Linux – Optical Disk HOWTO

Table of Contents

Table of Contents

Linux − Optical Disk HOWTO

Skip Rye, abr@preferred.com

v1.8, 12 May 2003

This document describes the installation and configuration of optical disk drives for Linux. Many of the articles are presented as is or with modifications without the actual hardware to double check the correctness. Use this information at your own risk! Please, if any one has experiences with optical storage under Linux, send it and I will update it in SGML and forward it to the Linux community. I'll assume its OK to include your E−mail address unless you specifically tell me NOT to! V1.8 is a long awaited update. I have been building a cabin since 1997 and its taken till now to complete. I appoligize to all who have given input long ago for my delay with this release!!!!

---

5. Optical jukeboxes

6. MO and other media technologies − Gene Cumm

7. Phase Change Optical Technology

8. Optical Disk HOWTO Development Page

---

1. Disclaimer

Neither the author nor the distributors, or any other contributor of this HOWTO are in any way responsible for physical, financial, moral or any other type of damage incurred by following the suggestions in this text.

---

2. Copyright

The "Optical Disk−HOWTO" and "LF1000 mini−HOWTO" are copyrighted.

2.1 LF1000 mini−HOWTO

2.2 Optical Disk−HOWTO

---

3. Magneto Optical Technology − Daniel Kobras

3.1 Introduction

Daniel Kobras <kobras@linux.de>

Magneto optical drives use a "far field" magnetic field and a laser to change polarization of a magnetic media. At temperatures below 180−200°C (350−390°F) magnetic polarization is "frozen" into the media. However when heated above this so−called Curie−temperature a static external magnetic field can change the polarization. When the media cools down below its Curie−temperature, the information is frozen again. A high power write laser is used to heat the disk surface to the appropriate temperature at which time the "Far field" can set the polarization on the disk magnetic surface.

Read back is based on the so−called Kerr effect, i. e. depending on the direction of the magnetic field on the disk's surface, the plane of polarization of the incoming laser beam is slightly rotated and the information can be restored.

The use of a laser for polarization change allows for bit and track densities much higher than on conventional "flying" magnetic heads. The "far field" means no more "head crashes" − that is assuming your disk label doesn't peal off during the load or you don't leave one of those sticky pads on the disk cartridge. Nowadays the most commonly used 3.5" media have a capacity of 640MB[*] per platter but there are still media with 540MB, 230MB and 128MB. On some models both sides of the media are used yielding up to a capacity of 1.3GB − you must remove the media and flip it over to use the other 640MB though. There are 5.25" media with up to a total of 2.6GB, but these have to be flipped as well.

The major drawback with ordinary magneto optical media was their need for an extra erase cycle considerably slowing down write speed. That's where LIMDOW media come in. LIMDOW (Light Intensity Modulated Direct OverWrite) disks use a more sophisticated set of five different magnetic layers. Thus the erase cycle can be omitted yielding a 33% speed up, as only one write and one verify cycle have to be performed. Read back is identical to ordinary disks. Please check with your drives manual if you want to use LIMDOW media. I only have experience with Fujitsu's M2513 which works well with LIMDOW. As far as other drives are concerned I simply don't know.

Manufacturers claim the life time of magneto optical media to be 30 years and up. Disks can be rewritten at least 10 million times (1 million for LIMDOW media). Reading is claimed safe for at least 100 million times.

[*] There's a sort of religious discussion going on whether 1MB should be understood as 1x1.000x1.000 bytes or rather 1x1.024x1.024 bytes. Here we use 1MB==1.000.000 bytes, the definition preferred by vendors for obvious reasons. Don't worry if Linux reports your media to be smaller − it's just a matter of definition.

3.2 Setup

First of all, make sure your MO drive is sanely jumpered, i.e. make sure its SCSI id is unique on your system, Parity checking and SCAM mode settings resemble those of your other SCSI devices as well as your controller and do _not_ enable any weird looking options such as "Mac Mode" or the like. Your drive might be equipped with an internal write cache, but since Linux already does pretty good caching on its own, don't expect too much of a performance gain, if any. Also keep in mind that each additional level of caching is a source of possible data loss or corruption in case of failing hardware. Consequently the recommended paranoia setting is to turn off the write cache.

As long as you're not using 640MB disks, setting up the MO drive is rather straightforward. Assuming your drive is properly installed, at boot/insmod time, your SCSI−Controller should notify you of the newly added drive and configure another SCSI device like /dev/sda, /dev/sdb... (Keep in mind that the SCSI bus is scanned with increasing SCSI id, so if your SCSI hard disk for example is ID 4 and used to be /dev/sda and your MO drive has ID 3, the MO will now be /dev/sda whilst the HD is /dev/sdb.) Working with your MO is no different from working with an ordinary hard disk. You can partition it (more information on this topic is given below), create file systems, mount it as usual. Note that as long as the disk is mounted the drive is locked and you won't be able to change the disk.

Be careful when trying to get 640MB disks to work. These use a hard−sector size of 2048 bytes, 2.0.xx kernels will support only 512 and 1024 bytes per sector. However 2048 byte support has been added to 2.1.32 and up. If you for some reason have to stick to 2.0.xx, there are several patches floating around, for example at

* http://liniere.gen.u−tokyo.ac.jp/2048.html, * http://wwwcip.informatik.uni−erlangen.de/ orschaer/mo/ * http://elektra.e−technik.uni−ulm.de/ mbuck/linux/patches.html

Be sure to use a either a patched version of fdisk available at some of the sites above or a recent enough version from the official util− linux package supporting the −b option. (Invoke with fdisk −b 2048 /dev/sdXX when partitioning 2048 byte media.)

3.3 Access

There are two alternatives of how to access your disks: the ordinary method of creating one or more partitions or just accessing the raw drive, which in Win/DOS environments is also known as the superfloppy format.

The first method will require non−640MB disks or a 2048−byte−aware fdisk, the latter is suitable for any kind of disk, however these disks cannot be read with Windows NT up to version 4.0. There's a comment on Fujitsu's web−pages that super−floppy support will be added to NT in the future.

Assume your MO drive is /dev/sdb. To create a partition simply enter fdisk /dev/sdb (or fdisk −b 2048 /dev/sdb with 640MB media and a recent copy of fdisk) as root and go on like you were to partition a hard disk. If unsure about what to do, have a look at the fdisk man page. Next create a file−system on each partition with a command like mke2fs −m 0 /dev/sdb1. For 640MB disks be sure to specify the −b 2048 flag. If you want to use super−floppies instead, leave out the fdisk part and create your file−system on the raw device, for example mke2fs −b 2048 −m 0 /dev/sdb. mke2fs will request confirmation before formatting a raw device. You might want to double check if /dev/sdb *really* is your MO drive and not your hard disk by chance. :) During the boot process (or when loading the low level SCSI module), Linux might moan about an invalid partition table if a super−floppy is in the MO drive. You can safely ignore this message.

*NOTE: Partitions on 2048 byte sectored media were broken throughout the whole 2.1 kernel series, meaning that you can happily partition your media with 2.1.xxx but will be unable to use them with any OS other than Linux 2.1! In other words: DON'T DO IT. If for any reason you still have to access your MOs on Linux 2.1 use super−floppies which do fine. This problem hopefully is completely fixed starting with Linux 2.2.2.

File−systems other than ext2 will work on non−640MB disks as well, for 640MB disks there are some caveats: 2048 byte blocks must be supported by the low level file−system code in the kernel and the appropriate mkfs tool should take an argument like −b 2048 to specify the block size. Kernel requirements are met at least for ext2 and msdos/vfat code in the 2.1.xx kernel line. The above mentioned patches should fix this as well for 2.0.xx kernels. I don't have any experience with other file−systems so I'd appreciate any comments. Of the mkfs tools mke2fs will definitely accept the −b flag. mkdosfs is trickier though: there's no top level maintainer anymore, but some distributions have their own maintainers and ship their own versions. Debian's mkdosfs is one such example. Beginning with version 1.0−17 it supports media with large sector sizes. You have to add an option −S 2048. Pass −I as well if you want to format a super−floppy. The latest Debian version of mkdosfs should always be available from ftp.debian.org. Look out for a package called dosfstools.

3.4 Speed

For the really curious but still undecided I've crammed up some figures as returned by bonnie. These are for the Fujitsu M2513 spinning at 3600 rpm, an outdated model now replaced by a version spinning at 4300 rpm. I guess the transfer rates for the new drive will scale with the spin ratio or pretty close to it. Tests were performed running a slightly patched 2.2.2pre4 kernel. (Err... looks like I've disabled verify on my drive, better not do that!)

```
LIMDOW - ext2-filesystem - superfloppy:

    -------Sequential Output-------- ---Sequential Input-- --Random--
    -Per Char- --Block--- -Rewrite-- -Per Char- --Block--- --Seeks---
 MB K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU  /sec %CPU
400  1024 16.3  1816  2.8   620  1.7   975 13.5  1952  2.2  41.4  0.7


LIMDOW - vfat-filesystem - superfloppy:

    -------Sequential Output-------- ---Sequential Input-- --Random--
    -Per Char- --Block--- -Rewrite-- -Per Char- --Block--- --Seeks---
 MB K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU  /sec %CPU
400   387  8.3   410  2.9   414  3.4   669 13.4   736  5.4   5.2  3.9
```

The bottom line is: performance on vfat sucks like hell. If you have an option, use ext2!

3.5 Sample session

Here's an example of what accessing the MO look like on my machine:

```
yksi:~# modprobe scsi_mod
scsi: ***** BusLogic SCSI Driver Version 2.1.15 of 17 August 1998 *****
scsi: Copyright 1995-1998 by Leonard N. Zubkoff <lnz@dandelion.com>
scsi0: Configuring BusLogic Model BT-930 PCI Ultra SCSI Host Adapter
scsi0:   Firmware Version: 5.02, I/O Address: 0xDE00, IRQ Channel: 18/Level
scsi0:   PCI Bus: 0, Device: 15, Address: 0xFE00F000, Host Adapter SCSI ID: 7
scsi0:   Parity Checking: Enabled, Extended Translation: Enabled
scsi0:   Synchronous Negotiation: Ultra, Wide Negotiation: Disabled
scsi0:   Disconnect/Reconnect: Enabled, Tagged Queuing: Enabled
scsi0:   Driver Queue Depth: 255, Scatter/Gather Limit: 128 segments
scsi0:   Tagged Queue Depth: Automatic, Untagged Queue Depth: 3
scsi0:   Error Recovery Strategy: Default, SCSI Bus Reset: Enabled
scsi0:   SCSI Bus Termination: Disabled, SCAM: Disabled
scsi0: *** BusLogic BT-930 Initialized Successfully ***
scsi0 : BusLogic BT-930
scsi : 1 host.
Vendor: PLEXTOR   Model: CD-ROM PX-32TS    Rev: 1.03
Type:   CD-ROM                             ANSI SCSI revision: 02
Vendor: FUJITSU   Model: M2513A            Rev: 1300
Type:   Optical Device                     ANSI SCSI revision: 02
```

```
scsi0: Target 1: Queue Depth 3, Synchronous at 20.0 MB/sec, offset 15
scsi0: Target 3: Queue Depth 3, Synchronous at 10.0 MB/sec, offset 10
```

As you can see, I have two SCSI devices attached: one CD−ROM drive and one MO drive. As CD−ROMs do have SCSI devices of their own (/dev/scdX), the MO is assigned to /dev/sda.

Let's create an ext2−based super−floppy on the media:

```
yksi:~# mke2fs -m 0 -b 2048 /dev/sda
mke2fs 1.12, 9-Jul-98 for EXT2 FS 0.5b, 95/08/09
/dev/sda is entire device, not just one partition!
Proceed anyway? (y,n) y
Detected scsi removable disk sda at scsi0, channel 0, id 3, lun 0
SCSI device sda: hdwr sector= 2048 bytes. Sectors= 310352 [606 MB] [0.6 GB]
sda: Write Protect is off
 sda:
Linux ext2 filesystem format
Filesystem label=
155344 inodes, 310352 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
Block size=2048 (log=1)
Fragment size=2048 (log=1)
19 block groups
16384 blocks per group, 16384 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
         16384, 32768, 49152, 65536, 81920, 98304, 114688,
         131072, 147456, 163840, 180224, 196608, 212992, 229376,
         245760, 262144, 278528, 294912

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

Now mount the media to directory /mnt/mo (which already exists).

```
yksi:~# mount -t ext2 /dev/sda /mnt/mo
yksi:~# ls /mnt/mo
lost+found
yksi:~# df
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/hda6             124407   48963    69020     42%   /
/dev/hda7             256592      30   243310      0%   /tmp
/dev/hda8             124407   31750    86233     27%   /var
/dev/hda9             505440  174092   305244     36%   /home
/dev/hda10           2028098 1278972   644304     66%   /usr
/dev/hda11           2028098 1551617   371659     81%   /usr/local
/dev/sda              601134      26   601108      0%   /mnt/mo
```

/mnt/mo can now be used like any ordinary hard disk. You may also choose to add a line like the following to your /etc/fstab:

```
/dev/sda    /mnt/mo    ext2 defaults,noauto 0 0
```

Then mount /mnt/mo will suffice to mount any ext2−formatted media. Before removing the media from your MO drive, don't forget to unmount it.

```
yksi:/mnt/mo# umount /mnt/mo           # (Whoops!)
umount: /mnt/mo: device is busy
yksi:/mnt/mo# cd ..
yksi:/mnt# umount /mnt/mo
yksi:/mnt#
```

Pretty easy, isn't it?

4. Magneto Optical Drive experiences under Linux

4.1 Olympus, Epson, Mitsubishi MK230LK3 − Stephan Shuichi Haupt

Stephan Shuichi Haupt <stephan@bios.t.u−tokyo.ac.jp>

```
Hi

I have noticed that there is not much information about
magneto-optical disks in the howto, which may be due to the fact that
these are not very popular in general. In Japan, MO drives are very
common, especially the 3.5' variety with media in 128MB (maybe not
available anymore), 230MB, and recently 640MB sizes. I suppose there
is plenty of info on usage of these drives with Linux in Japanese -
but that does not help most people for some reason ;-) MODs can be
used very much like any removable media and are handy for smaller
backups as the media are relatively inexpensive (about 10US$ / 640MB
as of 10-98). I can only comment on the usage of 230MB drives with
SCSI interface.

Drives used: several, no problems encountered (Olympus, Epson, currently
Mitsubishi MK230LK3). Drives may have strange jumper setting like "Mac
Mode" or such - naturally, disable.
If you decide to get a drive, pay attention the the
cache size - It can speed things up enormously, still speed will be
soso compared to hard disks, of course.

SCSI controllers: NCR53C810-based (Asus PCI-200), Adaptec APA-1460A,
Adaptec AHA2940.

Just install the drive as you would do with an additional SCSI hard
disk. It will show up as such. You don't need a disk in the drive when
booting.

There are two ways to format the disks:
a) A bit like a floppy. Just run mkfs on the raw device i.e. something like
sdb or sdc. I don't recommend this in general (see below).
b) Like a hard disk. Do fdisk on the raw device and then mkfs on the
partition as you would for a hard disk (like sdc0, I have never made
multiple partitions on a MOD).

What I have not tried is to boot from MOD, yet I cannot see why it
should not work. I would only recommend it for emergency system
recovery, however, due to MO drive performance.

Note: Purchased disks for Doze or Windog may be formatted "like
floppies" and cannot be used with either O(gre)S right away while MODs
formatted under linux as hard disks (partition FAT16 / type 6 and
mkdosfs) will work fine (only tested with NT 3.5/4.0).  Fdisk will
```

```
issue a warning upon exit that concerned FAT16 partitions and you do
better to take it seriously (look at the fdisk man-page).  The sector
size will not be automatically set properly for mkdosfs. Use "mkdosfs
-s 8". That came from some Japanese Web site in mid 1995 (Thanks to Ken
Kawabata for finding and deciphering it). Using the vfat file-system
with the disks works fine. I have only used FAT/DOSfs or Linux/ext2
formatted disks so far.

Additional Note: The media are probably a bit sensitive. Of course to
magnetic fields, but also to mechanical stress, some formats seem
to be more fragile than others (Mac format seemingly worst, data loss has
occurred when dropping disks during sneaker net traffic).


Though this does not steer anyone through particularly dense
jungle, it may be nice for completeness.


        Steve

--
***********************cut*here*or*do*not********************************
        S. Shuichi Haupt
        email stephan@bios.t.u-tokyo.ac.jp
        http://www.bios.t.u-tokyo.ac.jp/~stephan/

---------------- December 11 1998 update from Steve -------------------

OK, some problems will arise with MO disks occasionally. the safest
way to avoid them is not to use the disks "off the shelf".  trying to
mount disks can even result in kernel panics. i accidentally tried to
mount a 640MB disk (format windows95 it said, so maybe FAT32) as -t
vfat, this is not a thing to try.

also, 2.0.x kernels don't support 2048b block size (also 640MB disks).
a patch for 2.0.3x kernels seems to float around somewhere in Japan,
but i have not yet gotten hold of it.  here a link that certainly has
an English description:
http://elektra.e-technik.uni-ulm.de/~mbuck/linux/patches.html
or search the u-tokyo.ac.jp domain. the page of the developers is
hidden somewhere.

the best way to use these 640MB disks is therefore to do fdisk and
mkfs first. i have only done this with mke2fs on type 83 partitions:
mke2fs -b 2048 /dev/sdxy

i will check it out for FAT16 partitions and mkdosfs when i have some
spare time and disks.

my kernel version used is 2.1.124 (for all of the above).

Steve
--
***********************cut*here*or*do*not********************************
        Stephan Shuichi
office: Dept. for Mechano-Informatics, Yoshizawa Lab.
        Faculty for Engineering, University of Tokyo
        Tel 03-3812-2111 ext 6390, FAX 03-5802-2957
        email stephan@bios.t.u-tokyo.ac.jp
        http://www.bios.t.u-tokyo.ac.jp/~stephan/
private: --
```

4.2 Fujitsu DynaMO 640 − Phil Garcia

pgarcia@execpc.com

```
  You've probably already received a number of messages regarding the
Fujitsu DynaMO 640 - I have the 640SZI, which is the internal version;
the model number given in a SCSI probe is M2513-MCC3064SS.  I recently
installed this drive practically without a hitch.  I say practically
because the sector size of the 640 MB disks is 2048 bytes, which is
not supported in the Linux 2.0.x kernel but is supported in the
development kernels.  A patch for 2.0.x is available at
http://wwwcip.informatik.uni-erlangen.de/~orschaer/mo/
-- also at this site is a patched fdisk to use in conjunction with it.

Otherwise, installing the drive was no different from installing a
SCSI hard drive.  It runs well, and I'm very happy with it.

Phil Garcia
```

4.3 Panasonic LF−7010 − Philip Kerr

philip_kerr_at_wmc__brsf2@wmcmail.wmc.ac.uk

```
     Dear Skip

     In your Optical HOWTO, you asked for anyone else's experiences of
     installing optical drives under Linux.

     Please find below details of how I managed to get a Panasonic LF-7010
     (SCSI) working on my Sparc Classic.

     I'm using Redhat, 4.2 and 5.1

     Regards

     Philip Kerr
     philip.kerr@wmc.ac.uk


     ps I'm now trying to get the drive to work under Solaris 2.6... it's
     not an easy a job as it was under Linux!!
     ------------------------


     plugged the drive in (on id5)...

     powered up the Sparc...


     the following came up....

     scsi0 : Sparc ESP100A-FAST
     scsi : 1 host.
     Vendor: SAMSUNG   Model: WN32162U          Rev: 0100
     Type:   Direct-Access                      ANSI SCSI revision: 02

     Detected scsi disk sda at scsi0, channel 0, id 3, lun 0
     Vendor: MATSHITA  Model: LF-7010  (00:06)  Rev: 1.42
```

```
Type:   Optical Device                     ANSI SCSI revision: 02
Detected scsi removable disk sdb at scsi0, channel 0, id 5, lun 0 scsi
: detected 2 SCSI disks total.
esp0: target 3 [period 100ns offset 15 10.00MHz FAST SCSI-II]
SCSI device sda: hdwr sector= 512 bytes. Sectors= 4236661 [2068 MB]
[2.1 GB]
esp0: target 5 [period 248ns offset 4 4.03MHz synchronous SCSI] sdb :
READ CAPACITY failed.
sdb : status = 0, message = 00, host = 0, driver = 28 sdb : extended
sense code = 2
sdb : block size assumed to be 512 bytes, disk size 1GB.
sunlance.c:v1.9 21/Aug/96 Miguel de Icaza (miguel@nuclecu.unam.mx)
eth0: LANCE 08:00:20:04:3d:cf
eth0: using auto-carrier-detection.
Partition check:
sda: sda1 sda2 sda3 sda4 sda5 sda6 sda7 sda8
sdb:scsidisk I/O error: dev 08:10, sector 0, absolute sector 0 unable
to read partition table

I edited my fstab, adding the entry for the drive (on sdb)

==========
/etc/fstab
==========
/dev/sda1          /                        ext2    defaults        1 1
/dev/sda2          swap                     swap    defaults        0 0
/dev/fd0           /mnt/floppy              msdos   noauto,user     0 0
/dev/sr0           /mnt/cdrom               iso9660 noauto,ro,user  0 0
/dev/sdb           /mnt/optical             ext2    noauto,rw,user  0 0
none               /proc                    proc    defaults        0 0


Then mkfs'ed a blank disc as follows...

[root@localhost me]# /sbin/mkfs -t ext2 /dev/sdb

mke2fs 1.10, 24-Apr-97 for EXT2 FS 0.5b, 95/08/09 /dev/sdb is entire
device, not just one partition! Proceed anyway? (y,n) y
Linux ext2 filesystem format
Filesystem label=
118320 inodes, 472448 blocks
23622 blocks (5.00%) reserved for the super user First data block=1
Block size=1024 (log=0)
Fragment size=1024 (log=0)
58 block groups
8192 blocks per group, 8192 fragments per group 2040 inodes per group
Superblock backups stored on blocks:
8193, 16385, 24577, 32769, 40961, 49153, 57345, 65537, 73729, 81921,
90113, 98305, 106497, 114689, 122881, 131073, 139265,
147457,
155649, 163841, 172033, 180225, 188417, 196609, 204801,
212993, 221185,
229377, 237569, 245761, 253953, 262145, 270337, 278529,
286721, 294913,
303105, 311297, 319489, 327681, 335873, 344065, 352257,
360449, 368641,
376833, 385025, 393217, 401409, 409601, 417793, 425985,
434177, 442369,
450561, 458753, 466945

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

```
rebooted...

mounted the drive...

I've since then edited the fstab, adding the following mount-point...

/dev/sdb           /mnt/dostical          msdos   noauto,rw,user 0 0

I can now mount ext2 or dos formatted optical carts by mounting either
optical or dostical.
```

4.4 FUJITSU MCC3064AP, DYNAMO 640AI − Guido Brunner

```
Dear Skip,

hoping that this is interesting for other Linux-Users, I want to tell You
about my experiences
with optical disks under Linux:
I use an internal 640 MB MO-Drive with IDE-Interface from Fujitsu with
Linux-Kernel 2.2.x and
in the meantime it works fine. At Germany this drive is sold as DYNAMO 640
AI but according
to it's firmware it is a MCC 3064 AP.
Booting kernel 2.2.x the drive is detected like "hdx: FUJITSU MCC3064AP,
ATAPI OPTICAL drive". No driver is loaded, as there still is no ATAPI-
driver for optical disks. Older kernels
(2.0.x) do not detect the drive correctly and surely need some patches.
To use the drive I need kernel support for SCSI-emulation. So I compile
this (ide-scsi.o)
as a module together with the SCSI-disk-support (sd_mod.o). Making a
"modprobe ide-scsi",
the drive shows up in /proc/scsi/scsi. If it isn't done by kerneld I have
to make
"modprobe sd_mod" to be ready to mount the preformatted MO-disk.
If I want to use the disks with Dos/Windows, I use the Dos/Windows-tools
for formatting. I tried
mkdosfs under Linux too, but then most files on the disk seemed to be
corrupted for Dos.
They were still o.k. for Linux and could be restored without problems. With
the Dos-tools I
prefer the Superfloppy-Format as this can be used with most
operating-systems and it is slightly
faster in comparison to a partitioned disk.  This disks can be mounted like
other
Windows-Disks (e.g. "mount -t vfat  /dev/sdx /mountpoint").
Disks for Linux should be in ext2-Format. The 640 MB disks are hardsectored
with
2048 bytes/sector (smaller media aren't.) . This is no problem for kernel
2.2.x, but fdisk and mke2fs do not agree in how to manage this geometry. So
I don't use fdisk anyway and format
the disks with "mke2fs -b 2048 /dev/sdx". I have to tell mke2fs about the
2048 bytes/sector with
the "-b"-option, otherwise the format will fail. Mke2fs than asks to really
do his job, as it has do
format the whole disk not a single partion and I answer with "y".
Now the disk can be mounted with "mount -t ext2 /dev/sdx /mountpoint",
which gives a warning
in /var/log/messages about a nonexisting partion-table. This is o.k. as
fdisk wasn't used and
I can now use the disk. The MO-Disks are slow, but the most reliable media
```

```
available.
Smaller disks (230 MB) are hardsectored for 512 bytes/sector and can be
partioned with fdisk.
before formatting. This should be true for the 512 MB disks, but I didn't
test it.

Best regards and thanks for Your support for Linux,

Guido Brunner
```

4.5 Panasonic LF−7010 − Donald Kerns

```
Dear Skip,

I recently aquired a LF-7010 for a project.

My experience getting it up for Linux under ext2 mirrors what you
already have.

The msdos and vfat file systems also worked.

The project I was working on was based on a SunOS/Solaris formatted MO
disk. While it *should* have worked under the ufs file system, using
Redhat 5.2 and the stock 3.0.36 kernel it didn't.

I got, installed, debugged and compiled the u2fs into the kernel and it
DID mount the SunOS/Solaris MO disk.

Please write if you need/want additional details.

-Donald

 Donald Kerns <dkerns@cruzio.com>
```

4.6 PIONEER DE−C7001 − Paolo Droghetti

```
Hi Skip,


following your request of info from everyone else who is playing with
optical disks under Linux, here I am.

I'm an happy user of a PIONEER DE-C7001, mounted in my linux box
controlled by a PCI SYMBIOS (NCR) 53c815 SCSI controller and running
slackware 4.
After having set the jumpers in the correct way ( all the technical docs
are still available in the Pioneer web pages) and configured the kernel
for SCSI support ( generic SCSI support and support for SCSI disks), it
works.
Here you have the boot details:

sym53c8xx: at PCI bus 0, device 8, function 0
sym53c8xx: not initializing, device not supported
ncr53c8xx: at PCI bus 0, device 8, function 0
ncr53c8xx: 53c815 detected
ncr53c815-0: rev=0x04, base=0xe8000000, io_port=0xe000, irq=11
ncr53c815-0: ID 7, Fast-10, Parity Checking
ncr53c815-0: restart (scsi reset).
scsi0 : ncr53c8xx - version 3.2a-2
```

```
scsi1 : SCSI host adapter emulation for IDE ATAPI devices
scsi : 2 hosts.
Vendor: PIONEER   Model: DE−C7001          Rev: 0500
Type:   Direct−Access                      ANSI SCSI revision: 01 CCS
Detected scsi removable disk sda at scsi0, channel 0, id 1, lun 0
Vendor: ARCHIVE   Model: VIPER 150  21247  Rev: −005
Type:   Sequential−Access                  ANSI SCSI revision: 01
Detected scsi tape st0 at scsi0, channel 0, id 2, lun 0
Vendor: PHILIPS   Model: CDD3600 CD−R/RW   Rev: 2.00
Type:   CD−ROM                             ANSI SCSI revision: 02
Detected scsi CD−ROM sr0 at scsi0, channel 0, id 6, lun 0
scsi : detected 1 SCSI tape 1 SCSI cdrom 1 SCSI disk total.
ncr53c815−0−<6,*>: FAST−10 SCSI 10.0 MB/s (100 ns, offset 8)
sr0: scsi3−mmc drive: 2x/6x writer cd/rw xa/form2 cdda tray
Uniform CDROM driver Revision: 2.55
ncr53c815−0−<1,*>: FAST−5 SCSI 5.0 MB/s (200 ns, offset 8)
sda : READ CAPACITY failed.
sda : status = 1, message = 00, host = 0, driver = 28
sda : extended sense code = 2
sda : block size assumed to be 512 bytes, disk size 1GB.
PPP: version 2.3.7 (demand dialling)
TCP compression code copyright 1989 Regents of the University of
California
PPP line discipline registered.
3c59x.c:v0.99H 11/17/98 Donald Becker
http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html
Partition check:
sda:scsidisk I/O error: dev 08:00, sector 0
unable to read partition table
hda: hda1 hda2 hda3
```

Once the drive has been recognized by the system, I inserted a new disk and I created a partition on it with:

```
# fdisk /dev/sdanew file system on it :
```

After this step I created a new file system :

```
# mkfs −t ext2 /dev/sda1

Linux ext2 filesystem format
Filesystem label=
79872 inodes, 318448 blocks
15922 blocks (5.00%) reserved for the super user
First data block=1
Block size=1024 (log=0)
Fragment size=1024 (log=0)
39 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
8193, 16385, 24577, 32769, 40961, 49153, 57345, 65537, 73729, 81921,
90113, 98305, 106497, 114689, 122881, 131073, 139265, 147457, 155649,
163841, 172033, 180225, 188417, 196609, 204801, 212993, 221185, 229377,

237569, 245761, 253953, 262145, 270337, 278529, 286721, 294913, 303105,

311297

Writing inode tables:  0/39 ..... 38/39 done
Writing superblocks and filesystem accounting information: done
```

```
and then the disk is usable and accessible as a normal SCSI disk.
I don't have modified the /etc/fstab and /etc/mtab files, because I
prefer to mount the disk manually when needed.


Now I need your help. In my office, I have 3 LMS ( Philips ) LF4500
rapid changer that are completely not used. The LF4500 holds 5 12" WORM
optical disk. double sided 5,6 GB capacity. They were originally used
with a SPARC 1+ controller and a software written for solaris 1.x. The
controllers and the original sw are completely gone ( nobody knows where
they are !). I will try to connect one of thisLF4500 to my linux box. I
will let you know the results but I think I'll be able to see only the
drive but not the disk exchange device. Could please help in find out
more info/SW in order to fully drive this rapid changer ?

:-)  I work for Philips Medical Systems but I'm not able to find more
info on this formerly Philips product.

Regards

Paolo Droghetti <paolo.droghetti@philips.com>
```

4.7 Fujitsu MCD3130SS − Harald Husemann

```
Hi Skip,

I've used your 'Linux-Optical Disk HOWTO' to setup our magneto-optical
drive.
You mentioned somewhere in the HOWTO that you'd like to receive
additional informations, and since I've used a drive which was not
included, I'd like to tell you about it. Hope it can help someone!

Used hardware:

INTEL Pentium 90
SCSI-Controller ADAPTEC 2940
MO-Drive Fujitsu MCD3130SS (1.3 GB Capacity)

Software:

S.u.S.E.-LINUX 6.1, Kernel-Version 2.2.5

There is no "native" driver for the 2940AU, so I used the "aic7xxx"
which I load as a module during bootup (I didn't want to compile a new
kernel, because I need many other features, and expect of the MO-Drive,
everything worked fine before. So, why "change a running system"?!)

I can mount the MO-Disk, no matter what filesystem is used, entirely.
In addition to that, I set up autofs to ease my work:

in /etc/auto.misc, I added the line:

==================/SNIP/===============
/misc/mo-disk -fs auto /dev/sda1
==================/SNAP/==============

With that, I even don't need to mount the drive, I can access it
whenever I want, no matter what filesystem is used (tested with MSDOS
```

```
and ext2-fs)

Finally, I used SAMBA to export the drive to our WIN95-Clients with the
following inserted in /etc/smb.conf:

=======================/SNIP/======================
[mo-disk]
path = /misc/mo-disk
public = yes
writeable = yes ;write-only can of course still be controlled by
flipping the
;write-protect-switch at the MO-Disk!
readable = yes
browseable = yes
=======================/SNAP/=======================

(for further details abt. SAMBA, refer to the excellent HOWTO)

Now, every WIN-Client can use the MO-Drive as if it was a local hdd,
with one (minor) caveat:

When you map the exported SAMBA-Drive to a drive letter in WIN Explorer,
it's impossible to umount it under LINUX! Everytime you try, you get a
"device busy"...
So, unfortunately I can't map the drive during startup in WIN95, but I
think with some hacking in the SAMBA-Code this problem could be
solved...
I don't have the time at the moment, but perhaps somewhat later I will
try to "dig into the code" to do the hack.

Of course you can include my e-amil-address in the HOWTO, but please use
my private one:

dh9dat@cityweb.de instead of ds@leiterplattentechnik.de!

with regards,

Harald Husemann



LINUX - the operating system for people whose IQ is greater than 98...
```

Harald Husemann <ds@www.leiterplattentechnik.de>

4.8 Ricoh RO−5031E − Jeremy Hosford

```
Hi,

Just stumbled across your page on Tucows (dated Dec '98 so I hope this
still reaches you). You asked if anyone had any experience with optical
storage etc. under Linux, which I have, so here it is!

I worked for Ericsson (UK) Ltd. and some of their telephone switches use
optical media for system backups. I have used these optical drives on
i386 Linux boxes for some years now, with no problems whatsoever.

The units in question are the Ricoh RO-5031E (scsi) and it's bigger
brother, which unfortunately I cannot remember that name of (also Ricoh
+ scsi). The RO-5031E is a full-height, 5.25in magneto-optical drive
```

```
that uses 650Mb disk cartridges (325Mb per side), such as Sony's
EDM-650B. The other drive has similar spec but can use both 1.3Gb and
650Mb disks. Ricoh's website may have more on these drives, but they're
quite long in the tooth now and may not feature anymore.

Usage was very simple - The drives were treated almost as scsi fixed
disks. Pop a new disk in, use fdisk to create your filesystem (I've
tried both ext2 & msdos) then format with mkfs. That's it!

The one weird thing I did find was that a RedHat 6.x system (2.2.x
kernel) would not read a filesystem that had been created on an old
Slackware (2.0.x kernel) system, and vice versa. Other than that, 100
million re-writes... thankyou very much!!!

All the best // Jem

P.S. Please feel free to include my email if I've been of any help.

 jem <jem@monty.ericsson.se>
```

4.9 Maxoptix T6−5200 − Donovan Allen

I have used a Maxoptix T6−5200 with re−writable MO media without any problems. Donovan Allen admin@robot−factory.net

4.10 Maxoptix TMT3−1300 Magneto Optical drive. − Peter Knaggs

Maxoptix TMT3−1300 Magneto Optical drive. Accepts 1Gb and 1.3Gb magneto−optical read/write cartridges, these are double sided, so half the capacity is on each side, and the cartridge needs to be ejected to access the opposing side.

When configuring a Maxoptix drive for Linux, it should be configured such that the Removable Media Report Disable switch is OFF (the dip switch bank S2, switch 3 is OFF, i.e. in down position).

When configuring a Maxoptix drive for Linux, it should NOT be configured such that Removable Media Report Disable is switched ON (the dip switch bank S2, switch 3 is ON, i.e. in up position).

Setting this switch ON will set the RMB (removable media bit) to 0. This would indicate to Linux that the media is NOT removable, but Linux of course still allows one to eject it using the command eject /dev/sda or by pressing the invitingly large button on the drive itself. This could have the consequence of accidentally corrupting any of the good data stored on cartridges inserted subsequently, since Linux has still cached the directory information of the previous disk.

When the RMB is 0, even after the cartridge is ejected, it is possible to perform the mount command, and have it 'succeed', since Linux has still got the directory structure of the previous disk cached in its buffers, which have not been flushed. One way to force Linux to flush its buffers in this situation, is to do the following sequence: eject /dev/sda (do NOT insert a new cartridge before performing the next two steps)

```
mount /dev/sda5 /mnt/max
umount /mnt/max

Note that this will show the following on the console (Ctrl+Alt+F10):
Dec 14 19:32:14 kernel: SCSI disk error : host 1 channel 0 id 6 lun 0 return code = 28000000
```

```
Dec 14 19:32:14 kernel: [valid=0] Info fld=0x0, Current sd08:05: sense key Not Ready
Dec 14 19:32:14 kernel: Additional sense indicates Medium not present
Dec 14 19:32:14 kernel: scsidisk I/O error: dev 08:05, sector 2
Dec 14 19:32:14 kernel: SCSI disk error : host 1 channel 0 id 6 lun 0 return code = 28000000
Dec 14 19:32:14 kernel: [valid=0] Info fld=0x0, Current sd08:05: sense key Not Ready
Dec 14 19:32:14 kernel: Additional sense indicates Medium not present
Dec 14 19:32:14 kernel: scsidisk I/O error: dev 08:05, sector 2

Now we can insert the new disk, and the new disk's directory structure
will appear correctly as expected. This method is very error-prone, and not
recommended as it is very easy to forget to perform the 'dummy' mount before
inserting a disk, with the consequence of wiping out good data on the new disk.

The reason this method works is that Linux normally only calls invalidate_buffers()
(see the file sd.c in the routine check_media_change()) if the RMB is set to 1,
and the above sequence forces Linux to call invalidate_buffers() once it notices
that it can't mount the filesystem.
Performing the 'dummy' mount/unmount after ejecting forces Linux to call invalidate_buffers().

For Linux, to avoid using the above workaround, we should always have
the Removable Media Report Disable switch OFF (dip switch bank S2, switch 3, OFF, i.e. in down position).

When the Removable Media Report Disable switch is correctly set to OFF,
then attempting to mount the drive when the cartridge
is not present will show the following on the console (Ctrl+Alt+F10):

Dec 14 21:35:16 kernel: sda : READ CAPACITY failed.
Dec 14 21:35:16 kernel: sda : status = 0, message = 00, host = 0, driver = 28
Dec 14 21:35:16 kernel: sda : extended sense code = 2
Dec 14 21:35:16 kernel: sda : block size assumed to be 512 bytes, disk size 1GB.
Dec 14 21:35:16 kernel:  sda:scsidisk I/O error: dev 08:00, sector 0
Dec 14 21:35:16 kernel:  unable to read partition table
Dec 14 21:35:16 kernel: sda : READ CAPACITY failed.
Dec 14 21:35:16 kernel: sda : status = 0, message = 00, host = 0, driver = 28
Dec 14 21:35:16 kernel: sda : extended sense code = 2
Dec 14 21:35:16 kernel: sda : block size assumed to be 512 bytes, disk size 1GB.
Dec 14 21:35:16 kernel:  sda:scsidisk I/O error: dev 08:00, sector 0
Dec 14 21:35:16 kernel:  unable to read partition table

--------------------------------------------------------------------------------

When using the drive by means of an Adaptec PCMCIA card in slot 0,
if disk is in drive, when the command:

# cardctl insert 0

is performed, we see the following message on the console (Ctrl+Alt+F10):

Dec 14 19:05:31 kernel: aha152x: processing commandline: ok
Dec 14 19:05:31 kernel: aha152x: BIOS test: passed, detected 1 controller(s)
Dec 14 19:05:31 kernel: aha152x0: vital data: PORTBASE=0x140, IRQ=3, SCSI ID=7, reconnect=enabled, parity=enabled, synchronous=disabled, delay=100, extended translation=disabled
Dec 14 19:05:31 kernel: aha152x: trying software interrupt, ok.
Dec 14 19:05:31 kernel: scsi1 : Adaptec 152x SCSI driver; $Revision: 1.7 $
Dec 14 19:05:31 kernel: scsi : 2 hosts.
Dec 14 19:05:32 kernel:   Vendor: Maxoptix  Model: T3-1304           Rev: 1.1c
Dec 14 19:05:32 kernel:   Type:   Direct-Access                      ANSI SCSI revision: 02
Dec 14 19:05:32 kernel: Detected scsi disk sda at scsi1, channel 0, id 6, lun 0
Dec 14 19:05:32 kernel: SCSI device sda: hdwr sector= 512 bytes. Sectors= 904995 [441 MB] [0.4 GB]
Dec 14 19:05:33 kernel:  sda: sda1 < sda5 >


If no disk is inserted when booting Linux, get the following message on the console:
```

```
Dec 14 18:55:23 kernel: Detected scsi disk sda at scsi1, channel 0, id 6, lun 0
Dec 14 18:55:24 kernel: sda: Spinning up disk....<7>ROM image dump:
Dec 14 18:57:02 kernel: ..................................................................................................not responding...
Dec 14 18:57:02 kernel: sda : READ CAPACITY failed.
Dec 14 18:57:02 kernel: sda : status = 0, message = 00, host = 0, driver = 28
Dec 14 18:57:02 kernel: sda : extended sense code = 2
Dec 14 18:57:02 kernel: sda : block size assumed to be 512 bytes, disk size 1GB.
Dec 14 18:57:02 kernel:  sda:SCSI disk error : host 1 channel 0 id 6 lun 0 return code = 28000000
Dec 14 18:57:02 kernel: [valid=0] Info fld=0x0, Current sd08:00: sense key Not Ready
Dec 14 18:57:02 kernel: Additional sense indicates Medium not present
Dec 14 18:57:02 kernel: scsidisk I/O error: dev 08:00, sector 0
Dec 14 18:57:02 kernel:  unable to read partition table
Dec 14 18:57:03 kernel: scsi2 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI) 5.1.25/3.2.4
Dec 14 18:57:03 kernel:        <Adaptec PCMCIA SCSI controller>
Dec 14 18:57:03 kernel: scsi : 3 hosts.
Dec 14 18:57:08 kernel: scsi : 2 hosts.

--------------------------------------------------------------------------------

When using the disks which come pre-formatted with 1024 bytes per sector,
it's important to use the -b 1024 flag with fdisk, otherwise
the partitioning isn't correctly written by the hardware,
and mke2fs hangs, so that the system cannot be shutdown cleanly.

When making the filesystem using mke2fs, don't use the defaults,
since depending on the media size, the block size which is chosen by mke2fs
might only be 1024, which is way too small. We want to always use 4096 as the
block size, otherwise writing to the disk becomes very slow indeed.

So for a disk with 1024-bytes per sector, the sequence of commands would be:

# fdisk -b 1024 /dev/sda
# mke2fs -b 4096 -m 0 /dev/sda5
# e2fsck -f -B 4096 /dev/sda5 -b 98304   (-b 98304 uses an alternate superblock)

# mke2fs -b 4096 -m 0 /dev/sda5
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
79680 inodes, 159216 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
5 block groups
32768 blocks per group, 32768 fragments per group
15936 inodes per group
Superblock backups stored on blocks:
32768, 98304

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

# time (cp ~guest/kernel/linux-2.2.15.tar.gz .;sync;sync;sync)

real    0m28.411s
user    0m0.010s
sys     0m2.590s

# time (cp ~guest/kernel/*.gz .;sync;sync;sync)
```

```
real    3m54.046s
user    0m0.060s
sys     0m2.910s

-----------------------------------------------------------------------
For 512-byte sectors, the output of mke2fs is as follows for a 1G disk:

# mke2fs -m 0 /dev/sda5
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
112896 inodes, 451552 blocks
0 blocks (0.00%) reserved for the super user
First data block=1
56 block groups
8192 blocks per group, 8192 fragments per group
2016 inodes per group
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

# time (cp ~guest/kernel/*.gz .;sync;sync;sync)

real    3m54.046s
user    0m0.060s
sys     0m2.910s

--------------------------------------------------------------------------------
Note the difference in storage of a disk formatted for
1024 bytes per sectos, and 512 bytes per sector,
for the same 1.3GB capacity claimed on the disk label.
It is about 9.5% more for the 1024 format.

1.3 GB 512 bytes/sector:
df -k .
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               572436       740    571696   0% /mnt/max

1.3 GB 1024 bytes/sector, using 4096 block size in mke2fs:
df -k .
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               626840    112208    514632  18% /mnt/max


--------------------------------------------------------------------------------
This is the output of fdisk partitioning created by Win98 on the 1.3GB 1024 bytes/sector disk:

Disk /dev/sda: 64 heads, 32 sectors, 311 cylinders
Units = cylinders of 2048 * 1024 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/sda1   ?    937477   1203315 544437093   20  Unknown
Partition 1 has different physical/logical beginnings (non-Linux?):
phys=(356, 97, 46) logical=(937476, 3, 15)
Partition 1 has different physical/logical endings:
phys=(357, 116, 40) logical=(1203314, 30, 19)
Partition 1 does not end on cylinder boundary:
phys=(357, 116, 40) should be (357, 63, 32)
```

```
/dev/sda2   ?    649505    912677 538976288   6b  Unknown
Partition 2 has different physical/logical beginnings (non-Linux?):
phys=(288, 110, 57) logical=(649504, 0, 11)
Partition 2 has different physical/logical endings:
phys=(269, 101, 57) logical=(912676, 1, 10)
Partition 2 does not end on cylinder boundary:
phys=(269, 101, 57) should be (269, 63, 32)
/dev/sda3   ?    263179    945973 1398362912   53  OnTrack DM6 Aux3
Partition 3 has different physical/logical beginnings (non-Linux?):
phys=(345, 32, 19) logical=(263178, 26, 16)
Partition 3 has different physical/logical endings:
phys=(324, 77, 19) logical=(945972, 51, 15)
Partition 3 does not end on cylinder boundary:
phys=(324, 77, 19) should be (324, 63, 32)
/dev/sda4   *    680971    680981     21337   49  Unknown
Partition 4 has different physical/logical beginnings (non-Linux?):
phys=(87, 1, 0) logical=(680970, 34, 16)
Partition 4 has different physical/logical endings:
phys=(335, 78, 2) logical=(680980, 61, 8)
Partition 4 does not end on cylinder boundary:
phys=(335, 78, 2) should be (335, 63, 32)

Command (m for help): q
```

Similarly, this is the output of fdisk partitioning created by Win98 on the 1GB 512 bytes/sector disk:

```
# fdisk −b 512 /dev/sda

Command (m for help): p

Disk /dev/sda: 64 heads, 32 sectors, 441 cylinders
Units = cylinders of 2048 * 512 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/sda1   ?    937477   1203315 272218546+  20  Unknown
Partition 1 has different physical/logical beginnings (non-Linux?):
phys=(356, 97, 46) logical=(937476, 3, 15)
Partition 1 has different physical/logical endings:
phys=(357, 116, 40) logical=(1203314, 30, 19)
Partition 1 does not end on cylinder boundary:
phys=(357, 116, 40) should be (357, 63, 32)
/dev/sda2   ?    649505    912677 269488144   6b  Unknown
Partition 2 has different physical/logical beginnings (non-Linux?):
phys=(288, 110, 57) logical=(649504, 0, 11)
Partition 2 has different physical/logical endings:
phys=(269, 101, 57) logical=(912676, 1, 10)
Partition 2 does not end on cylinder boundary:
phys=(269, 101, 57) should be (269, 63, 32)
/dev/sda3   ?    263179    945973 699181456   53  OnTrack DM6 Aux3
Partition 3 has different physical/logical beginnings (non-Linux?):
phys=(345, 32, 19) logical=(263178, 26, 16)
Partition 3 has different physical/logical endings:
phys=(324, 77, 19) logical=(945972, 51, 15)
Partition 3 does not end on cylinder boundary:
phys=(324, 77, 19) should be (324, 63, 32)
/dev/sda4   *    680971    680981     10668+  49  Unknown
Partition 4 has different physical/logical beginnings (non-Linux?):
phys=(87, 1, 0) logical=(680970, 34, 16)
Partition 4 has different physical/logical endings:
phys=(335, 78, 2) logical=(680980, 61, 8)
Partition 4 does not end on cylinder boundary:
phys=(335, 78, 2) should be (335, 63, 32)
```

```
Question: Which partition is the data in, and how do we mount it?
None of the start/end values make any sense to Linux.
--------------------------------------------------------------------------------
Comparison of media sizes and speeds:
=====================================

Using the following files:

# ls -l ~guest/kernel/*gz
-rw-r-----   1 guest  users    16371764 May 24  2000 /home/guest/kernel/linux-2.2.15.tar.gz
-rw-rw-rw-   1 guest  users    17106471 Jun 15  2000 /home/guest/kernel/linux-2.2.16.tar.gz
-rw-r--r--   1 guest  users    20881782 May 25  2000 /home/guest/kernel/linux-2.3.99-pre9.tar.gz
-rw-rw-rw-   1 guest  users    21085275 Jun 16  2000 /home/guest/kernel/linux-2.4.0-test1.tar.gz
-rw-r--r--   1 guest  users    22888582 Oct  3 08:36 /home/guest/kernel/linux-2.4.0-test9.tar.gz

The tests consisted of simply copying the above files to the drive.
Sometimes the media is new, sometimes previously written to.
It seems that new media is slightly faster to write to.
%------------------------------------------------------------------------------%

On media identified as follows:
P/N 2015382-0010 Max-GL (Optical Glass) Jukebox Certified Rewritable 1.3GB 512

# fdisk /dev/sda

Command (m for help): p

Disk /dev/sda: 64 heads, 32 sectors, 568 cylinders
Units = cylinders of 2048 * 512 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/sda1             1       568    581616    5  Extended
/dev/sda5             1       568    581600   83  Linux

# mke2fs -m 0 -b 4096 /dev/sda5
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
72800 inodes, 145400 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
5 block groups
32768 blocks per group, 32768 fragments per group
14560 inodes per group
Superblock backups stored on blocks:
32768, 98304

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               572436        20    572416   0% /mnt/max

# time (cp ~guest/kernel/linux-2.2.15.tar.gz .;sync;sync;sync)

real    0m29.099s
user    0m0.010s
sys     0m1.430s
```

```
# time (cp ~guest/kernel/*.gz .;sync;sync;sync)

real    4m18.446s
user    0m0.010s
sys     0m3.880s

%-------------------------------------------------------------------------------%

On media identified as follows:
P/N 1015386RW Max-GL (Optical Glass) Jukebox Certified Rewritable 1GB 512

fdisk shows:
Disk /dev/sda: 64 heads, 32 sectors, 441 cylinders
Units = cylinders of 2048 * 512 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/sda1             1       441    451568    5  Extended
/dev/sda5             1       441    451552   83  Linux


# mke2fs -m 0 -b 4096 /dev/sda5
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
112896 inodes, 112888 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
4 block groups
32768 blocks per group, 32768 fragments per group
28224 inodes per group
Superblock backups stored on blocks:
32768, 98304

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               437384        20    437364   0% /mnt/max

# time (cp ~guest/kernel/linux-2.2.15.tar.gz .;sync;sync;sync)

real    0m34.220s
user    0m0.100s
sys     0m5.610s

# time (cp ~guest/kernel/*.gz .;sync;sync;sync)

real    4m8.846s
user    0m0.280s
sys     0m28.500s

%-------------------------------------------------------------------------------%

On media identified as follows: (obtained from eBay:Borismcbin)
MaxEP Rewritable 1GB 512 (Tahiti P/N 1015387-0040)

Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               437384        20    437364   0% /mnt/max
```

```
# time (cp ~guest/kernel/linux-2.2.15.tar.gz .;sync;sync;sync)

real    0m30.321s
user    0m0.010s
sys     0m1.340s

# time (cp ~guest/kernel/*.gz .;sync;sync;sync)

real    3m32.851s
user    0m0.010s
sys     0m4.340s

%--------------------------------------------------------------------------------%

On media identified as follows: (obtained from eBay:Surpuluseller)
MaxEP Rewritable 1.2GB 512 (P/N: PN2015383RW)
"Maxoptix PN2015383RW FORMATTED ERASABLE OPTICAL CARTRIDGE 1.2 GIGABYTE 512 BYTES/SECTOR"

# fdisk /dev/sda

Command (m for help): p

Disk /dev/sda: 64 heads, 32 sectors, 568 cylinders
Units = cylinders of 2048 * 512 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/sda1             1       568    581616    5  Extended
/dev/sda5             1       568    581600   83  Linux


# mke2fs -m 0 -b 4096 /dev/sda5
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
72800 inodes, 145400 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
5 block groups
32768 blocks per group, 32768 fragments per group
14560 inodes per group
Superblock backups stored on blocks:
32768, 98304

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               572436        20    572416   0% /mnt/max

# time (cp ~guest/kernel/linux-2.2.15.tar.gz .;sync;sync;sync)

real    0m28.979s
user    0m0.000s
sys     0m2.600s

# time (cp ~guest/kernel/*.gz .;sync;sync;sync)

real    4m0.486s
user    0m0.000s
sys     0m1.400s
```

```
%-----------------------------------------------------------------------------%

On media identified as follows:
HEWLLET PACKARD REWRITABLE OPTICAL DISK (Type R/W - CC Format)
1.3 Gbytes 1024 Byte/Sector
Reorder No: 92280T
Made in Japan.

# fdisk -b 1024 /dev/sda

Command (m for help): p

Disk /dev/sda: 64 heads, 32 sectors, 311 cylinders
Units = cylinders of 2048 * 1024 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/sda1             1       311    636896    5  Extended
/dev/sda5             1       311    636864   83  Linux

Command (m for help): q

Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda5               626840        20    626820   0% /mnt/max

# mke2fs -m 0 -b 4096 /dev/sda5

mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
79680 inodes, 159216 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
5 block groups
32768 blocks per group, 32768 fragments per group
15936 inodes per group
Superblock backups stored on blocks:
32768, 98304

Writing inode tables: done
Writing superblocks and filesystem accounting information: done


# time (cp ~guest/kernel/linux-2.2.15.tar.gz .;sync;sync;sync)

real    0m28.411s
user    0m0.020s
sys     0m1.570s

# time (cp ~guest/kernel/*.gz .;sync;sync;sync)

real    4m19.854s
user    0m0.010s
sys     0m2.350s

%-----------------------------------------------------------------------------%
```

Peter Knaggs <Peter.Knaggs@oracle.com>

4.9 Maxoptix T6−5200 − Donovan Allen

4.11 Magneto Optical Information, IDE/ATAPI and FAT/VFAT info − Alexander Voropay

Alexander supplied me with some very informative points. He has ask me to pass that information on to you. Without fully understanding all areas that he knows, I am attempting to pass it on to you as best I can;

Type of 3.5" MO drives

There are two kind of 3.5" Magneto Optical drives : DynamMO and GigaMO

DynaMO : 128Mb, 230Mb, 540Mb and 640Mb (this drive has 2048Kb/sector)

GigaMO : 1.3Gb, 2.3Gb (2048Kb/sector)

MO Cartridges specifications :

- Sony's GigaMO Media Specifications
- Professional Media Specifications

MO drives with IDE/ATAPI interface

(Sony, Fujitsu) has a models with different interfaces : SCSI, IDE/ATAPI, USB, FireWire, PCMCIA. I think, we should expand your HOWTO to this models.

For example, to work with IDE/ATAPI interface you should install "ide−scsi" kernel module :

```
# modprobe ide-scsi
# dmesg
....
sscsi1 : SCSI host adapter emulation for IDE ATAPI devices
Vendor: IomintSA  Model: MCE3130AP-MO1300  Rev: 0011
Type:   Optical Device                    ANSI SCSI revision: 02
Attached scsi removable disk sdb at scsi1, channel 0, id 1, lun 0 SCSI
device sdb: 605846 2048-byte hdwr sectors (1241 MB)
sdb: Write Protect is off
sdb:
```

Accessing FAT and VFAT MO

You could use "mtool" package to access a FAT/VFAT on MO. Mtools page to FAT and VFAT File systems

```
First of all, you should define a new drive M: in
/etc/mtools.conf
=====
drive m: file="/dev/sdb"
=====

# minfo m:
# mformat m:
# mdir m:


You should use -I key to format MO media
```

```
with mkfs.vfat :
# mkfs.vfat −I −v /dev/sdb


Unfortunately, Linux kernel uses bogus MO drive
geometry even for SCSI drives. It makes sense only
for BPB FAT filesystem (ext2fs does not depend on drive
geometry). To compatibility with Windows system you
should use real geometry (1 Head) while formatting.
(See cartridge specifications for real geometry.)

For example for 230Mb DynaMO you should define in
/etc/mtools.conf
=====
drive m:
file="/dev/sdb"
cylinders=17853 heads=1 sectors=25
mformat_only
=====

# mformat m:
...
## minfo m:

Alexander Voropay
<alec@vmb−service.ru>
```

---

5. Optical jukeboxes

I have no experience with optical jukeboxes with Linux!!!! I have had experiences with Optical jukeboxes under HP−UX. In this setup the the jukebox had a SCSI address of it's own. Each slot in the jukebox had an associated LUN number. A device name was assigned for each disk slot A side and B side. The mount command was run against the appropriate device name. I had a jukebox with just one drive and 16 optical disk slots − 20 Gig. I thought it was going to be a real hassle to write a disk mount manager to share this drive among users until I discovered you can mount as many disk as you want and the jukebox driver takes care of arbitration − what a nice feature. Granted, you only want archive type data here and your overall system configuration to be such that not too many processes will be accessing the jukebox at the same time. The disk spin down, carriage load, carriage move, carriage unload, carriage move to the next disk, carriage next disk load, carriage move, optical drive load, and spin up takes about 12 seconds − "seek−from−hell".

5.1 Maxoptix 520 − Zed Shaw

shawz@imap1.asu.edu

Zed's Original E−Mail − Feb 13 1998

```
Hi,

I was reading your howto (a life saver, thanks) and I was wondering what
kind of jukebox you were running?  I have a Maxoptix 520 Jukebox (20
disks at 2.6G each, nice!) and I would like to connect it to a Linux box
and serve the drives up to my users, but I'm having problems accessing
the individual drives.   Currently I can only access the two drives and
something called MAXLYB which I think is a controller device of some
sort.
```

```
Basically, I'm wondering if the jukebox you had was the same or similar
and how you set it up.  I know that you did it under HP-UX, but any help
right now would be nice.  Hey, I'll even let you log into my linux
server if you want to take a look at the jukebox and see what it does.
You can't beat 52Gig of storage!

Anyway, I'd really appreciate your help.


Zed A. Shaw
Application Systems Analyst
Arizona State University
```

Correspondence with Zed on Mon, 16 Feb 1998:

```
> It sounds like your Maxoptix 520 is a jukebox with two physical disk.
Yep, that's the one.

>
> All jukeboxes have a carriage controller. This is probably your MAXLYB
> device.
> ...

What I've come to find out is that Maxoptix is pretty stingy when it
comes to drivers.  Apparently, they don't make driver software for any of
their Jukebox carriage controller interfaces!  I don't know how some of
these companies stay in business.  I'm going to pester them again soon,
but you are right, this thing will need a carriage controller driver to
operate.  The cool thing is that this MX520 (that's the model number of
the juke) emulates a whole slew of other carriage controllers, so maybe
one of those other guys has a driver.  I'll be looking into that too.


>
> You might want to get a-hold of Maxoptix and see if they have a install
> package for your linux kernel version. If not ask them for the programmers
> specification for the carriage controller and maybe we can write one!
>

Hey, if I can't find any driver software, and I can convince Maxoptix to
give me the specs, I'd be more than glad to write a driver.  I'd could
sure use the help too since I haven't got enough time to do it on my
own.  Also, do you know of anyone else doing this that we might be able
to hack off of?


> Any information you find, let me know and we will roll the information
> into the Optical HOWTO, acknowledgments of course!
>

Sure, but let me get some new information first.  So far things are
looking pretty bleak.

>
> >Basically, I'm wondering if the jukebox you had was the same or similar
> >and how you set it up.  I know that you did it under HP-UX, but any help
> >right now would be nice.  Hey, I'll even let you log into my linux
> >server if you want to take a look at the jukebox and see what it does.
> >You can't beat 52Gig of storage!
>
> Nice. At home I can use PPP to mount my 84 platter HP-UX jukebox.
```

```
> It's slow though - I wish I had it at home.

Oh, I don't have this thing at home.  There's no way I could afford the
$30,000 my boss paid for this thing.  But he's stuck with it and has had
it sitting around collecting dust for a year, so he's letting me play
with it and try to find a use for it.

I'll get back with you when I have some more information.  It should be
sometime this week when I find out if I can get it to work or not.

Zed
```

5.2 Changer Devices − Jon Gerdes

```
Skip

Please find some guff on MO drives and SCSI agony ...

Note that the Changer software mentioned should work for virtually any
changer :-)

Now back to dreaming of cheap highbandwidth inet connections for home
use.  British Telecom really get on my !"£***&

Cheers
Jon Gerdes



Some notes on firing up SCSI changer support for a Magneto Optical jukebox
using "scsi-changer"
==============================================================

LSM entry from distribution used:
Title:          scsi-changer
Version:        0.14
Entered-date:   9 May 1998
Description:    SCSI Media Changer device driver (for the robot mechanism
of MO/CD Jukeboxes, tape libraries, ...).
autofs support included.  This is a BETA version.
Keywords:       scsi jukebox changer driver
Author:         Gerd Knorr <kraxel@cs.tu-berlin.de>
Maintained-by:  Gerd Knorr <kraxel@cs.tu-berlin.de>
Primary-site:   sunsite.unc.edu /pub/Linux/kernel/patches/scsi
22kB scsi-changer-0.14.tar.gz
627  scsi-changer.lsm
Alternate-site: none
Original-site:  none
Platform:       Linux
Copying-policy: GNU GPL
===============================================================
Latest version from here:

http://www.in-berlin.de/User/kraxel/linux.html

Tested on:
HP SurestoreOptical 40fx (1 drive 15x2.4Gb MO disks)
Adaptec 1520B (dodgy ISA 10MBs-1 SCSI II card)
Linux Mandrake 6.1
Kernel 2.2.13 and 2.2.14
```

```
tar -xzvf changer-0.14.tar.gz
read the README

Note that this program should work for most SCSI devices involving a robotic
picker and media slots.

patch kernel:
copy ch-2.2.7.diff to /usr/src/linux (ie kernel source)
patch -p1 < ch-2.2.7.diff

Compile in changer support to kernel:
make xconfig or menuconfig or just config
go to scsi section and tick changer support
put in NFS and automounter (see below) while you are at it.

If your using it as a module then add this to /etc/conf.modules:
alias char-major-86 ch
(no problems found using it as a module - recommended method of use)

run /usr/src/linux/drivers/MAKEDEV.sch (created by .diff) to create the
changer dev entries, one for the changer and one for the reader if you have
more than one reader, you will need more /dev/schx's (see
<src>/Documentation/devices.txt - again added by .diff)

Run make in the changer source directory
I had to copy scsi_ioctl.h to changer source directory and amend the Makefile.
Not sure why, so read and change the Makefile if compiler gives errors about
such a file not found (find /usr/src/linux -name <file_to_find> might be
handy)

copy binaries to /sbin (or /usr/sbin or whatever to taste)

mover load 0
fdisk /dev/sda (create sda1)
mke2fs -b 1024 /dev/sda1 1273011
mover unload 0
mover load 1
fdisk ...
etc

"mover mv d0 d0 1" will rotate disk in drive 1
etc etc

fdisk was OK but mke2fs had problems determining the correct geometry.
the number of blocks came from dmesg report hence specifying everything
explicitly - your milage may vary.  The block size was printed on the disks as
well as from "dmesg"

The above is crying out for a quick script to shuffle through the lot.  mover
without switches will show the contents of the box.

make sure that scsi supports > 1Gb
on Adaptec 1520B - aha152x needs (in /etc/lilo.conf):
append="aha152x=0x340,10,7,1,1,0,100,1"
check source for meaning of parameters.

I also had a weird problem where the adaptec driver appeared to caused Lilo
to forget much of the append= line - pretty esoteric but it may afflict
someone else.  I had a Frame Buffer init string in there as well and after
swapping these around (ie SCSI bit first then the video bit it worked)

The adapter BIOS had to be adjusted slightly in the CTRL-A menu on boot to
```

```
also enable extended translation.

If the geometry is wrong then mke2fs will attempt to access incorrect
addresses.  This causes the kernel to go absolutly mad, printing lots of
errors to the root console and filling up the system log.  I had to use
shift-alt-sysreq-b to re-boot after synching disks, when I got it wrong :-(

My console was totally unusable with hundreds of messages scrolling up it.  So
if this might happen to you, make sure that the "Magic Syskey" is enabled (see
kernel docs.)  Alternatively do it in X from an xterm, that way you can kill
the job from another one ...

Now dig out the automounter (instructions in README).  The devices are
/dev/sda1 and /dev/sdb1 (or similar) not the /dev/schx jobs.  Remember to
have automounter and nfs support in the kernel.  Attempting to cd /jukebox/0
will get the jukie to dig out the first disk and pop it into a spare drive and
even mount it.

Not sure what the problems I had were actually caused by.  On boot the devices
were found and reported correctly but mke2fs wasn't going to play.  I even
managed to wreck one side of my first MO disk so that even fdisk refuses to
play.  You have been warned <g>  Still I do have a ridiculously large amount
of filesystem space to play with at home.

Incidently, if you put a FAT f/s on a disk and leave it in the drive, then
Win98 can read it, if you happen to dual-boot (pah !)

Check the contents of the .diff file and the source itself - the author has
been pretty thorough with documentation.

Performance is not blistering but beats the heck out of a CDRW.

Finally, thanks to Mr Knorr for giving me an endless file system

Jon Gerdes - 17 January 2000
mailto:jon.gerdes@virgin.net
```

5.3 Changer Devices − Michael Heydenbluth

```
Hi,

I have the ANSI SCSI-II specs. Changer devices are described here.
Basically, changers are independent devices which respond to commands like:
Install disk in slot 7 in drive 2
Remove disk from drive 1 and store it in slot 13
Count all disks in the slots
and so on.
Btw. the SCSI specs make no differences between tape changers, od
jukeboxes, cd changers. These are all changer devices with the same command
set. The jukebox running under HP-UX seems to work differently from the
SCSI specs, where no LUNs for the slots/sides are defined. Three
possibilities:
1. LUNs for the disk sides are defined in SCSI-III specifications (I don't
know because I do not have them)
2. The jukebox and its driver is older than the SCSI specs, so the
manufacturer had to go its own way. This is the way my jukebox works, but
in my case the commands are similar.
3. The manufacturer choosed to ignore any specifications

Long time ago I have tried to make my jukebox work with linux. It worked
with the generic scsi driver /dev/sg*, some lines of C and a handful of
```

```
shell scripts, but I did not have the time (and knowledge) to build a real
kernel module from that.
There is still another problem: One has to write a new filesystem type for
write-once media. The ext2 filesystem can not be used because it writes
superblocks all over the disk and modifies them when data is written to the
disk. Sectors which have been written to are definitively lost. They can be
overwritten but in fact the new data is written to spare blocks and the old
data is discarded. This old data can be retrieved with special commands for
optical drives. There must be some kind of "append-only filesystem" to make
it really useful.

Greeting
Michael Heydenbluth
mh@heywei.de
```

---

6. MO and other media technologies − Gene Cumm

The PCtechGuide Home page has some excellent media history and evolving media technology information.
Gene Cumm − bg18179@binghamton.edu

---

7. Phase Change Optical Technology

7.1 Introduction

Optical Phase Change technology is used to create "In Phase" or "Out of Phase" bits on a special media for phase change writing. The drive uses a LASER of different power levels or LASER intensities to produce this effect. One power level allows the media to flow into a crystalline form while the other creates an "Out of Phase" condition. The crystallized areas reflect the read Lasers beam with a different coefficient of reflectivity than the non−crystallized areas. Thus, data can be read from the disk.

What makes the phase change optical disk special is that it the disk is formatted with concentric cylinders or tracks with each track being sectored much like a magnetic disk or read/write optical disk. The tracks are very close so a lot of data can be stored on a disk. This is different from a CD−ROM in that it gives your system the look and feel of a magnetic disk. CD−ROMs have a spiraling track much like a audio record. Having tracks and sectors alone would not make the phase change drive special from optical disk but the drive has some very special properties; The phase change drive allows for direct overwrite of data which magneto optical can't do inexpensively and the media has the very special property of NOT being susceptible to magnetic fields or as sensitive to static discharge which gives the media a very long shelf life.

7.2 Panasonic LF1000

POINTS OF INTEREST

- Read/Write optical disk.
- Can read CD−ROMs at 4X speed.
- Can read Kodak PhotoCDs.
- Media has a 15 Year shelf life.
- SCSI−2 Interface.
- Track/sector format as opposed to CD−ROMs spiraling record format.
- 165ms access time − much better than a tape file restore.
- 650Mb data storage per diskette.

- Diskettes are about $50 each.

THINGS YOU SHOULD KNOW

- Optical disk format not compatible with any other disk drive.
- Vendors don't seem to support UNIX very well − marketing is targeted for DOS/Windows and Macintosh.
- Do NOT purchase the PD drive which uses the parallel port interface − To my knowledge there is no Linux driver for it.

Installation

The LF1000 is SCSI−2 compatible device. It features a block size of 512 bytes and is compatible with the Linux SCSI drivers. This drive was installed on a PC compatible AMD 100MHZ 486 with an Adaptec 1542C SCSI bus−master controller. To install and mount a disk the following steps were taken;

Installation steps

- Install the drive and set the SCSI address to not interfere with other SCSI devices. Re−connect all cabling.
- Boot the computer. Your SCSI controller should note the new drive.
- During the Linux kernel boot, you should see an additional SCSI device. In my case, having a magnetic system disk for device /dev/sda it shows up as /dev/sdb.
- I did NOT partition the device because fdisk issued an overwrite warning and I did not want to change anything from a dosemu standpoint.
- mkfs −t ext2 /dev/sdb
- mkdir /pd
- mount −t ext2 −o ro,suid,dev,exec,auto,nouser,async /dev/sdb /pd − Read only
- mount −t ext2 −o defaults /dev/sdb /pd − Mount drive W/R

Your ready to "Rock'n'Roll"

Usage hints

- The media which comes with the drive is reported be re−writable about 500,000 times. This means that it is not advisable to install a live operating system such as Linux on the phase change optical drive. These live operating systems tend to cache processes to and from disk. Over time this can easily approach the phase change media life.
- Mount drive read only as much as possible.
- When writing to the drive do so in large chunks. This will help reduce any file fragmentation which will require more read seeks.
- This is however an excellent media for backups, gifs, mpeg or storing large programs which you don't use that often. The restore from backup is much faster that tape. Backups can be performed using the cp −rp command without the need for the ftape driver. This however, will replace symbolic links with the actual file.
- If while using the PD for writing, You find that the file you just wrote to the disk are not there, chances are that the disk write protect tab is in write protect mode and you mounted it in read/write mode.

7.3 Additional Configuration concerns by Jeff Rooze

Hello,

I read your article on configuring the Panasonic LF−1000 for Linux. I have configured my system so that the optical drive has its own device name and the CD−ROM has its own device name. This has allowed me to mount either media at any time. I do not require any media in the drive when I boot Linux. Also I am using the optical drive as an ext2 formatted media.

I had a couple of minor difficulties in doing so.

First, I had configured my hard drive at SCSI ID 6 and my PD at SCSI ID 4. (I wanted to have the hard drive at a higher priority that the PD). This caused a problem with the Linux SCSI driver. The driver scans the SCSI devices from the Lower SCSI id's to the higher (eg: 0 .. 6). Consequently my logical device names were assigned differently depending on which type of media was installed in the PD drive. This caused a big problem. My Linux partition is on my SCSI hard drive and the root device name would change! I corrected this problem by modifying the software in the kernel SCSI driver to scan the devices in reverse order.

Second, the distribution Linux kernel does not scan all SCSI LUNS. The PD/CD drive has a mode that establishes the CD−ROM at LUN 1 and the PD at LUN 0. This mode is selected by the configuration switches on the PD/CD drive. Switch #2 should be down (off?). If this switch is up (on?), the signature of the device is dependent upon the media that is installed and it only reports this device on LUN 0. If no media is installed I think it defaults to CD−ROM. I am using an Future Domain 16−xx SCSI interface card and the software in Linux kernel driver supports an optical device signature when scanning the LUNS. I assume that this is standard for most of the SCSI drivers. I reconfigured the kernel to enable the "scan all LUNS" switch. The kernel then assigns different device names for each device. The following is an excerpt from by boot log. You will note a series of errors in this log. This is because I did not have the optical media installed in the drive and the driver was attempting to look at the partition table to determine the block size. Fortunately it defaults to 512. I am planning on modifying the Future Domain SCSI driver to not do this when it detects the optical device.

```
>  scsi0 <fdomain>: BIOS version 3.2 at 0xde000 using scsi id 7
>  scsi0 <fdomain>: TMC−18C50 chip at 0x140 irq 12
>  scsi0 : Future Domain TMC−16x0 SCSI driver, version 5.28
>  scsi : 1 host.
>    Vendor: CONNER    Model: CP30545 545MB3.5  Rev: A9AF
>    Type:   Direct−Access                      ANSI SCSI revision: 02
>  Detected scsi disk sda at scsi0, id 6, lun 0
>    Vendor: MATSHITA  Model: PD−1 LF−1000      Rev: A109
>    Type:   Optical Device                     ANSI SCSI revision: 02
>  Detected scsi disk sdb at scsi0, id 4, lun 0
>    Vendor: MATSHITA  Model: PD−1 LF−1000      Rev: A109
>    Type:   CD−ROM                             ANSI SCSI revision: 02
>  Detected scsi CD−ROM sr0 at scsi0, id 4, lun 1
>  fdomain: Selection failed
>  scsi : detected 1 SCSI cdrom 2 SCSI disks total.
>  SCSI Hardware sector size is 512 bytes on device sda
>  fdomain: REQUEST SENSE Key = 2, Code = 3a, Qualifier = 0
>  last message repeated 3 times
>  sdb : READ CAPACITY failed.
>  sdb : status = 0, message = 00, host = 0, driver = 28
>  sdb : extended sense code = 2
>  sdb : block size assumed to be 512 bytes, disk size 1GB.
>  .
>  .
```

```
>  .
>  Partition check:
>    sda: sda1 sda2 sda3
>  scsidisk I/O error: dev 0810, sector 0
>    unable to read partition table of device 0810
```

Third, I modified my file system table (/etc/fstab) to list each device but do not attempt to auto mount when booting. I have included an excerpt from my fstab. The most important options are the noauto, rw(ro), and the checkpass flag.

To create a ext2 file system on the PD, I used the command "mkfs.ext2 −i 2048 /dev/sdb".

```
# fstab - List of file systems
#
# device  mount   type          options              dumpfrequency
checkpass
/dev/sdb /optd    ext2   rw,user,suid,noauto,sync,exec,dev,umask=0 0 2
/dev/sr0 /dist  iso9660  ro,user,suid,noauto,sync,exec,dev 0 2
```

After making these changes, I have had no problems with mounting either media. All I need to do is to load the media and type "mount /optd" or "mount /dist" and the system does all the rest.

I hope this information is useful.

```
Jeff
--
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
\ Jeff Rooze -- http://www.treknet.net/~jrooze -- jrooze@treknet.net /
/  If builders built buildings the way some programmers write        \
\  programs, then the first woodpecker that came along would destroy /
/  civilization.                                     GERALD WEINBERG \
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

I tried Jeff's suggestion. Here are the steps I performed;

- Modify my kernel using "make xconfig" in the /usr/src/linux directory and installed it.
- Change the mode jumper on the PD drive to non−DOS mode. I soldered a switch across the mode jumper connections and routed it the the back panel. I figured out which switch position was the open position and labeled this one for DOS. The other position is of course Linux. So before I boot my system I decide which OS I'll be using and set the switch accordingly. History shows it staying in the Linux position more and more.
- Re−boot your system. You should now see multiple LUN show up during boot for the PD SCSI device number − It works great!!! If you have an older kernel modify the "/usr/src/linux/drivers/scsi/config.in" file.
- Update the fstab for both CD and PD drives.
- Use appropriate mount command.
- "df" to make sure your ready.

I did try moving my primary SCSI drive to 6 but experienced some difficulties. Can't remember exactly what it was but it may have been that my controller "Adaptec 1542" with "Corel SCSI" requires a bootable disk and SCSI 0 for the BIOS install to work properly with DOS. So I switched it back and enjoyed playing with my properly install PD drive! With this configuration "workman" − the audio CD player util − works fine.

---

8. Optical Disk HOWTO Development Page

Skip's Homepage may contain a more revised copy than this HOWTO!