.2.N series of kernels, there is partial support for SCSI kernel modules. While none of the high level drivers (such as disk, tape, etc) can be used as modules, most of the low level drivers (i.e. 1542, 1522) can be loaded and unloaded as required. Each time you load a low-level driver, the driver first searches for cards that can be driven. Next, the bus is scanned for each card that is found, and then the internal data structures are set up so as to make it possible to actually use the devices attached to the cards that the driver is managing.

When you are through with a low-level driver, you can unload it. You should keep in mind that usage counts are maintained based upon mounted filesystems, open files, etc, so that if you are still using a device that the driver is managing, the rmmod utility will tell you that the device is still busy and refuse to unload the driver. When the driver is unloaded, all of the associated data structures are also freed so that the system state should be back to where it was before the module was loaded. This means that the driver could be reloaded at a later time if required.

4.3 Module support in the 1.3.N kernel

In the 1.3 series of kernels, the scsi code is completely modularized. This means that you can start with a kernel that has no scsi support whatsoever, and start loading modules and you will eventually end up with complete support.

If you wish, you can compile some parts of the SCSI code into the kernel and then load other parts later - it is all up to you how much gets loaded at runtime and how much is linked directly into the kernel.

If you are starting with a kernel that has no support whatsoever for SCSI, then the first thing you will need to do is to load the scsi core into the kernel - this is in a module called "scsi_mod". You will not be able to load any other scsi modules until you have this loaded into kernel memory. Since this does not contain any low-level drivers, the act of loading this module will not scan any busses, nor will it activate any drivers for scsi disks, tapes, etc. If you answered 'Y' to the CONFIG_SCSI question when you built your kernel, you will not need to load this module.

At this point you can add modules in more or less any order to achieve the desired functionality. Usage counts are interlocks are used to prevent unloading of any component which might still be in use, and you will get a message from rmmod if a module is still busy.

The high level drivers are in modules named "sd_mod", "sr_mod", "st", and "sg", for disk, cdrom, tape, and scsi generics support respectively. When you load a high level driver, the device list for all attached hosts is examined for devices which the high level driver can drive, and these are automatically activated.

The use of modules with low level drivers were described in the section of the 4.2. When a low-level driver is loaded, the bus is scanned, and each device is examined by each of the high level drivers to see if they recognize it as something that they can drive - anything recognized is automatically attached and activated.

5 Hosts

This section gives specific information about the various host adapters that are supported in some way or another under linux.

5.1 Supported and Unsupported Hardware

Drivers in the distribution kernel :

Adaptec 152x, Adaptec 154x (DTC 329x boards usually work, but are unsupported), Adaptec 174x, Adaptec 274x/284x (294x support requires a newer version of the driver), BusLogic MultiMaster Host Adapters, EATA-DMA and EATA-PIO protocol compilant boards (DPT PM2001, PM2011, PM2012A, PM2012B, PM2021, PM2022, PM2024, PM2122, PM2124, PM2322, PM2041, PM2042, PM2044, PM2142, PM2144, PM2322, PM3021, PM3122, PM3222, PM3224, PM3334 some boards from NEC, AT&T, SNI, AST, Olivetti, and Alphatronix), Future Domain 850, 885, 950, and other boards in that series (but not the 840, 841, 880, and 881 boards unless you make the appropriate patch), Future Domain 16x0 with TMC-1800, TMC-18C30, or TMC-18C50 chips, NCR53c8xx,PAS16 SCSI ports, Seagate ST0x, Trantor T128/T130/T228 boards, Ultrastor 14F, 24F, and 34F, and Western Digital 7000.

MCA :

MCA boards which are compatible with a supported board (ie, Adaptec 1640 and BusLogic 640) will work.

Alpha drivers :

Many ALPHA drivers are available at

```
ftp://tsx-11.mit.edu/pub/linux/ALPHA/scsi
```

Drivers which will work with modifications

NCR53c8x0/7x0:

```
   A NCR53c8xx driver has been developed, but currently will not work
with NCR53c700, NCR53c700-66, NCR53c710, and NCR53c720 chips.  A list
of changes needed to make each of these chips work follows, as well
as a summary of the complexity.

NCR53c720 (trivial) - detection changes, initialization changes, change
    fixup code to translate '810 register addresses to
    '7xx mapping.

NCR53c710 (trivial) - detection changes, initialization changes,
    of assembler, change fixup code to translate '810 register
    addresses to '7xx mapping, change interrupt handlers to treat
    IID interrupt from INTFLY instruction to emulate it.

NCR53c700, NCR53c700-66 (very messy) - detection changes,
    initialization changes, modification of NCR code to not use DSA,
    modification of Linux code to handle context switches.
```

SCSI hosts that will not work :

All parallel->SCSI adapters, Rancho SCSI boards, and Grass Roots SCSI boards. BusLogic FlashPoint boards, such as the BT-930/932/950, are currently unsupported.

SCSI hosts that will NEVER work :

Non Adaptec compatible, non NCR53c8xx DTC boards (including the 3270 and 3280).

CMD SCSI boards.

Acquiring programming information requires a non-disclosure agreement with DTC/CMD. This means that it would be impossible to distribute a Linux driver if one were written, since complying with the NDA would mean distributing no source, in violation of the GPL, and complying with the GPL would mean distributing source, in violation of the NDA.

If you want to run Linux on some other unsupported piece of hardware, your options are to either write a driver yourself (Eric Youngdale and I are usually willing to answer technical questions concerning the Linux SCSI drivers) or to commission a driver (Normal consulting rates mean that this will not be a viable option for personal use).

5.1.1 Multiple host adapters

With some host adapters (see 10.7), you can use multiple host adapters of the same type in the same system. With multiple adapters of the same type in the same system, generally the one at the lowest address will be scsi0, the one at the next address scsi1, etc.

In all cases, it is possible to use multiple host adapters of different types, provided that none of their addresses conflict. SCSI controllers are scanned in the order specified in the builtin_scsi_hosts[] array in drivers/scsi/hosts.c, with the order currently being

> BusLogic, Ultrastor 14/34F, Ultrastor 14F,, Adaptec 151x/152x, Adaptec 154x, Adaptec 174x, AIC7XXX, AM53C974, Future Domain 16x0, Always IN2000, Generic NCR5380, QLOGIC, PAS16, Seagate, Trantor T128/T130, NCR53c8xx, EATA-DMA, WD7000, debugging driver.

In most cases (ie, you aren't trying to use both BusLogic and Adaptec drivers), this can be changed to suit your needs (ie, keeping the same devices when new SCSI devices are added to the system on a new controller) by moving the individual entries.

5.2 Common Problems

5.2.1 SCSI timeouts

Make sure interrupts are enabled correctly, and there are no IRQ, DMA, or address conflicts with other boards.

5.2.2 Failure of autoprobe routines on boards that rely on BIOS for autoprobe.

If your SCSI adapter is one of the following :

> Adaptec 152x, Adaptec 151x, Adaptec AIC-6260, Adaptec AIC-6360, Future Domain 1680, Future Domain TMC-950, Future Domain TMC-8xx, Trantor T128, Trantor T128F, Trantor T228F, Seagate ST01, Seagate ST02, or a Western Digital 7000

and it is not detected on bootup, ie you get a

```
scsi : 0 hosts
```

message or a

```
scsi%d : type
```

message is not printed for each supported SCSI adapter installed in the system, you may have a problem with the autoprobe routine not knowing about your board.

Autodetection will fail for drivers using the BIOS for autodetection if the BIOS is disabled. Double check that your BIOS is enabled, and not conflicting with any other peripherial BIOSes.

Autodetection will also fail if the board's "signature" and/or BIOS address don't match known ones.

If the BIOS is installed, please use DOS and DEBUG to find a signature that will detect your board - Ie, if your board lives at 0xc8000, under DOS do

```
debug
d c800:0
q
```

and send a message to the SCSI channel of the mailing list with the ASCII message, with the length and offset from the base address (ie, 0xc8000). Note that the EXACT text is required, and you should provide both the hex and ASCII portions of the text.

If no BIOS is installed, and you are using an Adaptec 152x, Trantor T128, or Seagate driver, you can use command line or compile time overrides to force detection.

Please consult the appropriate subsection for your SCSI board as well as section 2.1.

5.2.3 Failure of boards using memory mapped I/O

(This include the Trantor T128 and Seagate boards, but not the Adaptec, Generic NCR5380, PAS16, and Ultrastor drivers)

This is often caused when the memory mapped I/O ports are incorrectly cached. You should have the board's address space marked as uncachable in the XCMOS settings.

If this is not possible, you will have to disable cache entirely.

If you have manually specified the address of the board, remember that Linux needs the actual address of the board, and not the 16 byte segment the documentation may refer to.

Ie, 0xc8000 would be correct, 0xc800 would not work and could cause memory corruption.

5.2.4 "kernel panic : cannot mount root device" when booting an ALPHA driver boot floppy

You'll need to edit the binary image of the kernel (before or after writing it out to disk), and modify a few two byte fields (little endian) to guarantee that it will work on your system.

1. default swap device at offset 502, this should be set to 0x00 0x00
2. ram disk size at offset 504, this should be set to the size of the boot floppy in K - ie, 5.25" = 1200, 3.5" = 1440.

   ```
        This means the bytes are

   3.5" : 0xA0 0x05
   5.25" : 0xB0 0x04
   ```

3. root device offset at 508, this should be 0x00 0x00, ie the boot device.

dd or rawrite the file to a disk. Insert the disk in the first floppy drive, wait until it prompts you to insert the root disk, and insert the root floppy from your distribution.

5.2.5 Installing a device driver not included with the distribution kernel

You need to start with the version of the kernel used by the driver author. This revision may be alluded to in the documentation included with the driver.

Various recent kernel revisions can be found at

```
nic.funet.fi:/pub/OS/Linux/PEOPLE/Linus
```

as linux-version.tar.gz

They are also mirrored at tsx-11.mit.edu and various other sites.

```
cd to /usr/src.
```

Remove your old Linux sources, if you want to keep a backup copy of them

```
mv linux linux-old
```

Untar the archive

```
gunzip < linux-0.99.12.tar.gz | tar xvfp -
```

Apply the patches. The patches will be relative to some directory in the filesystem. By examining the output file lines in the patch file (grep for ^—), you can tell where this is - ie patches with these lines

```
--- ./kernel/blk_drv/scsi/Makefile

--- ./config.in Wed Sep  1 16:19:33 1993
```

would have the files relative to /usr/src/linux.

Untar the driver sources at an appropriate place - you can type

```
tar tfv patches.tar
```

to get a listing, and move files as necessary (The SCSI driver files should live in /usr/src/linux/kernel/drivers/scsi)

Either cd to the directory they are relative to and type

```
patch -p0 < patch_file
```

or tell patch to strip off leading path components. Ie, if the files started with

```
--- linux-new/kernel/blk_drv/scsi/Makefile
```

and you wanted to apply them while in /usr/src/linux, you could cd to /usr/src/linux and type

```
patch -p1 < patches
```

to strip off the "linux-new" component.

After you have applied the patches, look for any patch rejects, which will be the name of the rejected file with a # suffix appended.

```
find /usr/src/linux/ -name "*#" -print
```

If any of these exist, look at them. In some cases, the differences will be in RCS identifiers and will be harmless, in other cases, you'll have to manually apply important parts. Documentation on diff files and patch is beyond the scope of this document.

See also 2.9.

5.2.6 Installing a driver that has no patches

In some cases, a driver author may not offer patches with the .c and .h files which comprise his driver, or the patches may be against an older revision of the kernel and not go in cleanly.

1. Copy the .c and .h files into /usr/src/linux/drivers/scsi
2. Add the configuration option

   Edit /usr/src/linux/config.in, and add a line in the

   ```
   *
   * SCSI low-level drivers
   *
   ```

   section, add a boolean configuration variable for your driver. Ie,

   ```
   bool 'Always IN2000 SCSI support' CONFIG_SCSI_IN2000 y
   ```

3. Add the makefile entries

   Edit /usr/src/linux/drivers/scsi/Makefile, and add an entry like

   ```
   ifdef CONFIG_SCSI_IN2000
   SCSI_OBS := $(SCSI_OBJS) in2000.o
   SCSI_SRCS := $(SCSI_SRCS) in2000.c
   endif
   ```

   before the

   ```
   scsi.a: $(SCSI_OBJS)
   ```

line in the makefile, where the .c file is the .c file you copied in, and the .o file is the basename of the .c file with a .o suffixed.

4. Add the entry points

   Edit /usr/src/linux/drivers/scsi/hosts.c, and add a #include for the header file, conditional on the CONFIG_SCSI preprocessor define you added to the configuration file. Ie, after

   ```
   #ifdef CONFIG_SCSI_GENERIC_NCR5380
   #include "g_NCR5380.h"
   #endif
   ```

   you might add

   ```
   #ifdef CONFIG_SCSI_IN2000
   #include "in2000.h"
   #endif
   ```

   You will also need to add the Scsi_Host_Template entry into the scsi_hosts[] array. Take a look into the .h file, and you should find a #define that looks something like this :

   ```
   #define IN2000 {"Always IN2000", in2000_detect, \
       in2000_info, in2000_command,    \
       in2000_queuecommand,            \
       in2000_abort,                   \
       in2000_reset,                   \
       NULL,                           \
       in2000_biosparam,               \
       1, 7, IN2000_SG, 1, 0, 0}
   ```

   the name of the preprocessor define, and add it into the scsi_hosts[] array, conditional on definition of the preprocessor symbol you used in the configuration file.

   Ie, after

   ```
   #ifdef CONFIG_SCSI_GENERIC_NCR5380
          GENERIC_NCR5380,
   #endif
   ```

   you might add

   ```
   #ifdef CONFIG_SCSI_IN2000
          IN2000,
   #endif
   ```

   See also 2.9.

5.2.7 Failure of a PCI board in a Compaq System

A number of Compaq systems map the 32-bit BIOS extensions used to probe for PCI devices into memory which is inaccessible to the Linux kernel due to the memory layout. If Linux is unable to detect a supported PCI SCSI board, and the kernel tells you something like

```
pcibios_init: entry in high memory, unable to access
```

Grab

```
ftp://ftp.compaq.com/pub/softpaq/Software-Solutions/sp0921.zip
```

which is a self-extracting archive of a program which will relocate the BIOS32 code.

5.2.8 A SCSI system with PCI boards hangs after the %d Hosts message

Some PCI systems have broken BIOSes which disable interrupts and fail to reenable them before returning control to the caller. The following patch fixes this

```
--- bios32.c.orig       Mon Nov 13 22:35:31 1995
+++ bios32.c    Thu Jan 18 00:15:09 1996
@@ -56,6 +56,7 @@
 #include <linux/pci.h>

 #include <asm/segment.h>
+#include <asm/system.h>

 #define PCIBIOS_PCI_FUNCTION_ID        0xb1XX
 #define PCIBIOS_PCI_BIOS_PRESENT       0xb101
@@ -125,7 +126,9 @@
        unsigned long address;          /* %ebx */
        unsigned long length;           /* %ecx */
        unsigned long entry;            /* %edx */
+       unsigned long flags;

+       save_flags(flags);
        __asm__("lcall (%%edi)"
                : "=a" (return_code),
                  "=b" (address),
@@ -134,6 +137,7 @@
                : "0" (service),
                  "1" (0),
                  "D" (&bios32_indirect));
+       restore_flags(flags);

        switch (return_code) {
```

```
                case 0:
@@ -161,11 +165,13 @@
        unsigned char present_status;
        unsigned char major_revision;
        unsigned char minor_revision;
+       unsigned long flags;
        int pack;

        if ((pcibios_entry = bios32_service(PCI_SERVICE))) {
                pci_indirect.address = pcibios_entry;

+               save_flags(flags);
                __asm__("lcall (%%edi)\n\t"
                        "jc 1f\n\t"
                        "xor %%ah, %%ah\n"
@@ -176,6 +182,7 @@
                        : "1" (PCIBIOS_PCI_BIOS_PRESENT),
                          "D" (&pci_indirect)
                        : "bx", "cx");
+               restore_flags(flags);

                present_status = (pack >> 16) & 0xff;
                major_revision = (pack >> 8) & 0xff;
@@ -210,7 +217,9 @@
 {
        unsigned long bx;
        unsigned long ret;
+       unsigned long flags;

+       save_flags(flags);
        __asm__ ("lcall (%%edi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -221,6 +230,7 @@
                  "c" (class_code),
                  "S" ((int) index),
                  "D" (&pci_indirect));
+       restore_flags(flags);
        *bus = (bx >> 8) & 0xff;
        *device_fn = bx & 0xff;
        return (int) (ret & 0xff00) >> 8;
@@ -232,7 +242,9 @@
 {
        unsigned short bx;
        unsigned short ret;
+       unsigned long flags;

+       save_flags(flags);
```

```
        __asm__("lcall (%%edi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -244,6 +256,7 @@
                  "d" (vendor),
                  "S" ((int) index),
                  "D" (&pci_indirect));
+       restore_flags(flags);
        *bus = (bx >> 8) & 0xff;
        *device_fn = bx & 0xff;
        return (int) (ret & 0xff00) >> 8;
@@ -254,7 +267,9 @@
 {
        unsigned long ret;
        unsigned long bx = (bus << 8) | device_fn;
+       unsigned long flags;

+       save_flags (flags);
        __asm__("lcall (%%esi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -273,7 +288,9 @@
 {
        unsigned long ret;
        unsigned long bx = (bus << 8) | device_fn;
+       unsigned long flags;

+       save_flags(flags);
        __asm__("lcall (%%esi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -292,7 +309,9 @@
 {
        unsigned long ret;
        unsigned long bx = (bus << 8) | device_fn;
+       unsigned long flags;

+       save_flags(flags);
        __asm__("lcall (%%esi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -303,6 +322,7 @@
                  "b" (bx),
                  "D" ((long) where),
                  "S" (&pci_indirect));
+       restore_flags(flags);
        return (int) (ret & 0xff00) >> 8;
 }
```

```
@@ -311,7 +331,9 @@
 {
        unsigned long ret;
        unsigned long bx = (bus << 8) | device_fn;
+       unsigned long flags;

+       save_flags(flags);
        __asm__("lcall (%%esi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -322,6 +344,7 @@
                  "b" (bx),
                  "D" ((long) where),
                  "S" (&pci_indirect));
+       restore_flags(flags);
        return (int) (ret & 0xff00) >> 8;
 }

@@ -330,7 +353,9 @@
 {
        unsigned long ret;
        unsigned long bx = (bus << 8) | device_fn;
+       unsigned long flags;

+       save_flags(flags);
        __asm__("lcall (%%esi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -341,6 +366,7 @@
                  "b" (bx),
                  "D" ((long) where),
                  "S" (&pci_indirect));
+       restore_flags(flags);
        return (int) (ret & 0xff00) >> 8;
 }

@@ -349,7 +375,9 @@
 {
        unsigned long ret;
        unsigned long bx = (bus << 8) | device_fn;
+       unsigned long flags;

+       save_flags(flags);
        __asm__("lcall (%%esi)\n\t"
                "jc 1f\n\t"
                "xor %%ah, %%ah\n"
@@ -360,6 +388,7 @@
```

```
                    "b" (bx),
                    "D" ((long) where),
                    "S" (&pci_indirect));
+       restore_flags(flags);
        return (int) (ret & 0xff00) >> 8;
 }
```

5.3 Adaptec 152x, 151x, 1505, 282x, Sound Blaster 16 SCSI, SCSI Pro, Gigabyte, and other AIC 6260/6360 based products (Standard)

Supported Configurations :

```
BIOS addresses : 0xd8000, 0xdc000, 0xd0000, 0xd4000, 0xc8000, 0xcc000, 0xe0000,
                 0xe4000.
Ports          : 0x140, 0x340
IRQs           : 9, 10, 11, 12
DMA            : not used
IO             : port mapped
```

Autoprobe :

```
Works with many boards with an installed BIOS.  All
other boards, including the Adaptec 1510, and Sound Blaster16 SCSI
must use a kernel command line or compile time override.
```

Autoprobe Override :

Compile time :

```
Define PORTBASE, IRQ, SCSI_ID, RECONNECT, PARITY as appropriate, see Defines
```

kernel command line :

```
aha152x=<PORTBASE>[,<IRQ>[,<SCSI-ID>[,<RECONNECT>[,<PARITY>]]]]
```

SCSI-ID is the SCSI ID of the HOST adapter, not of any devices you have installed on it. Usually, this should be 7.

To force detection at 0x340, IRQ 11, at SCSI-ID 7, allowing disconnect/reconnect, you would use the following command line option :

```
aha152x=0x340,11,7,1
```

Antiquity Problems, fix by upgrading :

1. The driver fails with VLB boards. There was a timing problem in kernels older than revision 1.0.5.

Defines :

```
AUTOCONF       : use configuration the controller reports (only 152x)
IRQ            : override interrupt channel (9,10,11 or 12) (default 11)
SCSI_ID        : override SCSI ID of AIC-6260 (0-7) (default 7)
RECONNECT      : override target disconnect/reselect (set to non-zero to
                 allow, zero to disable)
DONT_SNARF     : Don't register ports (pl12 and below)
SKIP_BIOSTEST  : Don't test for BIOS signature (AHA-1510 or disabled BIOS)
PORTBASE       : Force port base. Don't try to probe
```

5.4 Adaptec 154x, AMI FastDisk VLB, DTC 329x (Standard)

Supported Configurations :

```
Ports          : 0x330 and 0x334
IRQs           : 9, 10, 11, 12, 14, 15
DMA channels   : 5, 6, 7
IO             : port mapped, bus master
```

Autoprobe :

```
will detect boards at 0x330 and 0x334 only.
```

Autoprobe override :

```
aha1542=<PORTBASE>[,<BUSON>,<BUSOFF>[,<DMASPEED>]]
```

Notes:

1. BusLogic makes a series of boards that are software compatible with the Adaptec 1542, and these come in ISA, VLB, EISA, and PCI flavors.

2. No-suffix boards, and early 'A' suffix boards do not support scatter/gather, and thus don't work. However, they can be made to work for some definition of the word works if AHA1542_SCATTER is changed to 0 in drivers/scsi/aha1542.h.

Antiquity Problems, fix by upgrading :

1. Linux kernel revisions prior to .99.10 don't support the 'C' revision.

2. Linux kernel revisions prior to .99.14k don't support the 'C' revision options for

   - BIOS support for the extended mapping for disks $>$ 1G
   - BIOS support for $>$ 2 drives

- BIOS support for autoscanning the SCSI bus

3. Linux kernel revisions prior to .99.15e don't support the 'C' with the BIOS support for > 2 drives turned on and the BIOS support for the extended mapping for disks > 1G turned off.
4. Linux kernel revisions prior to .99.14u don't support the 'CF' revisions of the board.
5. Linux kernel revisions prior to 1.0.5 have a race condition when multiple devices are accessed at the same time.

Common problems :

1. There are unexpected errors with a 154xC or 154xCF board,

   Early examples of the 154xC boards have a high slew rate on one of the SCSI signals, which results in signal reflections when cables with the wrong impedance are used. Newer boards aren't much better, and also suffer from extreme cabling and termination sensitivity.

   See also Common Problems 2 and 3 and 5.2, 2.1.
2. There are unexpected errors with a 154xC or 154x with both internal and external devices connected.

   This is probably a termination problem. In order to use the software option to disable host adapter termination, you must turn switch 1 off.

   See also Common Problems 1 and 3 and 5.2, 2.1.
3. The SCSI subsystem locks up completely.

   There are cases where the lockups seem to occur when multiple devices are in use at the same time. In this case, you can try contacting the manufacturer of the devices and see if firmware upgrades are available which would correct the problem. As a last resort, you can go into aha1542.h and change AHA1542_MAILBOX to 1. This will effectively limit you to one outstanding command on the scsi bus at one time, and may help the situation. If you have tape drives or slow cdrom drives on the bus, this might not be a practical solution.

   See also Common Problems 1 and 2 and 5.2, 2.8.
4. An "Interrupt received, but no mail" message is printed on bootup and your SCSI devices are not detected.

   Disable the BIOS options to support the extended mapping for disks > 1G, support for > 2 drives, and for autoscanning the bus. Or, upgrade to Linux .99.14k or newer.
5. If infinite timeout errors occur on 'C' revision boards, you may need to go into the Adaptec setup program and enable synchronous negotiation.
6. Linux 1.2.x gives the message

   Unable to determine Adaptec DMA priority. Disabling board.

   This is due to a conflict on some systems with the obsolete BusLogic driver. Either rebuild your kernel without it, or give the BusLogic driver a command line option telling it to look somewhere other than where your controller is configured. Ie, if you have an Adaptec board at port 0x334, and nothing at 0x330, use a command line option like

```
buslogic=0x330
```

7. The system locks up with simultaneous access to multiple devices on a 1542C or 1540C and disconnection enabled

   Some Adaptec firmware revisions have bugs. Upgrading to BIOS v2.11 purportedly fixes these problems.

5.5 Adaptec 174x

Supported Configurations :

```
Slots          : 1-8
Ports          : EISA board, not applicable
IRQs           : 9, 10, 11, 12, 14, 15
DMA Channels   : EISA board, not applicable
IO             : port mapped, bus master
```

Autoprobe :

```
works with all supported configurations
```

Autoprobe override :

```
none
```

Note:

1. This board has been discontinued by Adaptec.

Common Problems :

1. If the Adaptec 1740 driver prints the message "aha1740: Board detected, but EBCNTRL = %x, so disabled it." your board was disabled because it was not running in enhanced mode. Boards running in standard 1542 mode are not supported.

5.6 Adaptec 274x, 284x (Standard) 294x (ALPHA)

A newer version which also supports the Adaptec 294x boards is available at

```
ftp://ftp.ims.com/pub/Linux/aic7xxx
```

Supported Configurations :

```
                 274x          284x          294x
EISA Slots     : 1-12          N/A           N/A
Ports          : N/A           ALL           ALL
IRQs           : ALL           ALL           ALL
DMA Channels   : N/A           ALL           N/A

IO             : port mapped, bus master
```

Autoprobe Override :

kernel command line :

```
aha274x=extended
(to force extended mapping)
```

Notes:

1. BIOS MUST be enabled
2. The B channel on 2742AT boards is ignored.
3. CONFIG_PCI must be set if you are using a PCI board.

5.7 Always IN2000 (Standard)

Supported Configurations :

```
Ports          : 0x100, 0x110, 0x200, 0x220
IRQs           : 10, 11, 14, 15
DMA            : not used
IO             : port mapped
```

Autoprobe :

```
BIOS not required
```

Autoprobe override :

```
none
```

Common Problems :

1. There are known problems in systems with IDE drives and with swapping.

5.8 BusLogic MultiMaster Host Adapters

(this section Copyright 1995 by Leonard N. Zubkoff <lnz@dandelion.com>) (see README.BusLogic for more complete BusLogic driver documentation)

```
                  BusLogic MultiMaster SCSI Driver for Linux

                       Version 1.2.2 for Linux 1.2.13
                       Version 1.3.2 for Linux 1.3.88

                 ftp://ftp.dandelion.com/BusLogic-1.2.2.tar.gz
                 ftp://ftp.dandelion.com/BusLogic-1.3.2.tar.gz

                                16 April 1996

                               Leonard N. Zubkoff
                               Dandelion Digital
                               lnz@dandelion.com

BusLogic, Inc. designs and manufactures a variety of high performance SCSI host
adapters which share a common programming interface across a diverse collection
of bus architectures by virtue of their MultiMaster ASIC technology.  This
driver supports all present BusLogic MultiMaster Host Adapters, and should
support any future MultiMaster designs with little or no modification.  Host
adapters based on the new FlashPoint architecture are not supported by this
driver; consult the README.FlashPoint file for information about a program to
upgrade Linux users from the unsupported FlashPoint LT to the supported BT-948.

My primary goals in writing this completely new BusLogic driver for Linux are
to achieve the full performance that BusLogic SCSI Host Adapters and modern
SCSI peripherals are capable of, and to provide a highly robust driver that can
be depended upon for high performance mission critical applications.  All of
the major performance and error recovery features can be configured from the
Linux kernel command line, allowing individual installations to tune driver
performance and error recovery to their particular needs.

BusLogic has been an excellent company to work with and I highly recommend
their products to the Linux community.  In November 1995, I was offered the
opportunity to become a beta test site for their latest MultiMaster product,
the BT-948 PCI Ultra SCSI Host Adapter, and then again for the BT-958 PCI Wide
Ultra SCSI Host Adapter in January 1996.  This was mutually beneficial since
BusLogic received a degree and kind of testing that their own testing group
cannot readily achieve, and the Linux community has available high performance
host adapters that have been well tested with Linux even before being brought
```

to market. This relationship has also given me the opportunity to interact directly with their technical staff, to understand more about the internal workings of their products, and in turn to educate them about the needs and potential of the Linux community. Their interest and support is greatly appreciated.

Unlike some other vendors, if you contact BusLogic Technical Support with a problem and are running Linux, they will not tell you that your use of their products is unsupported. Their latest product marketing literature even states "BusLogic SCSI host adapters are compatible with all major operating systems including: ... Linux ...".

BusLogic, Inc. is located at 4151 Burton Drive, Santa Clara, California, 95054, USA and can be reached by Voice at 408/492-9090 or by FAX at 408/492-1542. BusLogic maintains a World Wide Web site at http://www.buslogic.com, an anonymous FTP site at ftp.buslogic.com, and a BBS at 408/492-1984. BusLogic Technical Support can be reached by electronic mail at techsup@buslogic.com, by Voice at 408/654-0760, or by FAX at 408/492-1542. Contact information for offices in Europe and Japan is available on the Web site.

SUPPORTED HOST ADAPTERS

The following list comprises the supported BusLogic SCSI Host Adapters as of the date of this document. It is recommended that anyone purchasing a BusLogic Host Adapter not in the following table contact the author beforehand to verify that it is or will be supported.

"W" Series Host Adapters:

```
BT-948      PCI     Ultra Fast Single-ended SCSI-2
BT-958      PCI     Ultra Wide Single-ended SCSI-2
BT-958D     PCI     Ultra Wide Differential SCSI-2
```

"C" Series Host Adapters:

```
BT-946C     PCI     Fast Single-ended SCSI-2
BT-956C     PCI     Fast Wide Single-ended SCSI-2
BT-956CD    PCI     Fast Wide Differential SCSI-2
BT-445C     VLB     Fast Single-ended SCSI-2
BT-747C     EISA    Fast Single-ended SCSI-2
BT-757C     EISA    Fast Wide Single-ended SCSI-2
BT-757CD    EISA    Fast Wide Differential SCSI-2
BT-545C     ISA     Fast Single-ended SCSI-2
BT-540CF    ISA     Fast Single-ended SCSI-2
```

```
"S" Series Host Adapters:

BT-445S     VLB     Fast Single-ended SCSI-2
BT-747S     EISA    Fast Single-ended SCSI-2
BT-747D     EISA    Fast Differential SCSI-2
BT-757S     EISA    Fast Wide Single-ended SCSI-2
BT-757D     EISA    Fast Wide Differential SCSI-2
BT-545S     ISA     Fast Single-ended SCSI-2
BT-542D     ISA     Fast Differential SCSI-2
BT-742A     EISA    Single-ended SCSI-2 (742A revision H)
BT-542B     ISA     Single-ended SCSI-2 (542B revision H)

"A" Series Host Adapters:

BT-742A     EISA    Single-ended SCSI-2 (742A revisions A - G)
BT-542B     ISA     Single-ended SCSI-2 (542B revisions A - G)

AMI FastDisk Host Adapters that are true BusLogic clones are supported by this
driver.

                      BT-948/958/958D INSTALLATION NOTES

The BT-948/958/958D PCI Ultra SCSI Host Adapters have some features which may
require attention in some circumstances when installing Linux.

o PCI I/O Port Assignments

  When configured to factory default settings, the BT-948/958/958D will only
  recognize the PCI I/O port assignments made by the motherboard's PCI BIOS.
  The BT-948/958/958D will not respond to any of the ISA compatible I/O ports
  that previous BusLogic SCSI Host Adapters respond to.  This driver supports
  the PCI I/O port assignments, so this is the preferred configuration.
  However, if the obsolete BusLogic driver must be used for any reason, such as
  a Linux distribution that does not yet use this driver in its boot kernel,
  BusLogic has provided an AutoSCSI configuration option to enable a legacy ISA
  compatible I/O port.

  To enable this backward compatibility option, invoke the AutoSCSI utility via
  Ctrl-B at system startup and select "Adapter Configuration", "View/Modify
  Configuration", and then change the "ISA Compatible Port" setting from
  "Disable" to "Primary" or "Alternate".  Once this driver has been installed,
  the "ISA Compatible Port" option should be set back to "Disable" to avoid
  possible future I/O port conflicts.  The older BT-946C/956C/956CD also have
```

this configuration option, but the factory default setting is "Primary".

o PCI Slot Scanning Order

In systems with multiple BusLogic PCI Host Adapters, the order in which the PCI slots are scanned may appear reversed with the BT-948/958/958D as compared to the BT-946C/956C/956CD. For booting from a SCSI disk to work correctly, it is necessary that the host adapter's BIOS and the kernel agree on which disk is the boot device, which requires that they recognize the PCI host adapters in the same order. The motherboard's PCI BIOS provides a standard way of enumerating the PCI host adapters, which is used by the Linux kernel. Some PCI BIOS implementations enumerate the PCI slots in order of increasing bus number and device number, while others do so in the opposite direction.

Unfortunately, Microsoft decided that Windows 95 would always enumerate the PCI slots in order of increasing bus number and device number regardless of the PCI BIOS enumeration, and requires that their scheme be supported by the host adapter's BIOS to receive Windows 95 certification. Therefore, the factory default settings of the BT-948/958/958D enumerate the host adapters by increasing bus number and device number. To disable this feature, invoke the AutoSCSI utility via Ctrl-B at system startup and select "Adapter Configuration", "View/Modify Configuration", press Ctrl-F10, and then change the "Use Bus And Device \# For PCI Scanning Seq." option to OFF.

This driver will interrogate the setting of the PCI Scanning Sequence option so as to recognize the host adapters in the same order as they are enumerated by the host adapter's BIOS.

BUSLOGIC ANNOUNCEMENTS MAILING LIST

The BusLogic Announcements Mailing List provides a forum for informing Linux users of new driver releases and other announcements regarding Linux support for BusLogic SCSI Host Adapters. To join the mailing list, send a message to "BusLogic-announce-request@dandelion.com" with the line "subscribe" in the message body.

5.9 BusLogic FlashPoint Host Adapters

(this section Copyright 1995 by Leonard N. Zubkoff <lnz@dandelion.com>)

There are no Linux drivers for the FlashPoint LT/DL/LW (BT-930/932/950) available and it is not clear when or if there will be any. The FlashPoint boards have a different architecture from the MultiMaster boards and have no onboard CPU, only a SCSI sequencer engine. They are positioned as a desktop workstation product, and are not particularly well suited for a high performance multitasking operating system like Linux.

The MultiMaster BT-948/958 have an onboard CPU and the mailbox programming interface allows for parallelism and pipelining between the host operating system and the host adapter, whereas the FlashPoint boards require frequent host CPU intervention. As interrupt latencies rise in a loaded multitasking system, the BT-948/958 should maintain excellent performance whereas the FlashPoint's performance will likely drop quite rapidly. Furthermore, the firmware on the BT-948/958 contains the low level knowledge for proper interaction with the SCSI bus, whereas with a sequencer engine the Linux driver must contain some or all of this information, and it often takes quite a long time to get all the kinks worked out. Given the relatively small difference in the street price of these products, the BT-948 or BT-958 is clearly the better choice for Linux.

{$<$}begin quotation{$>$}

ANNOUNCEMENT
BusLogic FlashPoint/BT-948 Upgrade Program
1 February 1996

Ever since its introduction last October, the BusLogic FlashPoint LT has been problematic for members of the Linux community, in that no Linux drivers have been available for this new Ultra SCSI product. Despite it's officially being positioned as a desktop workstation product, and not being particularly well suited for a high performance multitasking operating system like Linux, the FlashPoint LT has been touted by computer system vendors as the latest thing, and has been sold even on many of their high end systems, to the exclusion of the older MultiMaster products. This has caused grief for many people who inadvertently purchased a system expecting that all BusLogic SCSI Host Adapters were supported by Linux, only to discover that the FlashPoint was not supported and would not be for quite some time, if ever.

After this problem was identified, BusLogic contacted its major OEM customers to make sure the BT-946C/956C MultiMaster cards would still be made available, and that Linux users who mistakenly ordered systems with the FlashPoint would be able to upgrade to the BT-946C. While this helped many purchasers of new systems, it was only a partial solution to the overall problem of FlashPoint support for Linux users. It did nothing to assist the people who initially purchased a FlashPoint for a supported operating system and then later decided to run Linux, or those who had ended up with a FlashPoint LT, believing it was supported, and were unable to return it.

In the middle of December, I asked to meet with BusLogic's senior management to discuss the issues related to Linux and free software support for the FlashPoint. Rumors of varying accuracy had been circulating publicly about BusLogic's attitude toward the Linux community, and I felt it was best that these issues be addressed directly. I sent an email message after 11pm one evening, and the meeting took place the next afternoon. Unfortunately, corporate wheels sometimes grind slowly, especially when a company is being acquired, and so it's taken until now before the details were completely determined and a public statement could be made.

BusLogic is not prepared at this time to release the information necessary for third parties to write drivers for the FlashPoint. The only existing FlashPoint drivers have been written directly by BusLogic Engineering, and there is no FlashPoint documentation sufficiently detailed to allow outside developers to write a driver without substantial assistance. While there are people at BusLogic who would rather not release the details of the FlashPoint architecture at all, that debate has not yet been settled either way. In any event, even if documentation were available today it would take quite a while for a usable driver to be written, especially since I'm not convinced that the effort required would be worthwhile.

However, BusLogic does remain committed to providing a high performance SCSI solution for the Linux community, and does not want to see anyone left unable to run Linux because they have a Flashpoint LT. Therefore, BusLogic has put in place a direct upgrade program to allow any Linux user worldwide to trade in their FlashPoint LT for the new BT-948 MultiMaster PCI Ultra SCSI Host Adapter. The BT-948 is the Ultra SCSI successor to the BT-946C and has all the best features of both the BT-946C and FlashPoint LT, including smart termination and a flash PROM for easy firmware updates, and is of course compatible with the present Linux driver. The price for this upgrade has been set at US $45, and the upgrade program will be administered through BusLogic Technical Support, which can be reached by electronic mail at techsup@BusLogic.com, by Voice at +1 408 654-0760, or by FAX at +1 408 492-1542.

I was a beta test site for the BT-948 and versions 1.2.1 and 1.3.1 of my BusLogic driver already include latent support for the BT-948. Additional cosmetic support for the Ultra SCSI MultiMaster cards will be added in a subsequent release. As a result of this cooperative testing process, several firmware bugs were found and corrected (make sure you have firmware version 5.05R or later). My heavily loaded Linux test system provided an ideal environment for testing error recovery processes that are much more rarely exercised in production systems, but are crucial to overall system stability. It was especially convenient being able to work directly with their firmware engineer in demonstrating the problems under control of the firmware debugging environment; things sure have come a long way since the last time I worked on firmware for an embedded system. I am presently working on some performance testing and expect to have some data to report in the not too distant future.

BusLogic asked me to send this announcement since a large percentage of the questions regarding support for the FlashPoint have either been sent to me directly via email, or have appeared in the Linux newsgroups in which I participate. To summarize, BusLogic is offering Linux users an upgrade from the unsupported FlashPoint LT (BT-930) to the supported BT-948 for US $45. Contact BusLogic Technical Support at techsup@BusLogic.com or +1 408 654-0760 to take advantage of their offer.

Leonard N. Zubkoff
lnz@dandelion.com

{$<$}end quotation{$>$}

5.10 EATA: DPT SmartCache, SmartCache Plus, SmartCache III, SmartCache IV and SmartRAID (Standard)

Supported boards: all, that support the EATA-DMA protocol.

Among them are:

DPT Smartcache (Plus) family:

PM2011	ISA	Fast Single-ended SCSI-2
PM2012B	EISA	Fast Single-ended SCSI-2

DPT Smartcache III family:

PM2021	ISA	Fast Single-ended SCSI-2
PM2021W	ISA	Wide Single-ended SCSI-2
PM2022	EISA	Fast Single-ended SCSI-2
PM2022W	EISA	Wide Single-ended SCSI-2
PM2024	PCI	Fast Single-ended SCSI-2
PM2024W	PCI	Wide Single-ended SCSI-2
PM2122	EISA	Fast Single-ended SCSI-2
PM2122W	EISA	Wide Single-ended SCSI-2
PM2124	PCI	Fast Single-ended SCSI-2

```
PM2124W     PCI     Wide Single-ended SCSI-2
PM2322      EISA    Fast Single-ended SCSI-2
PM2322W     EISA    Wide Single-ended SCSI-2

DPT Smartcache VI family:
PM2041W     ISA     Wide Single-ended SCSI-2
PM2041UW    ISA     Ultra Wide Single-ended SCSI-2
PM2042W     EISA    Wide Single-ended SCSI-2
PM2042UW    EISA    Ultra Wide Single-ended SCSI-2
PM2044W     PCI     Wide Single-ended SCSI-2
PM2044UW    PCI     Ultra Wide Single-ended SCSI-2
PM2142W     EISA    Wide Single-ended SCSI-2
PM2142UW    EISA    Ultra Wide Single-ended SCSI-2
PM2144W     PCI     Wide Single-ended SCSI-2
PM2144UW    PCI     Ultra Wide Single-ended SCSI-2
PM2322W     EISA    Wide Single-ended SCSI-2
PM2322UW    EISA    Ultra Wide Single-ended SCSI-2

DPT SmartRAID family:
PM3021      ISA     Fast Single-ended SCSI-2
PM3021W     ISA     Wide Single-ended SCSI-2
PM3122      EISA    Fast Single-ended SCSI-2
PM3122W     EISA    Wide Single-ended SCSI-2
PM3222      EISA    Fast Single-ended SCSI-2
PM3222W     EISA    Wide Single-ended SCSI-2
PM3224      PCI     Fast Single-ended SCSI-2
PM3224W     PCI     Wide Single-ended SCSI-2
PM3334W     PCI     Wide Single-ended SCSI-2
PM3334UW    PCI     Ultra Wide Single-ended SCSI-2
```

also the differential versions of the above controllers.

and some controllers from:

NEC, AT&T, SNI, AST, Olivetti, Alphatronix.

Supported Configurations :

```
Slots          : ALL
Ports          : ALL
IRQs           : ALL level \& edge triggered
DMA Channels   : ISA ALL, EISA/PCI not applicable
IO             : port mapped, bus master
SCSI Channels  : ALL
```

Autoprobe :

```
works with all supported configurations
```

The latest version of the EATA-DMA driver is available on:

```
ftp.i-Connect.Net:/pub/Local/EATA/
```

Mailinglist: The EATA Mailing List provides a forum to Linux users of the EATA-DMA and EATA-PIO driver for discussions and announcements of new releases and other announcements. To join the mailing list, send a message to ”linux-eata-request@i-connect.net” with the line ”subscribe” in the message body.

/proc/scsi support: To get advanced command statistics, do the following: echo ”eata_dma latency” >/proc/scsi/eata_dma/<driver_no> and to switch it off again: echo ”eata_dma nolatency” >/proc/scsi/eata_dma/<driver_no>

Common Problems :

1. Slackware doesn't find the controller. Solution: Use one of the ascsi* bootdisks.
2. The IDE driver can detect the ST-506 interface of the EATA board in old kernels (<v1.3).
   (a) This will look like similar to one of the following 2 examples:

   ```
   hd.c: ST-506 interface disk with more than 16 heads detected,
     probably due to non-standard sector translation.  Giving up.
     (disk %d: cyl=%d, sect=63, head=64)
   ```

   ```
   hdc: probing with STATUS instead of ALTSTATUS
   hdc: MP0242 A, 0MB w/128KB Cache, CHS=0/0/0
   hdc: cannot handle disk with 0 physical heads
   hdd: probing with STATUS instead of ALTSTATUS
   hdd: MP0242 A, 0MB w/128KB Cache, CHS=0/0/0
   hdd: cannot handle disk with 0 physical heads
   ```

   If the IDE driver gets into trouble because of this, ie. you can't access your (real) IDE hardware, change the IO Port and/or the IRQ of the EATA board.

   (b) If the IDE driver finds hardware it can handle ie. harddisks with a capacity <=504MB, it will allocate the IO Port and IRQ, so that the eata driver can't utilize them. In this case also change IO Port and IRQ (!= 14,15).
3. Some old SK2011 boards have a broken firmware. Please contact DPT's customer support for an update.

Notes:

1. CONFIG_PCI must be set if you are using a PCI board.

5.11 Future Domain 16x0 with TMC-1800, TMC-18C30, TMC-18C50, or TMC-36C70 chip

Supported Configurations :

```
BIOSs          : 2.0, 3.0, 3.2, 3.4, 3.5
BIOS Addresses : 0xc8000, 0xca000, 0xce000, 0xde000
Ports          : 0x140, 0x150, 0x160, 0x170
IRQs           : 3, 5, 10, 11, 12, 14, 15
DMA            : not used
IO             : port mapped
```

Autoprobe :

```
works with all supported configurations, requires installed BIOS
```

Autoprobe Override :

```
none
```

Antiquity Problems, fix by upgrading :

1. Old versions do not support the TMC-18C50 chip, and will fail with newer boards.
2. Old versions will not have the most current BIOS signatures for autodetection.
3. Versions prior to the one included in Linux 1.0.9 and 1.1.6 don't support the new SCSI chip or 3.4 BIOS.

Notes :

1. The Future Domain BIOS often scans for SCSI-devices from highest ID to 0, in reverse order of other SCSI BIOSes. sda will be the last "drive letter" (ie, D: rather than C:). You may also need to use a a disktab override for LILO.

5.12 Generic NCR5380 / T130B (Standard)

Supported and Unsupported Configurations :

```
Ports          : all
IRQs           : all
DMA channels   : DMA is not used
IO             : port mapped
```

Autoprobe :

```
none
```

Autoprobe Override :

```
Compile time : Define GENERIC_NCR5380_OVERRIDE to be an array of tuples
with port, irq, dma, board type - ie
\#define GENERIC_NCR5380_OVERRIDE \{\{0x330, 5, DMA_NONE, BOARD_NCR5380\}\}

for a NCR5380 board at port 330, IRQ 5.

\#define GENERIC_NCR5380_OVERRIDE \{\{0x350, 5, DMA_NONE, BOARD_NCR53C400\}\}

for a T130B at port 0x350.

Older versions of the code eliminate the BOARD_* entry.

The symbolic IRQs IRQ_NONE and IRQ_AUTO may be used.
```

kernel command line :

```
ncr5380=port,irq
ncr5380=port,irq,dma
ncr53c400=port,irq

255 may be used for no irq, 254 for irq autoprobe.
```

Common Problems :

1. Using the T130B board with the old (pre public release 6) generic NCR5380 driver which doesn't support the ncr53c400 command line option. The NCR5380 compatible registers are offset eight from the base address. So, if your address is 0x350, use

   ```
   ncr5380=0x358,254
   ```

   on the kernel command line.

Antiquity problems, fix by upgrading :

1. The kernel locks up during disk access with T130B or other NCR53c400 boards. Pre-public release 6 versions of the Generic NCR5380 driver didn't support interrupts on these boards. Upgrade.

Notes :

1. the generic driver doesn't support DMA yet, and pseudo-DMA isn't supported in the generic driver.

5.13 NCR53c8xx (Standard)

Supported and Unsupported Configurations :

```
Base addresses : ALL
IRQs           : ALL
DMA channels   : PCI, not applicable
IO             : port mapped, busmastering
```

Autoprobe :

```
requires PCI BIOS, uses PCI BIOS routines to
search for devices and read configuration space
```

The driver uses the pre-programmed values in some registers for initialization, so a BIOS must be installed.

Antiquity Problems, fix by upgrading :

1. Older versions of Linux had a problem with swapping See6.2.7

2. Older versions of Linux didn't recognize '815 and '825 boards.

3. Distribution kernels include release 4 or 5 of the driver, which does not support useful things like disconnect/reconnect (the most noticeable effect of this being attempts to retension/rewind/file space a tape lock you out of all SCSI devices), multiple host adapters, and BIOSless operation. The latest release of the driver is available at

   ```
   ftp://tsx-11.mit.edu/pub/linux/ALPHA/scsi/ncr53c810
   ```

   Currently, this is a 1.2.10 and newer patch, although the next release will be 1.3.x exclusively. These patches are NOT entirely clean due to some ELF and other patches which were in the baseline revision of my source tree, and if you can't manually correct the (four) problems you should get, you shouldn't use them. Note that only the newest patch is needed; these are not incremental.

   If you wish to run the newer NCR driver with a 1.3.x kernel before then, Harald Evensen <Harald.Evensen@pvv.unit.no¿ has adapted the patches for 1.3.x

   ```
   ftp://ftp.pvv.unit.no/pub/Linux/ALPHA/ncr
   ```

   These patches should be clean.

   Please see all of the READMEs in these directories. You should also join the NCR mailing list if you are interested in running the ALPHA code, since interim bug fixes and announcements of the next release are posted to this list.

   To subscribe, send mail to majordomo@colorado.edu with

   ```
   subscribe ncr53c810
   ```

   in the text. You can unsubscribe by sending mail to the same address and including

```
unsubscribe ncr53c810
```

in the text.

Common Problems :

1. Many people have encountered problems where the chip worked fine under DOS, but failed under Linux with a timeout on test 1 due to a lost interrupt. This is often due to a mismatch between the IRQ hardware jumper for a slot or mainboard device and the value set in the CMOS setup. DOUBLE CHECK

   - The IRQ you are using is used only by your onboard NCR chip, or the slot an NCR board is installed in
   - Any main board jumpers selecting the IRQ for the onboard chip or slot match your CMOS setup.a
   - Some PCI mainboards have an "auto" assignment feature, which will not work.

   It may also be due to PCI INTB, INTC, or INTD being selected on a PCI board in a system which only supports PCI INTA. If you are using an NCR board which has jumpers to select between PCI interrupt lines, make sure you are using INTA.

   Finally, PCI should be using level-sensitive rather than edge triggered interrupts. Check that your board is jumpered for level-sensitive, and if that fails try edge-triggered because your system may be broken.

   This problem is especially common with Viglen some Viglen motherboards, where the mainboard IRQ jumper settings are NOT as documented in the manual. I've been told that what claims to be IRQ5 is really IRQ9, your mileage will vary.

2. Lockups / other problems occur when using an S3 928, or Tseng ET4000W32 PCI video board.

   There are hardware bugs in at least some revisions of these chips. Don't use them.

3. You get a message on boot up indicating that the I/O mapping was disabled because base address 0 bits 0..1 indicated a non I/O mapping

   This is due to a BIOS bug in some machines which results in dword reads of configuration registers returning the high and low 16 bit words swapped.

4. Some systems have problems if PCI write posting, or CPU-¿ PCI buffering are enabled. If you have problems, disable these options.

5. Some systems with the NCR SDMS software in an onboard BIOS

   ROM and in the system BIOS are unable to boot DOS. Disabling the image in one place should rectify this problem.

6. If you encounter the message

   ```
   "scsi%d: IRQ0 not free, detaching"
   ```

or

```
"scsi%d: IRQ255 not free, detaching"
```

The NCR chip had all 0 or 1 bits stored in the PCI configuration register. Either you have configuration problems (see 1), or you have a defective mainboard BIOS.

As a work around, you could edit drivers/scsi/ncr53c7,8xx.c, and change pci_init() so that you have

```
irq = my_irq;
```

before

```
return normal_init (tpnt, board, chip, (int) base,
    (int) io_port, (int) irq, DMA_NONE, 1, bus, device_fn,
    options);
```

7. Some systems have hideous, broken, BIOS chips. Don't make any bug reports until you've made sure you have the newest ROM from your vendor.

8. The command line overrides ncr53c810=xxx, etc. don't work.

   In stock kernels, this is because their entry points are not included in init/main.c, which is quite intentional :

   The driver makes no attempt to avoid autoprobing for a board where a command line override was used, so if an override is used where the board actually showed up to the PCI configuration routines, you'll have big problems.

   The only reason you would need an override would be if the PCI hardware + BIOS were broken, in which case certain error recovery routines wouldn't work, rendering the override less than useful.

   Finally, nearly all of people who _think_ they need a command line override do because they get configuration or other error messages from the driver. If the driver says you have a configuration problem, you have a broken system or a configuration problem and no override is going to fix this.

   If some one has gone and added the appropriate entry points to init/main.c for command line overrides, they are totally unsupported and may not work.

9. Certain NCR boards (most notably Nexstor) which don't use an NCR BIOS get timeouts. Some of these ROMs handle synchronous and transfers, negotiate for sync. transfers on power up, and leave the drives in an unknown state. When the distribution Linux NCR driver attempts to talk with them, it gets timeouts and cannot recover because it won't do a bus reset or renegotiate.

   If you run into this problem, you can either disable synchronous transfers in the board's setup program, or upgrade to a newer ALPHA release of the NCR driver which will do synchronous negotiation.

10. Tyan S1365 '825 boards have problems with timeouts, especially when disconnects are enabled. Some of these boards have the documentation regarding the termination enable jumper reversed - so that termination is off when you need it, and on when it shouldn't be.

    Try reversing the position of the jumper.

Notes:

1. CONFIG_PCI must be set

5.14 Seagate ST0x/Future Domain TMC-8xx/TMC-9xx (Standard)

Supported and Unsupported Configurations :

```
Base addresses : 0xc8000, 0xca000, 0xcc000, 0xce000, 0xdc000, 0xde000
IRQs           : 3, 5
DMA channels   : DMA is not used
IO             : memory mapped
```

Autoprobe :

```
probes for address only, IRQ is assumed to be 5, requires installed BIOS.
```

Autoprobe Override :

Compile time :

```
Define OVERRIDE to be the base address, CONTROLLER to
FD or SEAGATE as appropriate, and IRQ to the IRQ.
```

kernel command line :

```
st0x=address,irq or tmc8xx=address,irq (only works for .99.13b and newer)
```

Antiquity Problems, fix by upgrading :

1. Versions prior to the one in the Linux .99.12 kernel had a problem handshaking with some slow devices, where This is what happens when you write data out to the bus

   (a) Write byte to data register, data register is asserted to bus

   (b) time_remaining = 12us

   (c) wait while time_remaining ¿ 0 and REQ is not asserted

   (d) if time_remaining ¿ 0, assert ACK

   (e) wait while time remaining ¿ 0 and REQ is asserted

   (f) deassert ACK

   The problem was encountered in slow devices that do the command processing as they read the command, where the REQ/ACK handshake takes over 12us - REQ didn't go false when the driver expected it to, so the driver ended up sending multiple bytes of data for each REQ pulse.

2. With Linux .99.12, a bug was introduced when I fixed the arbitration code, resulting in failed selections on some systems. This was fixed in .99.13.

Common Problems :

1. There are command timeouts when Linux attempts to read the partition table or do other disk access. The board ships with the defaults set up for MSDOS, ie interrupts are disabled. To jumper the board for interrupts, on the Seagate use jumper W3 (ST01) or JP3 (ST02) and short pins F-G to select IRQ 5.

2. The driver can't handle some devices, particularly cheap SCSI tapes and CDROMs.

   The Seagate ties the SCSI bus REQ/ACK handshaking into the PC bus IO CHANNEL READY and (optionally) 0WS signals. Unfortunately, it doesn't tell you when the watchdog timer runs out, and you have no way of knowing for certain that REQ went low, and may end up seeing one REQ pulse as multiple REQ pulses.

   Dealing with this means using a tight loop to look for REQ to go low, with a timeout incase you don't catch the transition due to an interrupt, etc. This results in a performance decrease, so it would be undesirable to apply this to all SCSI devices. Instead, it is selected on a per-device basis with the "broken" field for the given SCSI device in the scsi_devices array. If you run into problems, you should try adding your device to the list of devices for which broken is not reset to zero (currently, only the TENEX CDROM drives).

3. A future domain board (specific examples include the 840, 841, 880, and 881) doesn't work.

   A few of the Future domain boards use the Seagate register mapping, and have the MSG and CD bits of the status register flipped.

   You should edit seagate.h, swapping the definitions for STAT_MSG and STAT_CD, and recompile the kernel with CONTROLLER defined to SEAGATE and an appropriate IRQ and OVERRIDE specified.

4. When attempting to fdisk your drive, you get error messages indicating that the HDIO_REQ or HDIO_GETGEO ioctl failed, or

   ```
   You must set heads sectors and cylinders.
   You can do this from the extra functions menu.
   ```

   See 6.4

5. After manually specifying the drive geometry, subsequent attempts to read the partition table result in partition boundary not on a cylinder boundary, physical and logical boundaries don't match, etc. error messages.

   See 6.4

6. Some systems which worked prior to .99.13 fail with newer versions of Linux. Older versions of Linux assigned the CONTROL and DATA registers in an order different than that outlined in the Seagate documentation, which broke on some systems. Newer versions make the assignment in the correct way, but this breaks other systems.

The code in seagate.c looks like this now :

```
cli();
DATA = (unsigned char) ((1 {$<$}{$<$} target) | (controller_type == SEAGATE ? 0x80 : 0x40));
CONTROL = BASE_CMD | CMD_DRVR_ENABLE | CMD_SEL |
            (reselect ? CMD_ATTN : 0);
sti();
```

Changing this to

```
cli();
CONTROL = BASE_CMD | CMD_DRVR_ENABLE | CMD_SEL |
            (reselect ? CMD_ATTN : 0);
DATA = (unsigned char) ((1 {$<$}{$<$} target) | (controller_type == SEAGATE ? 0x80 : 0x40));
sti();
```

may fix your problem.

Defines :

```
FAST or FAST32 will use blind transfers where possible

ARBITRATE will cause the host adapter to arbitrate for the
        bus for better SCSI-II compatibility, rather than just
        waiting for BUS FREE and then doing its thing.  Should
        let us do one command per Lun when I integrate my
        reorganization changes into the distribution sources.

SLOW_HANDSHAKE will allow compatibility with broken devices that don't
        handshake fast enough (ie, some CD ROM's) for the Seagate
        code.

SLOW_RATE=x, x some number will let you specify a default
        transfer rate if handshaking isn't working correctly.
```

5.15 PAS16 SCSI (Standard)

Supported and Unsupported Configurations :

```
Ports          : 0x388, 0x384, 0x38x, 0x288
IRQs           : 10, 12, 14, 15
     IMPORTANT : IRQ MUST be different from the IRQ used for the sound
                 portion of the board.
DMA            : is not used for the SCSI portion of the board
IO             : port mapped
```

Autoprobe :

```
does not require BIOS
```

Autoprobe Override :

```
Compile time : Define PAS16_OVERRIDE to be an array of port, irq
tuples.  Ie

\#define PAS16_OVERRIDE \{\{0x388, 10\}\}

for a board at port 0x388, IRQ 10.
```

kernel command line :

```
pas16=port,irq
```

Defines :

```
AUTOSENSE - if defined, REQUEST SENSE will be performed automatically
for commands that return with a CHECK CONDITION status.

PSEUDO_DMA - enables PSEUDO-DMA hardware, should give a 3-4X performance
increase compared to polled I/O.

PARITY - enable parity checking.  Not supported

SCSI2 - enable support for SCSI-II tagged queuing.  Untested


UNSAFE - leave interrupts enabled during pseudo-DMA transfers.  You
         only really want to use this if you're having a problem with
         dropped characters during high speed communications, and even
         then, you're going to be better off twiddling with transfersize.

USLEEP - enable support for devices that don't disconnect.  Untested.
```

Common problems :

1. Command timeouts, aborts, etc. You should install the NCR5380 patches that I posted to the net some time ago, which should be integrated into some future alpha release. These patches fix a race condition in earlier NCR5380 driver cores, as well as fixing support for multiple devices on NCR5380 based boards.

   If that fails, you should disable the PSEUDO_DMA option by changing the #define PSEUDO_DMA line in drivers/scsi/pas16.c to #undef PSEUDO_DMA.

   Note that the later should be considered a last resort, because there will be a severe performance degradation.

5.16 Trantor T128/T128F/T228 (Standard)

Supported and Unsupported Configurations :

```
Base addresses :  0xcc000, 00xc8000, 0xdc000, 0xd8000
IRQs           : none, 3, 5, 7 (all boards)
                 10, 12, 14, 15 (T128F only)
DMA            : not used.
IO             : memory mapped
```

Autoprobe :

```
works for all supported configurations, requires installed BIOS.
```

Autoprobe Override :

```
Compile time : Define T128_OVERRIDE to be an array of address, irq
tuples.  Ie

\#define T128_OVERRIDE \{\{0xcc000, 5\}\}

for a board at address 0xcc000, IRQ 5.

The symbolic IRQs IRQ_NONE and IRQ_AUTO may be used.
```

kernel command line :

```
t128=address,irq
-1 may be used for no irq, -2 for irq autoprobe.
```

Defines :

```
AUTOSENSE - if defined, REQUEST SENSE will be performed automatically
for commands that return with a CHECK CONDITION status.

PSEUDO_DMA - enables PSEUDO-DMA hardware, should give a 3-4X performance
increase compared to polled I/O.

PARITY - enable parity checking.  Not supported

SCSI2 - enable support for SCSI-II tagged queuing.  Untested


UNSAFE - leave interrupts enabled during pseudo-DMA transfers.  You
         only really want to use this if you're having a problem with
         dropped characters during high speed communications, and even
         then, you're going to be better off twiddling with transfersize.

USLEEP - enable support for devices that don't disconnect.  Untested.
```

Common Problems :

1. Command timeouts, aborts, etc.

   You should install the NCR5380 patches that I posted to the net some time ago, which should be integrated into some future alpha release. These patches fix a race condition in earlier NCR5380 driver cores, as well as fixing support for multiple devices on NCR5380 based boards. If that fails, you should disable the PSEUDO_DMA option by changing the #define PSEUDO_DMA line in drivers/scsi/pas16.c to #undef PSEUDO_DMA.

   Note that the later should be considered a last resort, because there will be a severe performance degradation.

5.17 Ultrastor 14f (ISA), 24f (EISA), 34f (VLB) (Standard)

Supported Configurations :

```
Ports         : 0x130, 0x140, 0x210, 0x230, 0x240, 0x310, 0x330, 0x340
IRQs          : 10, 11, 14, 15
DMA channels  : 5, 6, 7
IO            : port mapped, bus master
```

Autoprobe :

```
does not work for boards at port 0x310, BIOS not required.
```

Autoprobe override :

```
compile time only, define PORT_OVERRIDE
```

Common Problems :

1. The address 0x310 is not supported by the autoprobe code, and may cause conflicts if networking is enabled.

   Please use a different address.

2. Using an Ultrastor at address 0x330 may cause the system to hang when the sound drivers are autoprobing.

   Please use a different address.

3. Various other drivers do unsafe probes at various addresses, if you are having problems with detection or the system is hanging at boot time, please try a different address.

   0x340 is recommended as an address that is known to work.

4. Linux detects no SCSI devices, but detects your SCSI hard disk on an Ultrastor SCSI board as a normal hard disk, and the hard disk driver refuses to support it. Note that when this occurs, you will probably also get a message

   hd.c: ST-506 interface disk with more than 16 heads detected, probably due to non-standard sector translation. Giving up. (disk %d: cyl=%d, sect=63, head=64)

   If this is the case, you are running the Ultrastor board in WD1003 emulation mode. You have

   (a) Switch the Ultrastor into native mode. This is the recommended action, since the SCSI driver can be significantly faster than the IDE driver, especially with the clustered read/write patches installed. Some users have sustained in excess of 2M/sec through the file system using these patches.

       Note that this will be necessary if you wish to use any non- hard disk, or more than two hard disk devices on the Ultrastor.

   (b) Use the kernel command line switch

       ```
       hd=cylinders,heads,sectors
       ```

       to override the default setting to bootstrap yourself, keeping number of cylinders <= 2048, number of heads <= 16, and number of sectors <= 255 such that cylinders * heads * sectors is the same for both mappings.

       You'll also have to manually specify the disk geometry when running fdisk under Linux. Failure to do so will result in incorrect partition entries being written, which will work correctly with Linux but fail under MSDOS which looks at the cylinder/head/sector entries in the table.

       Once Linux is up, you can avoid the inconvenience of having to boot by hand by recompiling the kernel with an appropriately defined HD_TYPE macro in include/linux/config.h.

5.18 Western Digital 7000 (Standard)

Supported Configurations :

```
BIOS Addresses : 0xce000
Ports          : 0x350
IRQs           : 15
DMA Channels   : 6
IO             : port mapped, bus master
```

Autoprobe :

```
requires installed BIOS
```

Common Problems :

1. There are several revisions of the chip and firmware. Supposedly, revision 3 boards do not work, revision 5 boards do, chips with no suffix do not work, chips with an 'A' suffix do.
2. The board supports a few BIOS addresses which aren't on the list of supported addresses. If you run into this situation, please use one of the supported addresses and submit a bug report as outlined in 3.

5.19 AM53/79C974 (ALPHA)

```
ftp://tsx-11.mit.edu/pub/linux/ALPHA/scsi/AM53C974-0.3.tar.gz
```

Supported Configurations :

```
Ports          : all
IRQs           : all
DMA Channels   : 6
IO             : port mapped, bus master (unintelligent)
```

5.20 qlogic (Standard)

Hey Drew, where is this section (I (D.F.) saw it only in the toc ;-)?

6 Disks

This section gives information that is specific to disk drives.

6.1 Supported and Unsupported Hardware

All direct access SCSI devices with a block size of 256, 512, or 1024 bytes should work. Other block sizes will not work (Note that this can often be fixed by changing the block and/or sector sizes using the MODE SELECT SCSI command)

Sector size refers to the number of data bytes allocated per sector on a device, ie CDROMs use a 2048 byte sector size.

Block size refers to the size of the logical blocks used to interface with the device. Although this is usually identical to sector size, some devices map multiple smaller physical sectors (ie, 256 bytes in the case of 55M Syquest drives) to larger logical blocks or vice versa (ie, 512 byte blocks on SUN compatible CDROM drives).

Removable media devices, including Bernoulis, flopticals, MO drives, and Syquests.

In theory, drives up to a terabyte in size should work. There is definitely no problem with tiny 9G drives.

6.2 Common Problems

6.2.1 Cylinder > 1024 message

When partitioning, you get a warning message about "cylinder > 1024" or you are unable to boot from a partition including a logical cylinder past logical cylinder 1024.

This is a BIOS limitation.

See Disk 6.5 and 6.4 for an explanation.

6.2.2 You are unable to partition "/dev/hd*"

/dev/hd* aren't SCSI devices, /dev/sd* are.

See 6.3 and Disk 6.5 and 6.4 for the correct device names and partitioning procedure.

6.2.3 Unable to eject media from a removable media drive

Linux attempts to lock the drive door when a piece of media is mounted to prevent filesystem corruption due to an inadvertent media change.

Please unmount your disks before ejecting them.

6.2.4 Unable to boot using LILO from a SCSI disk

In some cases, the SCSI driver and BIOS will disagree over the correct BIOS mapping to use, and will result in LILO hanging after 'LI' at boot time and/or other problems.

To workaround this, you'll have to determine your BIOS geometry mapping used under DOS, and make an entry for your disk in /etc/lilo/disktab.

Alternatively, you may be able to use the "linear" configuration jfile option.

6.2.5 Fdisk responds with

```
You must set heads sectors and cylinders.
You can do this from the extra functions menu.
```

and disk geometry is not 'remembered' when fdisk is rerun.

See 6.4

6.2.6 Only one drive is detected on a bridge board with multiple drives connected.

Linux won't search LUNs past zero on SCSI devices which predate ANSI SCSI revision 1. If you wish devices on alternate LUNs to be recognized, you will have to modify drivers/scsi/scsi.c:scan_scsis().

6.2.7 System hangs when swapping

We think this has been fixed, try upgrading to 1.1.38.

6.2.8 Connor CFP1060S disks get corrupted

This is due to a microcode bug in the read-ahead and caching code.

¿From Soenke Behrens of Conner tech. support :

```
During the past few weeks, we got several calls from customers stating
that they had severe problems with Conner CFP1060x 1GB SCSI drives
using the Linux operating system. Symptoms were corrupt filesystems
(damaged inodes) reported by e2fsck on each system boot and similar
errors.

There is now a fix available for customers with a CFP1060x (microcode
revisions 9WA1.62/1.66/1.68) and Linux. To apply the upgrade, you
will need a DOS boot disk and ASPI drivers that can access the hard
drive. The upgrade downloads new queuing and lookahead code into the
non-volatile SCSI RAM of the drive.

If you are experiencing problems with a disk that has microcode
revision 9WA1.60, you will have to contact your nearest Conner service
centre to get the disk upgraded. The microcode revision can be found
on the label of the drive and on the underside of the drive on a label
on one of the ICs.

If you are confident that you can perform the upgrade yourself, please
contact Conner Technical Support and have your microcode revision
ready. Conner Technical Support Europe can be reached on +44-1294-315333,
Conner Technical Support in the USA can be reached on 1-800-4CONNER.

Regards
Soenke Behrens
European Technical Support
```

6.3 Device Files

SCSI disks use block device major 8, and there are no "raw" devices ala BSD.

16 minor numbers are allocated to each SCSI disk, with minor % 16 == 0 being the whole disk, minors 1 <= (minor % 16) <= 4 the four primary partitions, minors 5 <= (minor % 16) <= 15 any extended partitions.

Ie, a configuration may work out like this (with one host adapter)

```
Device                 Target, Lun     SCSI disk
84M Seagate            0       0       /dev/sda
SCSI->SMD bridge disk 0 3       0       /dev/sdb
SCSI->SMD bridge disk 1 3       1       /dev/sdc
Wangtek tape           4       0       none
213M Maxtor            6       0       /dev/sdd
```

Etc.

The standard naming convention is

/dev/sd{letter} for the entire disk device ((minor % 16) == 0) /dev/sd{letter}{partition} for the partitions on that device (1 <= (minor % 16) <= 15)

Ie

```
/dev/sda        block device major 8 minor 0
/dev/sda1       block device major 8 minor 1
/dev/sda2       block device major 8 minor 2
/dev/sdb        block device major 8 minor 16
```

etc.

6.4 Partitioning

You can partition your SCSI disks using the partitioning program of your choice, under DOS, OS/2, Linux or any other operating system supporting the standard partitioning scheme.

The correct way to run the Linux fdisk program is by specifying the device on the command line. Ie, to partition the first SCSI disk,

```
fdisk /dev/sda
```

If you don't explicitly specify the device, the partitioning program may default to /dev/hda, which isn't a SCSI disk.

In some cases, fdisk will respond with

```
You must set heads sectors and cylinders.
You can do this from the extra functions menu.

Command (m for help):
```

and/or give a message to the effect that the HDIO_REQ or HDIO_GETGEO ioctl failed. In these cases, you must manually specify the disk geometry as outlined in 6.5 when running fdisk, and also in /etc/disktab if you wish to boot kernels off that disk with LILO.

If you have manually specified the disk geometry, subsequent attempts to run fdisk will give the same error message. This is normal, since PCs don't store the disk geometry information in the partition table. In and of itself, will cause _NO PROBLEMS_, and you will have no problems accessing partitions you created on the drive with Linux. Some vendors' poor installation code will choke on this, in which case you should contact your vendor and insist that they fix the code.

In some cases, you will get a warning message about a partition ending past cylinder 1024. If you create one of these partitions, you will be unable to boot Linux kernels off of that partition using LILO. Note, however, that this restriction does not preclude the creation of a root partition partially or entirely above the 1024 cylinder mark, since it is possible to create a small /boot partition below the 1024 cylinder mark or to boot kernels off existing partitions.

6.5 Disk Geometry

Under Linux, each disk is viewed as the SCSI host adapter sees it : N blocks, numbered from 0 to N-1, all error free, where as DOS/BIOS predate intelligent disks and apply an arbitrary head / cylinder / sector mapping to this linear addressing.

This can pose a problem when you partition the drives under Linux, since there is no portable way to get DOS/BIOS's idea of the mapped geometry. In most cases, a HDIO_GETGEO ioctl() can be implemented to return this mapping. Unfortunately, when the vendor (ie Seagate) has chosen a perverse, non-standard, and undocumented mapping, this is not possible and geometry must be manually specified

If manual specification of the is required, you have one of several options :

1. If you don't care about using DOS, or booting kernels from the drive with LILO, create a translation such that heads * cylinders * sectors * 512 < size of your drive in bytes (a megabyte is defined as 2^20 bytes).

   ```
   1 <= heads <= 256
   1 <= cylinders <= 1024
   1 <= sectors <= 63
   ```

2. Use the BIOS mapping. In some cases, this will mean reconfiguring the disk so that it is at SCSI ID 0, and disabling the second IDE drive (if you have one).

You can either use a program like NU, or you can use the following program :

```
begin 664 dparam.com
MBAZ``##_B+^!`+N!`(H'0SP@=/D\,'5:@#]X=`6`/UAU4(!_`3AU2H!_`P!U
M1(I7`H#J,(#Z`7{$<$}Y@,*`M`C-$PCD=3-14HC()#\PY.@R`.@J`%J(\/[`,.3H
M)0#H'0!8AL2Q!M+LO.@7`+K"`;0)S2'#NIP!ZR"ZQO'K\&[K5`>L6N]T!,=*Y
M"@#W\8#",$N(%PG`=>^)VK0)S2'#=7-A9V4Z(\&1P87)A;2`P>#@P#0H@("!0
L{$<$}B`@9'!A{$<$}F%M(#!X.#$-"B1);G9A;\&ED(\&1R:79E#0HD("`D```````D``!0
`
end
```

When run it prints the sectors, heads, and cylinders of the drive whose BIOS address was specified on the command line (0x80 is the first disk, 0x81 the second).

Ie, dparam 0x80

```
60      17      1007
```

Would mean that C: had 60 sectors, 17 heads, and 1007 cylinders.

7 CD ROMs

This section gives information that is specific to cdrom drives.

7.1 Supported and Unsupported Hardware

SCSI CDs with a block size of 512 or 2048 bytes should work. Other block sizes will not work.

7.2 Common Problems

7.2.1 Unable to mount cdrom

The correct syntax to mount an ISO-9660 CDROM is

```
mount -t iso9660 /dev/sr0 /mount_point -o ro
```

Note that for this to work, you must have the kernel configured with support for SCSI, your host adapter, the SCSI CDROM driver, and the iso9660 filesystem.

Note that as of Linux 1.1.32, read-only devices such as CDROMs CANNOT be mounted with the default read/write options.

7.2.2 Unable to eject cdrom

Linux attempts to lock the drive door when a piece of media is mounted to prevent filesystem corruption due to an inadvertent media change.

7.2.3 Unable to play audio

The programs Workman or xcdplayer will do this for you.

7.2.4 Workman or Xcdplayer do not work

The functions to control audio functions are part of the SCSI-II command set, so any drive that is not SCSI-II will probably not work here. Also, many SCSI-I and some SCSI-II CDROM drives use a proprietary command set for accessing audio functions instead of the SCSI-II command set. For NEC drives, there is a version of xcdplayer specially adapted to use this command set floating around - try looking on tsx-11.mit.edu in pub/linux/BETA/cdrom.

These programs may work with some of the non-SCSI cdrom drives if the driver implements the same ioctls as the scsi drivers.

7.2.5 Additional drives on CD ROM changers do not work

Most CD changers assign each disc to a logical unit. Insure that you have special files made for each platter (see 6.3) and see 2.10.

7.3 Device Files

SCSI CD ROMs use major 11.

Minors are allocated dynamically (See 6, 6.3 for an example) with the first CDROM found being minor 0, the second minor 1, etc.

The standard naming convention is

/dev/sr{digit}, although some distributions have used /dev/scd{digit}, with examples being

```
/dev/sr0        /dev/scd0
/dev/sr1        /dev/scd1
```

8 Tapes

This section gives information that is specific to scsi tape drives.

8.1 Supported and Unsupported Hardware

Drives using both fixed and variable length blocks smaller than the the driver buffer length (set to 32K in the distribution sources) are supported.

Parameters (block size, buffering, density) are set with ioctls (usually with the mt program), and remain in effect after the device is closed and reopened.

Virtually all drives should work, including :

- Archive Viper QIC drives, including the 150M and 525M models
- Exabyte 8mm drives
- Wangtek 5150S drives
- Wangdat DAT drives

8.2 Common Problems

8.2.1 Tape drive not recognized at boot time

Try booting with a tape in the drive.

8.2.2 Tapes with multiple files cannot be read properly

When reading a tape with multiple files, the first tar is successful, a second tar fails silently, and retrying the second tar is successful.

User level programs, such as tar, don't understand file marks. The first tar reads up until the end of the file. The second tar attempts to read at the file mark, gets nothing, but the tape spaces over the file mark. The third tar is successful since the tape is at the start of the next file.

Use mt on the no-rewind device to space forward to the next file.

8.2.3 Decompression fails

Decompressing programs cannot handle the zeros padding the last block of the file.

To prevent warnings and errors, wrap your compressed files in a .tar file - ie, rather than doing

```
tar cfvz /dev/nrst0 file.1 file.2 ...
```

do

```
tar cfvz tmp.tar.z file.1 file.2 ...
```

```
tar cf /dev/nrst0 tmp.tar.z
```

8.2.4 Problems taking tapes to/from other systems

You can't read a tape made with another operating system or another operating system can't read a tape written in Linux.

Different systems often use different block sizes. On a tape device using a fixed blocksize, you will get errors when reading blocks written using a different block size.

To read these tapes, you must set the blocksize of the tape driver to match the blocksize used when the tape was written, or to variable.

NOTE : this is the hardware block size, not the blocking factor used with tar, dump, etc.

You can do this with the mt command -

```
mt setblk {$<$}size>
```

or

```
mt setblk 0
```

to get variable block length support.

Note that these mt flags are NOT supported under the GNU version of mt which is included with some Linux distributions. Instead, you must use the BSD derived Linux SCSI mt command. Source should be available from

```
tsx-11.mit.edu:/pub/linux/ALPHA/scsi
```

Also note that by default, ST_BUFFER_BLOCKS (defined in /usr/src/linux/drivers/scsi/st_options.h in newer kernels, st.c in older kernels) is set to allow for a 32K maximum buffer size; you'll need to edit the source to use larger blocks.

8.2.5 "No such device" error message

All attempts to access the tape result in a

"No such device"

or similar error message. Check the type of your tape device - it MUST be a character device with major and minor numbers matching those specified in 6.3.

8.2.6 Tape reads at a given density work, writes fail

Many tape drives support reading at lower densities for compatibility with older hardware, but will not write at those same densities.

This is especially the case with QIC drives, which will read old 60M tapes but only write new 120, 150, 250, and 525M formats.

8.2.7 Repositioning the tape locks out access to all SCSI devices

This is most common with SCSI drivers which only support one outstanding command at a time (see 10.5 for an explanation, and 10.7 to see which drivers suffer from this limitation), although there may be a few tape drives out there which refuse to disconnect.

In either case, you can work around the problem by editing drivers/scsi/st.c and adding a

```
#define ST_NOWAIT
```

at the top and rebuilding the kernel.

Note that this will defer error condition reporting until the next SCSI command is executed. For this reason, you may want to do something like a

```
mt status
```

after a mt file positioning command so you don't overwrite tape files if the positioning command failed.

You may also wish to consider changing to a better-supported SCSI board or newer tape drive if you need to use this workaround and are writing multiple files to tapes.

8.3 Device Files

SCSI tapes use character device major 9.

Due to constraints imposed by Linux's use of a sixteen bit dev_t with only eight bits allocated to the minor number, the SCSI tape minor numbers are assigned dynamically starting with the lowest SCSI HOST/ID/LUN.

Rewinding devices are numbered from 0 - with the first SCSI tape, /dev/rst0 being c 9 0, the second /dev/rst1 c 9 1, etc. Non-rewinding devices have the high bit set in the minor number, ie /dev/nrst0 is c 9 128.

The standard naming convention is

```
/dev/nst\{digit\}       for non-rewinding devices
/dev/st\{digit\}               for rewinding devices
```

9 Generic

This information gives information that is specific to the generic scsi driver.

9.1 Supported Hardware

The Generic SCSI device driver provides an interface for sending SCSI commands to all SCSI devices - disks, tapes, CDROMs, media changer robots, etc.

Everything electrically compatible with your SCSI board should work.

9.2 Common Problems

None :-).

9.3 Device Files

SCSI generic devices use character major 21. Due to constraints imposed by Linux's use of a 16 bit dev_t, minor numbers are dynamically assigned from 0, one per device, with

```
/dev/sg0
```

corresponding to the lowest numerical target/lun on the first SCSI board.

10 Buyers' Guide

A frequent question is:

"Linux supports quite a number of different boards, so which scsi host adapter should I get."

The answer depends upon how much performance you expect or need, motherboard, and the scsi peripherals that you plan on attaching to your machine.

10.1 Transfer types

The biggest factor affecting performance (in terms of throughput and interactive response time during SCSI I/O) is the transfer type used. The table below lists the various transfer types, the effects they have on performance, and some recommendations as to their use.

Transfer type

Description / Performance / Recommendations

Pure Polled

A pure polled I/O board will use the CPU to handle all of the SCSI processing, including the REQ/ACK handshaking.

Even a fast CPU will be slower handling the REQ/ACK handshake sequence than a simple finite state machine, resulting in peak transfer rates of about 150K/sec on a fast machine, perhaps 60K/sec on a slow machine (through the filesystem).

The driver also must sit in a tight loop as long as the SCSI bus is busy, resulting in near 100% CPU utilization and extremely poor responsiveness during SCSI I/O. Slow CDROMs which don't disconnect/reconnect will kill interactive performance with these boards.

Not recommended.

Interlocked Polled

Boards using interlocked polled I/O are essentially the same as pure polled I/O boards, only the SCSI REQ/ACK handshaking signals are interlocked with the PC bus handshaking signals. All SCSI processing beyond the handshaking is handled by the CPU.

Peak transfer rates of 500-600K/sec through the filesystem are possible on these boards.

As with pure polled I/O boards, the driver must sit in a tight loop as long as the SCSI bus is busy, resulting in CPU utilization dependent on the transfer rates of the devices, and when they disconnect/reconnect. CPU utilization may vary between 25% for single speed CDs which handle disconnect/reconnect properly to 100% for faster drives or broken CD ROMs which fail to disconnect/reconnect.

On my 486-66, with a T128, I use 90% of my CPU time to sustain a throughput of 547K/sec on a drive with a headrate of 1080K/sec with a T128 board.

Sometimes acceptable for slow tapes and CDROMs when low cost is essential.

FIFO Polled

Boards using FIFO polled I/O put a small (typically 8K) buffer between the CPU and the SCSI bus, and often implement some amount of intelligence. The net effect is that the CPU is only tied up when it is transferring data at top speed to the FIFO and when it's handling the rest of the interrupt processing for FIFO empty conditions, disconnect/reconnect, etc.

Peak transfer rates should be sufficient to handle most SCSI devices, and have been measured at up to 4M/sec using raw SCSI commands to read 64K blocks on a fast Seagate Baracuda with an Adaptec 1520.

CPU utilization is dependent on the transfer rates of the devices, with faster devices generating more interrupts per unit time which require more CPU processing time. Although CPU usage may be high (perhaps 75%) with fast devices, the system usually remains usable. These boards will provide excellent interactive performance with broken devices which don't disconnect/reconnect (typically cheap CDROM drives)

Recommended for persons on a budget.

Slave DMA

Drivers for boards using slave DMA program the PC's DMA controller for a channel when they do a data transfer, and return control to the CPU.

Peak transfer rates are usually handicapped by the poor DMA controller used on PCs, with one such 8-bit board having problems going faster than 140-150K/sec with one mainboard.

CPU utilization is very reasonable, slightly less than what is seen with FIFO polled I/O boards. These boards are very tolerant of broken devices which don't disconnect/reconnect (typically cheap CSG limitDROM drives).

Acceptable for slow CDROM drives, tapes, etc.

Busmastering DMA

These boards are intelligent. Drivers for these boards throw a SCSI command, the destination target and lun, and where the data should end up in a structure, and tell the board "Hey, I have a command for you." The driver returns control to various running programs, and eventually the SCSI board gets back and says that it's done.

Since the intelligence is in the host adapter firmware and not the driver, drivers for these boards typically support more features - synchronous transfers, tagged queuing, etc.

With the clustered read/write patches, peak transfer rates through the file system approach 100% of head rate writing, 75% reading.

CPU utilization is minimal, irregardless of I/O load, with a measured 5% CPU usage while accessing a double speed CDROM on an Adaptec 1540 and 20% while sustaining a 1.2M/sec transfer rate on a SCSI disk.

Recommended in all cases where money is not extremely tight, the main board is not broken (some broken main boards do not work with bus masters), and applications where time to data is more important than throughput are not being run (bus master overhead may hit 3-4ms per command).

10.2 Scatter/gather

The second most important driver/hardware feature with respect to performance is support for scatter/gather I/O. The overhead of executing a SCSI command is significant - on the order of milliseconds. Intelligent bus masters like the Adaptec 1540 may take 3-4ms to process a SCSI command before the target even sees it. On unbuffered devices, this overhead is always enough to slip a revolution, resulting in a transfer rate of about 60K/sec (assuming a 3600RPM drive) per block transfered at a time. So, to maximize performance, it is necessary to minimize the number of SCSI commands needed to transfer a given amount of data by transferring more data per command. Due to the design of the Linux buffer cache, contiguous disk blocks are not contiguous in memory. With the clustered read/write patches, 4K worth of buffers are contiguous. So, the maximum amount of data which can be transfered per SCSI command is going to be 1K * # of scatter/gather regions without the clustered read/write patches, 4K * # of regions with. Experimentally, we've determined that 64K is a reasonable amount to transfer with a single SCSI command - meaning 64 scatter/gather buffers with clustered read/write patches, 16 without. With the change from 16K to 64K transfers, we saw an improvement from 50% of headrate, through the filesystem, reading and writing, to 75% and 100% respectively using an Adaptec 1540 series board.

10.3 Mailbox vs. non-mailbox

A number of intelligent host adapters, such as the Ultrastor, WD7000, Adaptec 1540, 1740, and BusLogic boards have used a mailbox-metaphor interface, where SCSI commands are executed by putting a SCSI command structure in a fixed memory location (mailbox), signaling the board (ie, raising the outgoing mail flag), and waiting for a return (incoming mail). With this high level programming interface, users can often upgrade to a newer board revision to take advantage of new features, such as FAST + WIDE SCSI, without software changes. Drivers tend to be simpler to implement, may implement a larger feature set, and may be more stable.

Other intelligent host adapters, such as the NCR53c7/8xx family, and Adaptec AIC-7770/7870 chips (including the 274x, 284x, and 2940 boards) use a lower level programming interface. This may prove faster since processing can be shifted between the board's processor and faster host CPU, allow better flexibility in implementing certain features (ie, target mode for arbitrary devices), and these boards can be built for less money (In some cases, this is passed on to the consumer (ie, most NCR boards)). On the down side, drivers tend to be more complex (read : there is more potential for bugs), and must be modified to take advantage of the features present on newer chips.

10.4 Bus types

Bus type is the next thing to consider, with choices including ISA, EISA, VESA, and PCI. Marketing types often spout of absurd bandwidth numbers based on burst transfer rates and fiction, which isn't very useful. Instead, I've chosen to state "real-world" numbers based on measured performance with various peripherals.

Bus

Bandwidth, description,

ISA

Bandwidth is slightly better than 5M/sec for busmastering devices. With an ISA bus, arbitration for busmasters is performed by the venerable 8237 third party DMA controller, resulting in relatively high bus acquisition times. Interrupt drivers are tri-state and edge triggered, meaning interrupts cannot be shared. Generally, ISA is unbuffered, meaning the host/memory bus is tied up whenever a transfer is occuring. No mechanism is provided to prevent bus-hogging.

VESA

Bandwidth is about 30M/sec. Some VESA systems run the bus out of spec, rendering them incompatible with some boards, so this should be taken into consideration before purchasing hardware without a return guarantee. Generally, VESA is unbuffered, meaning meaning the host/memory bus is tied up whenever a transfer is occuring.

EISA

Bandwidth is about 30M/sec, with busmastering operations generally being faster than VESA. Some EISA systems buffer the bus, allowing burst transfers to the faster host/memory bus and minimizing impact on CPU performance. EISA interrupt drivers may be either tri-state edge-triggered or open collector level-active, allowing interrupt sharing with drivers that support it. Since EISA allocates a separate address space for each board, it is usually less prone to resource conflicts than ISA or VESA.

PCI

Bandwidth is about 60M/sec. Most PCI systems implement write posting buffers on the host bridge, allowing speed mismatches on either side to have a minimum impact on bus/CPU performance. PCI interrupt drivers are open collector level-active, allowing interrupt sharing with drivers that support it. Mechanisms are provided to prevent bus hogging, and for both master and slave to suspend a bus-mastering operation.

Since PCI provides a plug-n-play mechanism with writeable configuration registers on every board, in a separate address space, a properly implemented PCI system is plug-and play.

PCI is extremely strict as to trace length, loading, mechanical specifications, etc. and ultimately should be more reliable than VESA or ISA.

In summary, PCI is the best PC bus, although it does have its dark side. PCI is still in its infancy, and although most manufacturers have ironed out the problems, there is still stock of older, buggy PCI hardware and broken main BIOSes. For this reason, I _strongly_ recommend a return guarantee on the hardware. While the latest PCI mainboards are truly plug-and-play, older PCI boards may require the user to set options with both jumpers and in software (ie, interrupt assignments). Although many users have resolved their PCI problems, it has taken time and for this reason I cannot recommend a PCI purchase if having the system operational is extremely time critical.

For many slower SCSI devices, such as disks with head rates around 2M/sec or less, CDROMs, and tapes, there will be little difference in throughputs with the different PC bus interfaces. For faster contemporary SCSI drives (Typical high end multi-gigabyte drives have a head rate of 4-5M/sec, and at least one company is currently ALPHA testing a parallel head unit with a 14M/sec head rate), throughput will often be significantly better with controllers on faster busses, with one user noting a 2.5 fold performance improvement when going from an Adaptec 1542 ISA board to a NCR53c810 PCI board.

With the exception of situations where PCI write-posting or a similar write-buffering mechanism is being used, when one of the busses in your system is busy, all of the busses will be unaccessible. So, although bus saturation may not be interfering with SCSI performance, it may have a negative effect on interactive performance. Ie, if you have a 4M/sec SCSI disk under ISA, you'll have lost 80% of your bandwidth, and in an ISA/VESA system would only be able to bitblt at 6M/sec. In most cases, a similar impact on processing jobs in the background would also be felt.

Note that having over 16M of memory does not preclude using an ISA busmastering SCSI board. Unlike various broken operating systems, Linux will double buffer when using a DMA with an ISA controller and a transfer is ultimately destined for an area above 16M. Performance on these transfers only suffers by about 1.5%, ie not noticeably.

Finally, the price difference between bus masters offered with the different bus interfaces is often minimal.

With all that in mind, based on your priorities you will have certain bus preferences

```
Stability, time critical installations,        EISA ISA VESA PCI
           and poor return policies
Performance, and typical hobbiest              PCI EISA VESA ISA
           installations
```

As I pointed out earlier, bus mastering versus other transfer modes is going to have a bigger impact on total system performance, and should be considered more important than bus type when purchasing a SCSI controller.

10.5 Multiple devices

If will you have multiple devices on your SCSI bus, you may want to see whether the host adapter/driver that you are considering supports more than one outstanding command at one time. This is almost essential if you'll be running a tape drive, and very desirable if you are mixing devices of different speeds, like a CD ROM and a disk drive. If the linux driver only supports one outstanding command, you may be locked out of your disk drive while a tape in the tape drive is rewinding or seeking to end of media (perhaps for half an hour). With two disk drives, the problem will not be as noticeable, although throughput would approach the average of the two transfer rates rather than the sum of the two transfer rates.

10.6 SCSI-I, SCSI-II, SCSI-III FAST and WIDE options, etc.

Over the years, SCSI has evolved, with new revisions of the standard introducing higher transfer rates, methods to increase throughput, standardized commands for new devices, and new commands for previously supported devices.

In and of themselves, the revision levels don't really mean anything. Excepting minor things like SCSI-II not allowing the single initiator option of SCSI-I, SCSI is backwards compatible, with new features being introduced as options and not mandatory. So, the decision to call a SCSI adapter SCSI, SCSI-II, or SCSI-III is almost entirely a marketing one.

10.7 Driver feature comparison

Driver feature comparison (supported chips are listed in parenthesis)

```
Driver                               Simultaneous    SG             > 1
                Transfer mode        Commands        limit          Boards
```

		total/LUN		
AM53C974	Busmastering DMA	12s/1s	255s	Y
aha152x	FIFO(8k) Polled	7s/1s	255s	N
(AIC6260,				
AIC6360)				
aha1542	Busmastering DMA	8s/1s	16	Y
aha1740	Busmastering DMA	32s	16	N
aha274x	Busmastering DMA	4s/1s	255s	Y
BusLogic	Busmastering DMA	192/31	128s, 8192h	Y
(values are for BT-948/958/958D, older boards support fewer commands)				
eata_dma	Busmastering DMA	64s-8192h/2-64	512s, 8192h	Y
fdomain	FIFO(8k) Polled	1s	64s	N
(TMC1800,	except TMC18c30			
TMC18c30,	with 2k FIFO			
TMC18c50,				
TMC36c70)				
in2000*	FIFO(2k) Polled	1s	255s	N
g_NCR5380	Pure Polled	16s/2s	255s	Y
(NCR5380,				
NCR53c80,				
NCR5381,				
NCR53c400)				
gsi8*	Slave DMA	16s/2s	255s	
(NCR5380)				
PAS16	Pure Polled	16s/2s	255s	Y
(NCR5380)	or Interlocked Polled			
	(fails on some systems!)			
seagate	Interlocked Polled	1s/1s	255s	N
wd7000	Busmastering DMA	16s/1s	16	Y
t128	Interlocked Polled	16s	255s	Y
(NCR5380)				
qlogic	Interlocked Polled	1s/1s	255s	N
ultrastor	Busmastering DMA	16s/2s	32	Y
53c7,8xx	Busmastering DMA			
(NCR53c810,				
NCR53c815,				
NCR53c820,				
NCR53c825)				
rel5		1s/1s	127s	N
rel10		8s/1s	127s	Y

Notes :

1. drivers flagged with an '*' are not included with the distribution kernel, and binary boot images may be unavailable.
2. numbers suffixed with an 's' are arbitrary limits set in software which may be changed with a compile time define.
3. hardware limits are indicated by an 'h' suffix, and may differ from the software limits currently imposed by the Linux drivers.
4. unsuffixed numbers may indicate either hard or soft limits.
5. rel5 of the NCR53c810 driver is included in the stock 1.2.x and 1.3.x kernels; rel10 is available via anonymous FTP.
6. With the exception of the AM53C974, the busmastering DMA boards are intelligent; with the NCR executing microcode from main memory, the AIC7770 executing microcode from on-chip RAM, and the rest using a mailbox-style interface.

10.8 Board comparison

Board	Driver	Bus	Price	Notes
Adaptec AIC-6260	aha152x	ISA		chip, not board
Adaptec AIC-6360 (Used in most VESA/ISA multi-IO boards with SCSI, Zenon mainboards)	aha152x	VLB		chip, not board
Adaptec 1520	aha152x	ISA		
Adaptec 1522	aha152x	ISA	$80	1520 w/FDC
Adaptec 1510	aha152x	ISA		1520 w/out boot ROM, won't autoprobe.
Adaptec 1540C	aha1542	ISA		
Adaptec 1542C	aha1542	ISA		1540C w/FDC
Adaptec 1540CF	aha1542	ISA		FAST SCSI-II
Adaptec 1542CF	aha1542	ISA	$200	1540CF w/FDC
Adaptec 1640	aha1542	MCA		
Adaptec 1740	aha1740	EISA		discontinued
Adaptec 1742	aha1740	EISA		discontinued, 1740 w/FDC
Adaptec 2740	aha274x	EISA		
Adaptec 2742	aha274x	EISA		w/FDC
Adaptec 2840	aha274x	VLB		
Adaptec 2842	aha274x	VLB		w/FDC
Adaptec 2940	aha274x	PCI		
Always IN2000	in2000	ISA		

BusLogic BT-948	BusLogic	PCI	$180	Ultra SCSI
BusLogic BT-958	BusLogic	PCI	$230	Wide Ultra SCSI

(see the section 5.8 for additional BusLogic board descriptions)

DPT	PM2011	eata_dma	ISA		FAST SCSI-II
	PM2012A	eata_dma	EISA		FAST SCSI-II
	PM2012B	eata_dma	EISA		FAST SCSI-II
	PM2021	eata_dma	ISA		FAST SCSI-II
	PM2022	eata_dma	EISA		FAST SCSI-II
	PM2024	eata_dma	PCI		FAST SCSI-II
	PM2122	eata_dma	EISA		FAST SCSI-II
	PM2322	eata_dma	EISA		FAST SCSI-II
	PM2124	eata_dma	PCI		FAST SCSI-II
	PM2124	eata_dma	PCI		FAST SCSI-II
	PM2124	eata_dma	PCI		FAST SCSI-II
	PM2124	eata_dma	PCI		FAST SCSI-II
	PM2124	eata_dma	PCI		FAST SCSI-II
	PM2124	eata_dma	PCI		FAST SCSI-II
	PM2041W	eata_dma	ISA		Wide Single-ended SCSI-II
	PM2041UW	eata_dma	ISA		Ultra Wide Single-ended
	PM2042W	eata_dma	EISA		Wide Single-ended
	PM2042UW	eata_dma	EISA		Ultra Wide Single-ended
	PM2044W	eata_dma	PCI		Wide Single-ended
	PM2044UW	eata_dma	PCI		Ultra Wide Single-ended
	PM2142W	eata_dma	EISA		Wide Single-ended
	PM2142UW	eata_dma	EISA		Ultra Wide Single-ended
	PM2144W	eata_dma	PCI		Wide Single-ended
	PM2144UW	eata_dma	PCI		Ultra Wide Single-ended
	PM3021	eata_dma	ISA		multichannel raid/simm sockets
	PM3122	eata_dma	EISA		multichannel/raid
	PM3222	eata_dma	EISA		multichannel raid/simm sockets
	PM3224	eata_dma	PCI		multichannel raid/simm sockets
	PM3334	eata_dma	PCI		Wide Ultra SCSI multichannel raid/simm sockets

DTC 3290	aha1542	EISA		Although it should work, due to documentation release polcies, DTC hardware is unsupported

DTC 3130	53c7,8xx	PCI		'810
DTC 3130B	53c7,8xx	PCI		'815
DTC 3292	aha1542	EISA		3290 w/FDC
DTC 3292	aha1542	EISA		3290 w/FDC
Future Domain 1680	fdomain	ISA		FDC
Future Domain 3260	fdomain	PCI		
NCR53c810 (boards sold by FIC, Chaintech, Nextor, Gigabyte, etc. Mainboards with chip by AMI, ASUS, J-Bond, etc. Common in DEC PCI systems)	53c7,8xx	PCI	$60 (board)	chip, not board. Boards don't include BIOS, although most non-NCR equipped main boards have the SDMS BIOS
NCR53c815 (Intel PCISCSIKIT, NCR8150S, etc)	53c7,8xx	PCI	$100	NCR53c810 plus bios
NCR53c825	53c7,8xx	PCI	$120	Wide variant of NCR53c815. Note that the current Linux driver does not negotiate for wide transfers.
Pro Audio Spectrum 16	pas16	ISA		Sound board w/SCSI
Seagate ST01	seagate	ISA	$20	BIOS only works with some drives
Seagate ST02	seagate	ISA	$40	ST01 w/FDC
Sound Blaster 16 SCSI	aha152x	ISA		Sound board w/SCSI
Western Digital 7000	wd7000	ISA		w/FDC
Trantor T128	t128	ISA		
Trantor T128F	t128	ISA		T128 w/FDC and support for high IRQs
Trantor T130B	g_NCR5380	ISA		
Ultrastor 14F	ultrastor	ISA		w/FDC
Ultrastor 24F	ultrastor	EISA		w/FDC
Ultrastor 34F	ultrastor	VLB		

Notes :

1. Trantor was recently purchased by Adaptec, and some products are being sold under the Adaptec name.
2. Ultrastor recently filed for Chapter 11 Bankruptcy, so technical support is non-existent at this time.
3. The price for the busmastering NCR53c810 boards is not a typo, includes the standard ASPI/CAM driver package for DOS, OS/2 and Windows (32 bit access), and other drivers are available for free download.

Some people have had luck with the following companies :

```
SW (swt@netcom.com) (214) 907-0871 fax (214) 907-9339
```

As of 23 Dec 1995, their price was $53 on '810 boards.

4. Adaptec's recent SCSI chips show an unusual sensitivity to cabling and termination problems. For this reason, I cannot recommend the Adaptec 154x C and CF revisions or the 2xxx series.

   Note that the reliability problems do not apply to the older 154x B revision boards, 174x A revision boards, or to my knowledge AIC-6360/AIC-6260 based boards (1505, 1510, 1520, etc).

   Also, the quality of their technical support has slipped markedly, with long delays becoming more common, and their employees being ignorant (suggesting there were non-disclosure policies affecting certain literature when there were none), and hostile (ie, refusing to pass questions on to some one else when they couldn't answer them).

   If users desire handholding, or wish to make a political statement, they should take this point into consideration. Otherwise, the Adaptec 152x/1510/1505 are nicer than the other ISA boards in the same price range, and there are some excellent deals on used and surplus 154x B revision boards and 1742 boards which IMHO outweigh the support problems.

5. All DPT boards can be upgraded with cache and raid modules, most of the boards are also available in Wide and/or Differential versions.

6. The various NCR boards are not entirely equivalent. Ie, while the ASUS SC200 uses active termination, many other NCR53c810 boards use passive termination. Most '825 boards use active termination, but some use a ROM for BIOS and others have a FLASH ROM. Most '825 boards have a WIDE external connector, WIDE internal connector, and narrow internal connector, although a few (ie, CSC's less expensive model) lack the narrow internal connector.

10.9 Summary

Most ISA, EISA, VESA, and PCI users will probably be served best by a BusLogic MultiMaster board, due to its performance, features such as active termination, and Adaptec 1540 compatibility. There are a number of models available with EISA, ISA, PCI, and VESA local bus interfaces, in single ended and differential, and 8/16 bit SCSI bus widths. The most recent Ultra SCSI PCI models, the BT-948/958/958D, also include Flash ROM for easy firmware updates, as well as automatic "smart" termination.

People with the need for the highest possible IO performance at their fingertips should consider the boards from DPT, which are the only ones that support RAID, caching and more than one SCSI channel.

People with PCI systems should consider NCR53c8xx based boards. These are bus mastering SCSI controllers, '810s are available quantity one for $53 (ie, cheaper than the Adaptec 1520). C't magazine benchmarked the boards as faster than both the Adaptec 2940 and BusLogic BT-946C (under DOS), and they get reasonable performance under Linux (up to 6M/sec through the file system). The disadvantages of these boards versus the BusLogics are that they aren't Adaptec 1540 compatible, may or may not come with active termination, you'll need the latest driver revision (standard in 1.3.5x, also available via anonymous FTP for 1.2.x) to make full use of the hardware, and are more likely to have problems than with a mailbox interface board like a BusLogic or DPT.

Where everything working right on the first try is imperative, a BusLogic MultiMaster or DPT board is probably optimal due to the complexity and potential for problems in non-mailbox interface boards like the NCR53c8xx and Adaptec AIC7xxx .

People wanting non-PCI SCSI on a limited budget will probably be happiest finding a surplus or used Adaptec 154x B revision or 174x A revision, or an Adaptec 1520 clone of some sort (about $80) if they want new hardware. These boards offer reasonable throughput and interactive performance at a modest price.

11 Assignment of minor numbers

Due to constraints imposed by Linux's use of a sixteen bit dev_t with only eight bits allocated to the minor number, SCSI disk, tape, CDROM, and generic minor numbers are assigned dynamically. according to the following procedure :

```
For all SCSI host adapters, from scsi0 through scsiN
 For all SCSI IDs on this bus, from 0 through 7, except for
   this host adapter's ID
     For all logical units, from 0 through max_scsi_luns
       - Probe the bus, target, and LUN combination by
         issuing a TEST UNIT READY command.  If we don't
         think a unit was here, don't probe any more LUNs
         on this bus + SCSI ID.
       - Send an INQUIRY command to determine what we've
         found; including the device type, vendor, model,
         firmware revision, etc.
       - Pass the results of this to a special recognition
         function for each high level driver present (i.e. disk,
         tape, etc).  Attach this device to the next available
         unit for any drivers that are willing to drive this.
         The generic device will attach to all devices.
       - If it was SCSI-I, or in a list of devices known
         not to handle multiple LUNs, don't probe any more
         LUNs on this bus + SCSI ID.
       - If it is a device known to have multiple LUNs, then
         a scan of the full LUN spectrum is forced, overriding
         max_scsi_luns.
```

There are frequently problems with this approach because if you have a system where some devices are only present some of the time, then the minor numbers for a given device will depend upon which devices were present at boot time. This can present problem, because rc scripts or the file /etc/fstab might contain instructions for mounting specific partitions which fails when the disk appears with a different minor number.

This problem has not yet been fully solved. There is a program which can be found on tsx-11 that creates a /dev/scsi hierarchy based upon host number, id and lun. This is a bit clumsy, but it would help to alleviate some of the problems.

A better solution will probably come out of the /proc/scsi pseudo directory. This is currently a work in progress, so at present we cannot say exactly the form of the solution, but at the time of this writing this appears to be a promising approach for resolving some of these issues.