Fedora Core Installation to an External USB Disk Drive

1. Introduction

The speed and size of external USB disk drives now makes it practical to run an operating system directly from such a drive. A typical requirement might be the testing of a new operating system release, or indeed a different operating system, without interfering with any existing computer configuration. The ability to enable a rescue attempt upon a faulty disk is an added attraction.

The approach should also appeal to nervous Windows users who wish to investigate the pleasures of a full Linux installation on a separate external USB drive with no impact whatsoever upon their existing system.

This document describes the installation of Fedora Core 3 on a Pentium Shuttle which uses a PATA disk (hda) as its *standard* disk in *normal* operating mode. Comments regarding FC4T1 and FC4T2 are also included. It may well be that when the stable version of FC4 is released much of this document will be redundant and will only be useful for trouble shooting purposes. The inclusion of two grub installation modes, one manual and one automatic has complicated the flow of this document. It is recommended that the reader should be very clear as to which sections are relevant to their particular requirements.

A great deal of this document is derived from the following excellent sources:

- http://simonf.com/usb/
  Booting Linux off of a USB drive.
  Author: Simon Ilyushchenko (simonf@simonf.com)
- http://www.linuxforums.org/forum/topic−29548−0.html
  Author: mjman
- Advice and proof reading by David Niemi is also gratefully acknowledged.

2. Objectives

2.1 Ideal Solution

- To install to a USB disk from the standard FC3 (FC4T1/2) distribution CD's or DVD.
- To be able to boot from the USB disk.
- To make **no** modification to any existing disks, including their Master Boot Records (MBR).
- To be able to set the BIOS boot order to CD first, USB−HDD second, PATA or SATA disk third, .. such that the presence or absence of the USB disk determines the operating system that is booted.

2.2 An Assessment of the Possible

2.2.1 USB Booting Considerations

To leave the installed hard drives untouched it is essential that the motherboard BIOS has the ability to boot from a USB 2.0 device. Three distinct situations can arise with regard to BIOSs and their recognition of USB disks.

1. The BIOS recognises the USB drive and is able to select it as a boot device.
2. The BIOS recognises the USB drive but is unable to select it as a boot device.
3. The BIOS does not see the USB drive at all.

The first two points should be easy to establish by inspection of the BIOS menu, whether the machine will really satisfactorily boot from a USB device will not become apparent until a correctly configured disk is available!

2.2.2 Testing USB Disk Recognition at Boot Time Using a Grub Enabled CDROM

A more positive test of the recognition of a USB disk during booting is now described as it is highly desirable to discover the true situation with regard to the machine in question as soon as possible. A simple grub

enabled CDROM ISO image is available from **www.vigla.eclipse.co.uk**. The image has been created using the script shown in Section 6. and it can be burnt to CD and used to test the machine. Simply set the BIOS boot sequence as described in Section 3.1.1 and enter a "c" when grub boots. Use the interactive features of grub as explained in Sections 4.3.3 and 5.1.2. to determine if the USB disk is available. Typical commands would be:

- grub>**cat (hdTAB** to see the available disks
- grub>**cat (hd1,TAB** to see the available partitions on a particular disk.

Experience has shown that some BIOS's will only recognise a USB device during booting if is plugged in and powered when the machine itself is switched on. If the USB disk is not found then in the above test **switch off** the computer and **unplug the mains lead**, re−plugin the mains lead and power up. It is possible that only some of the USB sockets will recognise the USB disk, try them all systematically.

If the USB disk cannot be identified at this stage then the BIOS is probably not going to play ball!!!!

2.2.3 USB Disk Found but No USB Disk Boot Option in the BIOS

Two alternative installations are possible if the BIOS does not support USB 2.0 device booting.

1. Booting from a pre−generated CDROM where again no modifications are made to any existing hard disk drives. See Subsection 2.2.2 and Section 6.
2. Modifying the Master Boot Record (MBR) of the *standard* disk. See Section 7.

3. Operating System Installation Stage

3.1 BIOS Settings

3.1.1 Boot Order

Configure the BIOS boot order settings of the Computer to

- First: CDROM
- Second: USB Hard Disk Drive
- Third: Hard Disk Drive

3.1.2 Disk Identification

Browse through the BIOS menus and make note of any information regarding the make and size of **all** the disks installed on the machine. This will make it easier later in unambiguously identifying which disk is which during both operating system installation and booting.

3.1.3 Linux Disk Naming

The Linux disk naming convention is as follows:

IDE Disks − The disks with the wide flat cables. Usually two controllers on the motherboard with two devices permitted per controller. First controller is called Primary and the second Secondary. The device connected at the end of each cable should be the master, the device connected half way along each cable should be the slave. A link on the disk/CDROM must be set to either master or slave.

- Primary Master Usually the First PATA IDE disk /dev/hda
- First Partition on /dev/hda /dev/hda1
- Second Partition on /dev/hda /dev/hda2
- .....
- Primary Slave Maybe Second PATA IDE disk /dev/hdb
- Secondary Master Usually CDROM /dev/hdc
- Secondary Slave Maybe Second PATA Disk /dev/hdd

True SCSI Disks are unusual on a “home” computer but SATA and USB disks are recognised as SCSI devices. It is not always clear which SCSI disk will considered the first, the boot order often being influential. Often trial and error is required, the installation disk in “linux rescue” mode or the bootable grub CDROM being suitable investigation tools.

- First disk /dev/sda
- First Partition on /dev/sda /dev/sda1
- Second Partition on /dev/sda /dev/sda2
- ...
- Second Disk /dev/sdb
- First Partition on Second Disk /dev/sdb1

Logical Volume Management (LVM) and RAID arrays are not considered here.

3.1.4 Grub Disk Naming

It may seem perverse that the boot loader labels the disks in yet another manner. The reason is that GRUB intended to operating system and hardware neutral. In fact the naming convention is straight forward!

- First Disk (MBR) (hd0)
- First Disk – First Partition (hd0,0)
- First Disk – Second Partition (hd0,1)
- Second Disk (MBR) (hd1)
- Second Disk – First Partition (hd1,0)

3.2 Installation

Insert the FC3 DVD or CD1 into the optical drive and boot the computer.

3.2.1 Initial Dialogue

- At the prompt enter “linux expert” – not mentioned in the Help menus!

This option makes available USB devices when the disk partitioning stage of the installation is reached. FC4T1/2 removes the necessity to use expert mode as the standard boot option now includes USB disk drive support.

3.2.2 Media Check

- If you are feeling lucky SKIP the media check.
- Note that the media check for FC4T1/2 is not functional – do not use.

3.2.3 Language and Keyboard Selection

- Select the appropriate keyboard layout and language.

3.2.4 Upgrade or Install

- Be most careful to select **Install Fedora Core** rather than Upgrade.

3.2.5 Initial Partitioning Dialogue

- Be most careful to select **Manually Partition with Disk Druid**
- **Do NOT** let the machine carry out the disk selection and partitioning automatically as implied by the other option.

3.2.6 Disk Selection

Disk identification and naming by the installer appears to have changed from FC3 to FC4T1/2, the latter ordering them in the same manner as the BIOS.

- Carefully examine the disk information provided by DiskDruid to be absolutely certain which disk listed corresponds with the USB disk.
- If the make of the disks is known this is shown at the top of the window and will help identification.
- In the case described here the *standard* disk appears as **/dev/hda** and the USB disk as **/dev/sda**. If the *standard* disk were to be a SCSI disk, or possibly a single SATA disk connected to a raid controller it may well appear as **/dev/sda** and the USB disk as **/dev/sdb**.
- Situations where the *standard* disk(s) is/are configured as logical volume(s) or as RAID array(s) can cause problems and are not considered here. A current hot topic!
- Examine the information looking for Windows partitions NTFS or FAT32 if the *standard* disk contains a Windows operating system.
- **Write down** the name of the target USB disk – **/dev/sda** say

3.2.7 Partitioning

The detailed sizes shown here are for a 20GB disk, different size disks will obviously require different partition sizes. The /boot partition can stay a constant at 100MB, the swap partition should be approximately twice the size of the machine's main memory, the remaining space can be assigned as one big lump for the operating system and user file space. Many alternative arrangements are possible but only this straight forward layout is considered here.

- It may be that the target USB disk has been used previously and contains unwanted partitions. Select the appropriate partitions and **Delete** them.
- Select NEW and enter the following information into the window for the first partition.
  1. Mount Point **/boot**
  2. File System **ext3**
  3. Allowable drives **sda** Leave the target USB disk ticked and **untick** the remaining drives.
  4. Size in MBytes **100**
  5. Additional Size Option **Select Fixed Size**
  6. Force to be Primary **Tick**
- Select NEW and enter the following information into the window for the second partition.
  1. Mount Point /
  2. File System **ext3**
  3. Allowable drives **sda** Leave the target USB disk ticked and **untick** the remaining drives.
  4. Size in MBytes **18000**
  5. Additional Size Option **Select Fixed Size**
  6. Force to be Primary **Tick**
- Select NEW and enter the following information into the window for the third partition.
  1. Mount Point **Not Applicable**
  2. File System **swap**
  3. Allowable drives **sda** Leave the target USB disk ticked and **untick** the remaining drives.
  4. Size in MBytes **1000**
  5. Additional Size Option **Select Fixed Size**
  6. Force to be Primary **Tick**

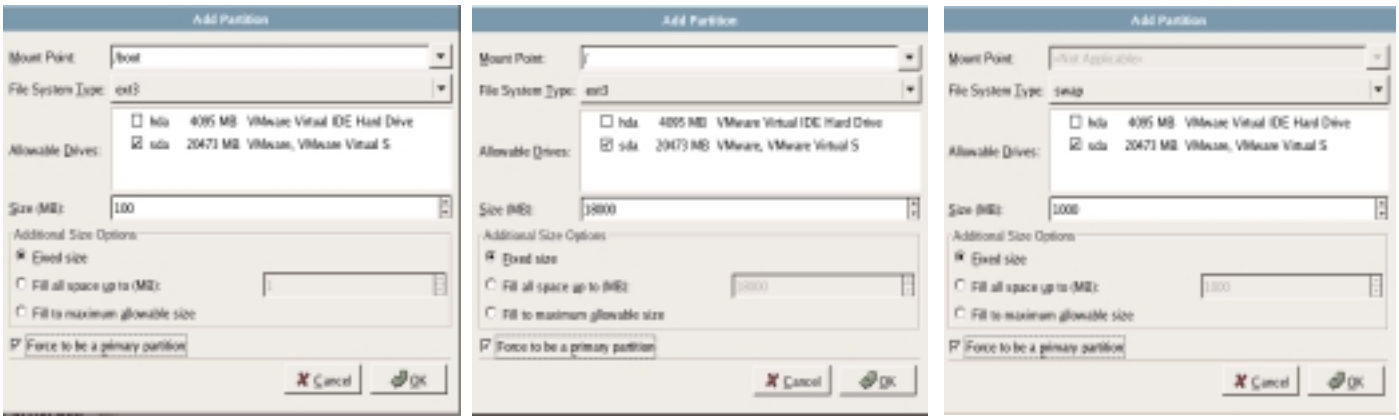

Fig 1 Partition Dialogue Boxes for **/boot**, / and **swap**

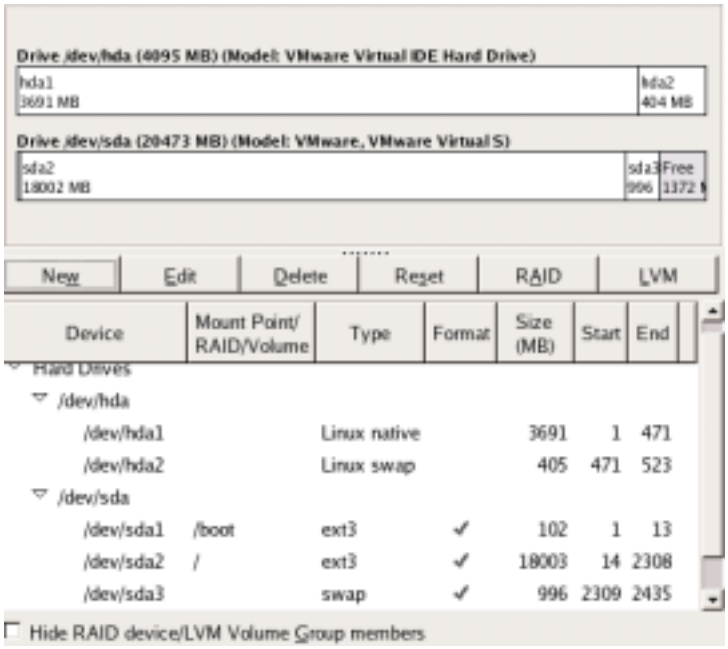

Fig 1 Disk Druid GUI Showing Disk Information **After** Partitioning
The partition information shown corresponds to these notes but the disk names do not as it was obtained during a VMware Installation.

3.2.8 An Aside – Skip as Required

- Disk Size BIOS problems are rare when installing to large IDE disks with a modern mother board and hence only two primary partitions are really necessary, the operating system (**root** file system) on /dev/hda1, and **swap** on /dev/hda2. However on a P4 Shuttle machine using a USB disk (only 20GB), a full install of FC3 (7GB or more) gave BIOS errors reported via GRUB when failing to boot. Introducing a separate 100MB partition for **/boot**, as described here, and reinstalling avoided this problem. However, the same two partition USB disk i386 installation that failed to boot on a 32bit Shuttle booted successfully on an AMD 64 bit Shuttle. It can be concluded that some mother boards/BIOSs have disk size booting problems with USB disks that are not evident on IDE or SCSI disks.

3.2.9 Summary of Final Disk Partition Arrangement

- All partitions have been set to primary

P1	/dev/sda1	/boot	100MB
P2	/dev/sda2	/	18GB approx
P3	/dev/sda3	swap	1GB approx

Continue with the installation when completely satisfied that the partitioning is correct. Confirm if/when asked to FORMAT the appropriate partitions.

3.2.10 GRUB Configuration During Installation – Version 1

Version 1 avoids the configuration of a boot loader during installation. Grub is installed on the USB disk *after* the installation has taken place. Things are improving fast with regard to the Redhat grub installer **grub–install**. For FC4T1 use the version 1 approach. For FC4T2 then version 2 described in Section 3.2.11 is reliable.

Right at the top of the screen select

- **Change Boot Loader**
- Select **Do not install a boot loader**
- **OK**
- **Continue with no boot loader**

3.2.11 GRUB Configuration During Installation – Version 2

Right at the top of the screen a statement of the form "The GRUB bootloader will be installed on /dev/sdb" is shown. The disk drive indicated is probably not the USB disk.

Make any necessary changes to the disk labels in the centre of the screen before moving to the lower left hand corner.

- Tick "Advanced Configuration" and select NEXT

The information at the top of the screen indicates that two locations are available to which "grub things" may be written. One will refer to the MBR of a disk (eg /dev/hda, /dev/sda, /dev/sdb), the second will refer to the *partition* on which the installation of the operating system is taking place (eg /dev/hda1, /dev/sda1). In the case described here the required location is the MBR of the USB disk. This can be selected by high–lighting the required drive in the table and using the arrows to move the drive to the top of the list. The top of the screen should reflect the changes made.

When satisfied that all is well press NEXT.

3.2.12 Network Configuration

The use of fixed IP addresses is described here, ignore this and rely on DHCP if preferred. Only one network card is assumed. The numbers required for this section are likely to be pre–determined by the user, by a local network administrator or an ISP and hence totally unrelated to those listed below.

- Select **Edit** and then *deselect* **Configure using DHCP**, leaving **Activate on Boot** ticked, enter
- 192.168.10.1 something applicable
- 255.255.255.0 almost certainly correct
- Set the device name manually **fred** say, or something applicable
- Default route 192.168.10.254 would probably match the above
- DNS server 192.168.10.1 maybe suitable if local, or use the IP address provided by your ISP

3.2.13 Security Considerations

To a new user both the firewall and SELinux configuration may well cause grief. On the other hand being hacked from the Internet causes even greater grief. Living behind a reasonably good firewall with no hooligans in the vicinity can lead to laziness. On no account expose a machine with a Public IP address to the internet without a firewall. Most ADSL modem, firewall routers do a reasonable job of buffering the world if configured correctly. The **unsafe** option is shown below – make your own choice !!!!

- Select **No Firewall** at the top of the screen
- Enable SELinux Set to **Disabled**
- **PROCEED** The installer is not happy !!!!

3.2.14 Additional Languages

- Select as required

3.2.15 Time Zone

- Select the nearest city
- Set System clock uses UTC

3.2.16 Password

- Enter a suitable password – twice

3.2.17 Package Installation

- Select required packages – if in doubt scroll to Miscellaneous at the bottom and select Everything – a full install takes 30mins to 90mins depending upon the speed of the machine.

3.2.18 The Installation

- Go for it with Next and Next again – go for coffee

4. Post Install Configuration

For all Fedora Core releases to date it is necessary to create a magic initial ram disk file containing USB orientated modules. In addition, if no booting mechanism was created during installation, Section 3.2.10, it is also necessary to copy grub files to their correct location and write grub type things to the MBR of the USB disk. Some of this may appear tricky but provided care is taken all will be well.

The devices named below correspond to those used above. If your disk configuration is different be very careful to translate the following to your environment. It may well be worth printing these pages and going through it changing all the device names such that they correspond to your machine configuration.

4.1 The First Boot – From Distribution DVD Into Rescue Mode

4.1.1 Booting into Rescue Mode

When all packages have been copied the installer requests that the DVD/CD is removed from the drive and that the machine is re–booted. Don't always do as you are told !

- If using a DVD leave it in place and reboot or
- if using CD's remove the last CD and insert the **first** CD and reboot.
- The installation DVD or CD will boot again
- At the prompt type **linux rescue**
- Select the language and keyboard as required
- Setup Network Interfaces – **No**
- ... search for Linux installations ... **Skip**

A Linux shell prompt is now available awaiting commands.

4.1.2 Mounting the USB Files Systems in Rescue Mode

The following three commands mount the main disk partition (/dev/sda2) and the boot partition (/dev/sda1) such that they can be modified. The **chroot** command ensures that all commands and file name are taken relative to **/mnt/source**. Type

- **mount /dev/sda2 /mnt/source**
- **mount /dev/sda1 /mnt/source/boot**
- **chroot /mnt/source**

4.2 Initial RAM Disk (initrd)

4.2.1 Introduction

During booting, modules (device drivers) for the particular machine may need to be installed. The initial ram disk (**initrd**) is in the form of a file (named **initrd_usb.gz** below) which is uncompressed and loaded into RAM.

Initrd has sufficient content such that it can act as a temporary "root" file system. It will typical have the following directories – /bin, /dev, /lib, /etc, /proc, /sysroot, /rootfs, /sbin ... together with the file **init**. The file is an executable and is in the form of a shell script. It is this file that does the work of installing the required modules, creating device driver entries in /dev, selecting the final root device etc.

4.2.2 Creating an Initial Ram Disk (initrd)

The **mkinitrd** command that generates **initrd_usb.gz** is long and the dumb terminal window is small, triple check the syntax and do not hit return until completely happy with your typing! The **mkinitrd** command may be split across two lines by placing a **space** followed by a **backslash** at the end of the first line before hitting **return** as shown below. Note the difference between underscore _ and dash –. The parameters presented to the **––preload** option are the names of the modules eg **usb–storage**, **/boot/initrd_usb.gz** is the name of the generated file and **2.6.9–1.667** is the version of the kernel to be used such that matching drivers can be selected. The kernel version must match that present in **/boot** so it is probably wise to type **ls –l /boot** and check the **vmlinuz–2.6....** file name.

```
ls -l /boot
total 5280
-rw-r--r--  1 root root    5824 Jun 16  2004 boot.b
-rw-r--r--  1 root root     612 Jun 16  2004 chain.b
-rw-r--r--  1 root root   50929 Nov  2 19:53 config-2.6.9-1.667
drwxr-xr-x  2 root root    4096 Mar 13 10:58 grub
-rw-r--r--  1 root root  394895 Nov  9 15:38 initrd-2.6.9-1.667.img
-rw-r--r--  1 root root   81860 Sep 29  2004 memtest86+-1.26
-rw-r--r--  1 root root     640 Jun 16  2004 os2_d.b
-rw-r--r--  1 root root  714266 Nov  2 19:53 System.map-2.6.9-1.667
-rw-r--r--  1 root root 1405861 Nov  2 19:53 vmlinuz-2.6.9-1.667
```

```
mkinitrd --preload=ehci-hcd --preload=usb-storage --preload=scsi_mod \
--preload=sd_mod  /boot/initrd_usb.gz  2.6.9-1.667
```

```
ls –l /boot
```

The new file **initrd_usb.gz** should now be present.

*4.2.3 If GRUB **has been** Configured During Fedora Core Installation*

If manual GRUB installation and configuration, Section 4.3, is not required then

- **exit** To leave **chroot** environment
- **umount /mnt/source/boot**
- **umount /mnt/source**
- remove the DVD/CD
- **exit**

The machine should reboot from the USB disk and offer a **grub** prompt.

NB It is possible that later, during booting, timing problems arise when modules are installed by the commands present in linuxrc. This will cause considerable difficulty to a novice, hopefully they will not occur. A description of the remedies available are given by http://simonf.com/usb/. It may be necessary to *go inside* the compressed initrd image(**initrd_usb.gz**) and introduce **sleep** commands at appropriate locations in the **linuxrc** script. See Section

*4.3 Configuring GRUB – the **GR**and **U**nified **B**ootloader*

4.3.1 Introduction

The actions described in this section should not be required if GRUB has been installed correctly during the installation phase, see Section 3.2.11. It does however provide detailed advice on how to install GRUB if things have gone wrong or if installation of another operating system has corrupted the grub configuration.

The script **grub–install** is provided to carry out this task but may well fail to work correctly with FC4T1 or with a complicated disk configuration. The actual commands to install **grub** by hand are relatively straight forward and it is recommended at the time of writing that they should be used instead of **grub–install**.

During the following activities the machine will need to be booted into **linux rescue mode** with the disks mounted in a **chroo**t environment as described in **Sections 4.1.1** and **4.1.2**. The machine may well currently be in this state if **Section 4.2.3** has been skipped.

Grub can be used in two distinct modes:

1. When the machine has been booted into *rescue mode* from the distribution. In this situation it is used to write "grub type things" to the MBR (and probably elsewhere) of the USB disk.
2. During the booting of the machine from the USB disk. In this case grub usually reads commands from a configuration file **/boot/grub/grub.conf**. However, it is still possible to issue grub commands interactively during the booting procedure.

4.3.2 Copying GRUB Files

It is necessary to copy a set of grub files to the **/boot/grub** directory. These files are used when grub writes to the USB disk MBR.

- **cd /usr/share/grub/i386−redhat**
- **cp * /boot/grub**

It is also a good idea to create a unique file within this directory such that it can be listed later.

- **echo This is a USB disk > /boot/grub/tester**

4.3.3 Writing GRUB to the MBR of the USB Disk

Some experimentation can now be undertaken to ensure there is no confusion as to the way **grub** names the disks. If a TAB is typed into a grub prompt then the software will try and auto−complete the line, it will also show the alternatives if there is a choice. This can be used to confirm the grub naming convention.

Exiting from the chroot environment but remembering that the USB disk partitions are still mounted the **grub** command can be executed.

- **exit** To leave **chroot** environment
- **/mnt/source/sbin/grub** To run the **grub** command located on the **USB** disk

Experiment with the **"grub" cat** command and find the file **tester** created earlier. Use that **TAB** for auto−completion !!!!

- grub>**cat (hd1,0)/grub/tester** To confirm the USB disk identity and location of copied files

Interactively explore the disks on the system using grub. grub> **help** will list the available commands, stick to **cat** unless you know what you are doing! When satisfied with the identity of the correct disk set grub to work.

- grub>**root (hd1,0)** Tell grub where the vital files are !!!!
- grub>**setup (hd1)** Write to the MBR of the USB disk !!!!
- grub>**quit**

At this stage no grub configuration file has been created. A prototype could be generated now and then modified dependant upon the response of the computer BIOS. However a reboot and experimentation with grub during booting should clarify the situation. So ..

- **umount /mnt/source/boot**
- **umount /mnt/source**
- remove the DVD/CD
- **exit**

The machine should reboot from the USB disk and offer a **grub** prompt.

5. Final Configuration

5.1 First Boot − From USB Disk

5.1.1 Introduction

The relevance of this section is determined by whether or not a **grub.conf** file was generated automatically during operating installation. However the information and techniques presented here are of great help

during trouble shooting mis−configured or corrupted installations. During booting the grub command line interface can be invoked by entering a “c” before the timeout period is over.

5.1.2 Determining the Disk Identities at Boot Time

The BIOS may well recognise the disk differently during the install procedure compared with that determined during the boot sequence. Recall that during the installation the machine was booted from the CDROM drive, it then appeared to recognise IDE drives (hd0) and then USB (hd1) drives. When booting from a USB disk the BIOS under consideration here recognised the USB disk first (hd0). To be certain of the configuration the **“grub” cat** command can again be utilised. Find that **tester** file again using

- grub>**cat (hdTAB ...** To discover the truth !!!!

When satisfied that the USB disk has been identified correctly an attempt to boot interactively can be made. A careful note should be made of the **successful grub commands** as these form the basis of the **grub.conf** file to be created later. The two commands required are of the form:

- grub>**kernel (hd0,0)/vmlinuz−2.6.9−1.667 ro root=/dev/sda2**
- grub>**initrd (hd0,0)/initrd_usb.gz**
- boot>**boot**

The file **/vmlinuz−2.6.9−1.667** is the kernel itself, referred to previously, the **/initrd_usb.gz** file is the one generated earlier. The last vital piece of information required for a successful boot identifies which disk contains the “operating system” files and hence should be mounted as the / or **root** device. In the situation considered throughout this document this is the **second partition** on the **USB disk** which in Linux terms is **/dev/sda2**.

5.1.3 Creating GRUB Configuration File

Having successfully booted by typing interactive grub commands the grub configuration file **/boot/grub/grub.conf** should be created essentially containing the same commands. A minimalist version is shown here.

```
timeout=10
title FC3 – i386 USB
              kernel  (hd0,0)/vmlinuz-2.6.9-1.667  ro  root=/dev/sda2
              initrd  (hd0,0)/initrd_usb.gz
```

6. Creating a Bootable CDROM

6.1 Bash Script

6.1.1 Introduction

The following script has been successfully tested using FC4T2. The section concerned with **grub.conf** high−lighted in bold should be modified as required. The CD writing commands are located at the end of the script and again will probably require modification.

```
#!/bin/bash
#create_grub_cd
#Run as root in root’s home directory
#Create an empty file floppy sized
dd if=/dev/zero of=fd_file bs=512 count=2880
#Write an MSDOS file system to the file
mkfs –t msdos fd_file
#Check for a mount point
if ([ ! –d /mnt/zip ]);then mkdir /mnt/zip;fi
#Mount the file as a loopback device (/dev/loop0 by default)
mount –o loop fd_file /mnt/zip
#Create the directory structure for grub
mkdir /mnt/zip/boot
mkdir /mnt/zip/boot/grub
#Copy the essential grub files
```

```
cp /usr/share/grub/i386-redhat/* /mnt/zip/boot/grub
#Create a prototype grub.conf file
cat > /mnt/zip/boot/grub/grub.conf<<END_OF_CONF
timeout=10
default=0
title FC4T2 i386 USB
              kernel (hd0,0)/vmlinuz-2.6.11-1.1226_FC4 ro root=/dev/sda2
              initrd (hd0,0)/initrd_usb.gz
#title Windows
#             rootnoverify (hdx,y)
#             chainloader +1
END_OF_CONF
#Check if /dev/fd0 exists, mv it if it does
if ([ -e /dev/fd0 ]);then mv /dev/fd0 /dev/fd0_orig;fi
#Link /dev/fd0 to /dev/loop1 to fool grub !!
ln -s /dev/loop0 /dev/fd0
#Run grub in batch mode
/sbin/grub --batch <<EOF
root (fd0)
setup (fd0)
quit
EOF
#restore fd0 if required
rm -f /dev/fd0
if([ -e /dev/fd0_orig ]);then mv /dev/fd0_orig /dev/fd0;fi
umount /mnt/zip
#########################################################
#Now make a bootable CD ISO image
# Make a dummy directory in /tmp
mkdir /tmp/dummy
#Copy the floppy image file to /tmp/dummy
cp fd_file /tmp/dummy
# Create the ISO image grub_image.iso
mkisofs -b fd_file -o /tmp/grub_image.iso /tmp/dummy
rm -rf /tmp/dummy
#########################################################
#cdrecord -scanbus                              to discover CD Writer device
#cdrecord dev=ATA:1,0,0 speed=8 /tmp/grub_image.iso
#########################################################
```

7. Modifying MBR of Existing Disk for GRUB

7.1 Introduction

The most usual disk layout for a “home” user is to have Windows installed on the first disk and that the MBR of this disk has been modified by Windows. If an automatic GRUB configuration is selected during FC3 installation and the “Windows” MBR selected then GRUB will have replaced the Windows information on the MBR and created a grub.conf file to permit dual booting of Windows and FC3.

7.2 Typical Dual Boot grub.conf

The grub.conf file below shows the situation where the first disk recognised by the BIOS contains Windows (hd0) and that the USB disk is detected as the second disk (hd1).

```
timeout=10
default=0
title FC4T2 i386 USB
              kernel (hd1,0)/vmlinuz-2.6.11-1.1226_FC4 ro root=/dev/sda2
              initrd (hd1,0)/initrd_usb.gz
title Windows
              rootnoverify (hd0,0)
              chainloader +1
```

8. Modifying an Existing Initial Ram Disk

8.1 Introduction

It is possible that problems arise during booting of USB devices, these are sometimes caused by delays whilst loading USB modules. Various references have indicated that introducing sleep commands can solve such problems. This section briefly describes how an **initrd_usb.gz** file may be unscrambled, modified and put back together again. Examination of the **initrd−2.6...img** generated by **mkinitrd** from FC4T2 shows a single sleep 10 command following the insmod of usb−storage.

8.2 Detailed Instructions

8.2.1 Unpacking

- **cd** go to root's home dir
- **mkdir clean; cd clean** make and go somewhere clean
- **cp /boot/initrd_usb.gz .** cp initrd_usb.gz to clean dir
- **gunzip −c initrd_usb.gz | cpio −i** gunzip and expand

This unpacks a complete directory structure − the vital file is **init**, edit as required.

8.2.2 Repacking

Remember to move the original file first or it will be incorporated in the new initrd image!!

- **mv initrd_usb.gz ∼**
- **find . | cpio −c −o | gzip −9 > ∼/initrd_usb_modified.gz**

The file can then be relocated to **/boot** and **grub.conf** modified to match.