For Details, Contact:
Thay 'Soma'sundaram
Telephone: 850-644-6448
E-mail: soma@sb.fsu.edu

Thayumanasamy Somasundaram
Kasha Laboratory of Biophysics
Institute of Molecular Biophysics
Florida State University
Tallahassee, FL 32306-4380

Soma's Installation Notes

Linux Installation of a USB Mini Drive

Installation notes for Iomega's USB Mini Drive under RedHat Linux (ver. 2.4.4)

Tallahassee, March 2003: The following notes describe how to install an Iomega's USB Mini Drive (*aka* USB Flash Storage, USB Smart Drive, etc) under RedHat 7.0 Linux (ver. 2.4.4). Several vendors including Iomega, Belkin, Fuji, and Sony sell these drives. They are compatible even with Windows (Win 98, WinXP, Win2000) and Macintosh operating systems (O/Ss). These drives are compatible with new the USB 2.0 standard, however, their transfer speeds are akin to that of old USB 1.1 standard (~ 1Mb/s read rate).

Unit Description

The USB Mini Drive is a portable and easy-to-use storage device that looks like a key chain. It has two parts: 1) the storage side with LED and an USB interface and 2) a cover for the USB interface.

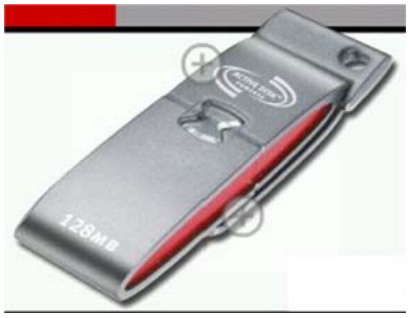

Figure 1: Iomega's 128 MB USB MiniDrive.

During transport and storage the cover protects the interface and while in use the user removes the cover and plugs the USB interface to an USB port. The USB Mini Drive does not require additional power supply or software and is fully compatible with Linux Kernel version 2.4 and above.

Installation

The USB Mini Drive installation is very simple and quick. As a `root` first create a mount point for the drive and decide whether to allow the user to have mounting authority and whether you want to allow auto-mount. Then open the USB MiniDrive exposing its USB interface and connect it to any open USB port in your Linux machine (see *Figure2* with closed and open USB interface). In principle, the USB MiniDrive should be recognized as a new device. If not run through the `/var/log/dmesg` file to find out what the device name is and specifically mount it.

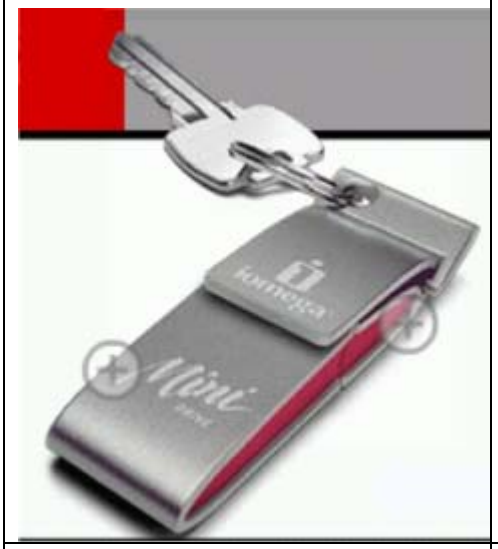

Fig. 2a Iomega USB MiniDrive (closed)

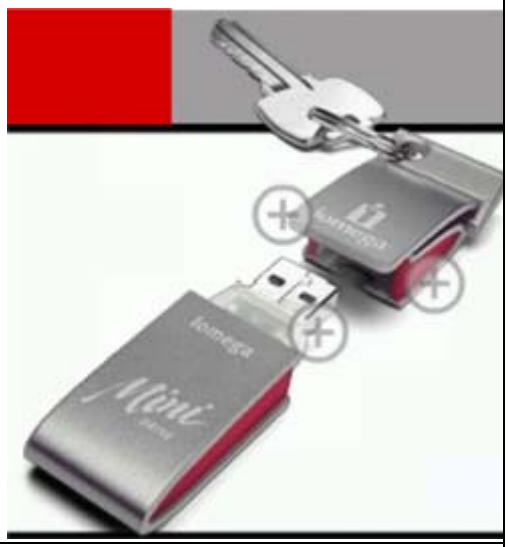

Fig. 2b Iomega USB MiniDrive (open)

Specific Installation (First time)

In this section, it is assumed that you are installing the drive for the first time. First log in as `root` and make a copy of the `/var/log/dmesg` into a text file (say `preusb-dmesg.txt`). Then open the USB MiniDrive and plug it to an open USB port. It is likely that there will be more information appended to the `dmesg` about the drive and save it once again as a text message (say, `postusb-dmesg.txt`), if not look at the tail section of `/var/log/messages`. Close examination of the `postusb-dmesg.txt` may reveal the device name assignment for the USB drive (it is usually a SCSI disk and therefore likely to be `/dev/sdb1` or `/dev/sdc1`; depending upon whether the

machine already has one or two SCSI disks, respectively; system ide-interface harddisks, i.e., bootable and Linux o/s are usually assigned `/dev/hda1` through `/dev/hda9` etc depending upon the partition numbers 1 through 9). In the event that you are unable to find out the device designation from `dmesg` file, try using the `/sbin/fdisk -l` command. This command lists the partition tables for the Linux system and will show the designation of the device. Another way to find out whether the device is available is to look at `/proc/scsi/scsi` (text file) or issue the command `cdrecord -scanbus`. A typical session is shown below:

```
root@raccoon[2:27pm]~>more post-usb-msg.txt | /var/log/dmesg
hub.c: USB new device connect on bus1/1, assigned device number 2
scsi1 : SCSI emulation for USB Mass Storage devices
  Vendor: I0MEGA    Model: Mini 128*IOM      Rev: 3.04
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi removable disk sdb at scsi1, channel 0, id 0, lun 0
SCSI device sdb: 256320 512-byte hdwr sectors (131 MB)
sdb: Write Protect is off
 sdb: sdb1   | Here the device designation is established as /dev/sdb1
```

In the event dmesg does not reveal the device designation try using the command `fdisk -l`

```
root@neptune[2:42pm]~>/sbin/fdisk -l
Disk /dev/sda: 255 heads, 63 sectors, 9732 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1             1      9729  78148161   83  Linux
| This SCSI disk (/dev/sda1; first SCSI disk, partition 1) is actually a 80 GB Maxtor FireWire drive

Disk /dev/sdb: 255 heads, 63 sectors, 9732 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1             1      4178  33559753+  83  Linux
/dev/sdb2          4179      9732  44612505   83  Linux
| These SCSI disks (/dev/sdb1 + /dev/sdb2) are two partitions of a single 80 GB Maxtor FireWire drive

Disk /dev/sdc: 6 heads, 63 sectors, 678 cylinders
Units = cylinders of 378 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdc1   *         1       678    128110+   6  FAT16
| This is our SCSI disk (/dev/sdc1; third SCSI disk, partition 1) with FAT16 (Windows) filesystem
```

```
Disk /dev/hda: 255 heads, 63 sectors, 2431 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1       129   1036161   83  Linux
/dev/hda2           130       162    265072+  82  Linux swap
/dev/hda3           163       546   3084480   83  Linux
/dev/hda4           547      2431  15141262+   5  Extended
/dev/hda5           547       930   3084448+  83  Linux
/dev/hda6           931      2431  12056751   83  Linux
```

| These are system Linux ide-interface disks with several partitions, including the bootable partition.

Another way of looking at the SCSI disk is examining the text file `/proc/scsi/scsi`:

```
root@neptune[2:59pm]/>more /proc/scsi/scsi
Attached devices:
Host: scsi2 Channel: 00 Id: 00 Lun: 00
  Vendor: Maxtor   Model: 1394 storage     Rev: v1.2
  Type:   Direct-Access                    ANSI SCSI revision: 06
Host: scsi2 Channel: 00 Id: 01 Lun: 00
  Vendor: Maxtor   Model: 1394 storage     Rev: v1.2
  Type:   Direct-Access                    ANSI SCSI revision: 06
Host: scsi3 Channel: 00 Id: 00 Lun: 00
  Vendor: I0MEGA   Model: Mini 128*IOM     Rev: 3.04
  Type:   Direct-Access                    ANSI SCSI revision: 02
```

Subsequent Installation

Once the exact device designation has been known, the root may decide to allow auto-mount and umount of this specific drive by users, as long as no existing devices are removed or new devices are added. In the latter event the device designation may change, requiring the root to update the `/etc/fstab` file. The process of setting up the auto-mount by the user is for `root` to edit the `/etc/fstab` file to include the device name, mount point, file system, and options. A typical entry in the fstab file may look like the following:

```
# TS March 2003 in Dittmer
#Device Name     Mount point   Filesystem    Options             Retries
/dev/sdb1       /mnt/mini-usb  vfat         auto,owner,rw,user      0 0
```

| The root is allowing auto- and user- mount of the device with /dev/sdb1 designation to /mnt/mini-usb mount point

As soon as one plugs in the USB MiniDrive, the user can log in and issue the following mount command to mount the device:

```
soma@raccoon[1:24pm]~>mount /mnt/mini-usb/
soma@raccoon[1:24pm]~>df -k
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda1              1011928     80684    879840   9% /
/dev/hda9               256667    174671     68744  72% /home
/dev/hda6              3028080   1561144   1313116  55% /usr
/dev/hda5              3028080    862256   2012004  30% /usr/local
/dev/hda7              2016016     35240   1878364   2% /var
/dev/hda10             8815372   3021264   5346300  37% /d5
/dev/sda1             17639220  10763268   5979932  65% /d3
/dev/sda2             17639248   1487972  15255252   9% /d4
spruce:/d1            30842136  11977744  17297656  41% /spruce/d1
spruce:/d2            35278540  19478516  14007976  59% /spruce/d2
/dev/sdb1               127844      3948    123896   4% /mnt/mini-usb
```

If, however, the entry on `/etc/fstab` is slightly modified (from `user` to `users`) any user can `mount` and `umount` and even `umount` a device which has been mounted by another user (including the `root`; so be careful):

```
# TS March 2003 in Dittmer
#Device Name     Mount point   Filesystem    Options              Retries
/dev/sdb1      /mnt/mini-usb  vfat          noauto,owner,rw,users      0 0
| The root is allowing user- mount of the device with umount of the same device by any other user as well
```

One subtle point for users who have some experience as `root`. If you issue the command `mount` with more specific options (while you are an ordinary user) the `mount` will fail with an error message shown below, even though every user is allowed to mount the device (you are not allowing the o/s to read the `/etc/fstab` file):

```
soma@raccoon[1:38pm]~>mount -t vfat /dev/sdb1 /mnt/mini-usb/
mount: only root can do that
soma@raccoon[1:38pm]~>mount  /dev/sdb1 /mnt/mini-usb/
mount: only root can do that
soma@raccoon[1:38pm]~>mount -t vfat /mnt/mini-usb/
mount: only root can do that
```

so, issue `mount` with just `mount point` as shown below (this time the shell will read the `/etc/fstab` file and correctly mount the device with specified filetype):

```
soma@raccoon[1:38pm]~>mount  /mnt/mini-usb/
soma@raccoon[1:41pm]~>mount    | Mount with no arguments
/dev/sdb1 on /mnt/mini-usb type vfat (rw,noexec,nosuid,nodev,user=soma)
soma@raccoon[1:40pm]~>df -k
spruce:/d2            35278540  19478516  14007976  59% /spruce/d2
/dev/sdb1               127844      3948    123896   4% /mnt/mini-usb
```

once the data has been accessed and processing is complete, `umount` the device and remove the USB MiniDrive for storage and transport.

```
soma@raccoon[1:38pm]~>umount  /mnt/mini-usb/
```

Conclusion

This completes the Installation of Iomega USB Mini Drive. For further help consult with a systems manager, Iomega site, or read a Linux manual.

Note: All the figures are courtesy of Iomega Corporation .