

## Section 1 - Some File Types

### Everything in Linux is a File or a Process.

#### ISO Image Files

An .ISO image is an **archive file** with a .iso file extension and has a file format (ISO 9660) defined by the International Organization for Standardization (ISO). An .ISO file is also called a 'disk image' file when it contains the contents of a CD or DVD.

The main features of an ISO image file are that it is a compressed block by block copy of the original files on a hard drive and it can be easily burned to a DVD or CD.

#### Device Files (or Special Files)

a device file or special file is an interface for a device driver; it presents in a file system as if it were an ordinary file. Device files can be found in the directory '/dev'

##### Character Device Files (also called Serial Devices)

Character device files are 'device nodes' used to drive physical devices (such as mice& keyboards) one byte (character) at a time..

##### Block Device Files

Block device files are 'device nodes' through which the operating system moves data in the form of random addressable data blocks (e.g. 512 byte disk blocks) to physical devices such as hard disks, CD-ROM drives, or memory-regions.

##### Pseudo-Devices

Device nodes on are not always associated with physical devices. Nodes that lack an association with a physical device are called pseudo-devices. Psuedo-devices provide various functions handled by the operating system. Some of the most commonly-used (character-based) pseudo-devices include:

/dev/null	Accepts inputs from the OS, discards the input and produces no output.
/dev/zero	Produces a continuous stream of NUL (zero value) bytes.
/dev/random	Produces a variable-length stream of random numbers.

#### Loop devices

a loop device, vnd (vnode disk), or lofi (loopback file interface) is a pseudo-device that makes a file accessible as a block device.

**Sometimes, the loop device is erroneously referred to as 'loopback' device, do not use this terminology when referring to loop devices, it can cause confusion.**

#### Mounting an ISO as a loop device

Mounting a file containing a disk image(.ISO file) on a directory requires two steps:

1. association of the file with a loop device node (by the command `losetup`),
2. mounting of the loop device at a mount point directory.

These two operations can be performed either using two separate commands, or through special flags to the mount command.

example (use sudo):

```
losetup /dev/loop0 ~/example.iso  
mount /dev/loop0 ~/isomount
```

The mount utility is usually capable of handling the entire procedure:

```
mount -o loop example.iso ~/isomount
```

Unmount the device (use sudo):

```
umount ~/isomount  
umount /dev/loop0
```

## Section 2 - Grub Boot Process

### Grub2 Installation.

The `grub-install` command creates a directory `/boot/grub` on the nominated drive's boot partition and installs into `/boot/grub` a number of grub files. It then installs a Grub MBR on LBA 1 and 'embeds' a second stage boot loader, which is normally located in the 'unused blocks' LBA 2 to 63.

### Booting a Grub2 Linux System.

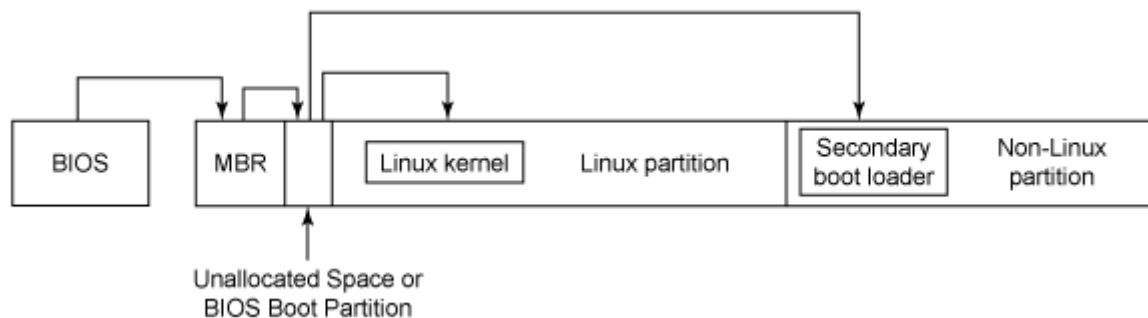
On Boot-up BIOS executes and initiates the Grub2 MBR loader which loads the second stage Grub boot loader, which uses the data in the `/boot/grub/grub.cnf` file data to complete the boot process. `grub.cnf` contains information on the available boot operating systems and their booting parameters.

Two of those parameters are

- 1.the location of the Kernel (`vmlinuz`)
- 2.the location of the RAM-drive (`initrd`)

The Kernel is a program that constitutes the central core of a computer operating system and when loaded into memory and started it remains in memory for the whole time that the computer is in operation. `vmlinuz` may be a compressed version of the Linux kernel or a link to a compressed or an uncompressed kernel.

The Initial RAM drive (`initrd`) is a temporary root file system created in RAM and mounted during system boot. `initrd` is initiated by the Grub2 second stage, it contains executables and drivers that do the hardware detection, device discovery and the module decompression/loading (`vmlinuz + .mod`) necessary to get the Linux root file system mounted. Once this is done `initrd` is unmounted and its memory freed.



It is possible to install GRUB's code in a Linux boot partition's first 63 blocks rather than in the MBR. Used in this way, another boot loader resides in the MBR and that boot loader then 'chain-loads' GRUB.

## Section 3 Making a Bootable EXT 2/3/4 USB

### Casper

Casper is a 'hook' for `initrd` used to produce an `initrd` capable of booting '*live iso*' files. Debian uses a 'fork' (i.e, a derivative) of casper called 'live'.

A 'hook' is a technique for altering or augmenting the behaviour of a software by intercepting, inspecting and redirecting certain messages that are passed between two software components.

## Persistence

A persistent USB Linux install is one that allows the user to save data changes to the USB device instead of leaving the data in system RAM and consequently losing it when the device is removed. Some changes are not saved persistently, e.g. graphic and network card settings. Using persistence can be convenient but it is less secure.

## Partitioning the USB

Using GParted, partition a USB Flash drive into two partitions. ext2 is non journaling and results in less USB-memory wear than the journaling systems ext3/4. Casper requires that you label the partitions, **casper-rw** and **home-rw**. The partition casper-rw should be at least 2GB. This will provide room for 2 or 3 system iso files. The partition home-rw can be whatever size you consider appropriate for your home directory.

For the purpose of this explanation we will assume that your USB memory stick is drive 'sdb', therefore 'sdb1' will be 'casper-rw' and 'sdb2' will be 'home-rw'. Exit and close GParted. Hint, use the command '**sudo blkid**' any-time you need to identify drives and partitions; '**df -h**' is also useful for this purpose.

'Safely remove' your USB casper-rw; unplug it for a few seconds, then plug it in again and open it with your file manager. The following is based on the assumption the file manager mounts the casper-rw at /media/casper-rw.

## Install grub2

Where '/media/casper-rw' is the mount point for the USB stick and '/dev/sdb' is the USB's system device name. This command creates a new /boot and /boot/grub/ directory

```
sudo grub-install --no-floppy --root-directory=/media/casper-rw /dev/sdb
response [sudo] password for user:
response Installation finished. No error reported.
sudo chmod 777 -R /media/casper-rw
sudo chmod 777 -R /media/home-rw
sudo mkdir /media/casper-rw/boot/isos
```

The chmod commands relaxes the file permissions - it makes your experimentation a little easier. The mkdir command makes the directory into which the Kubuntu/Ubuntu iso's will be placed.

At the completion of this operation your casper-rw (/media/casper-rw/) will contain directories /boot/grub. Grub will be populated with a whole pile of grub stuff. All the '.mod' files you can see in /boot/grub are loadable Kernel modules that can be selectively added to the Kernel at boot time. Copy your Kubuntu/Ubuntu .iso's into /media/casper-rw/boot/isos.

Create a plain text Grub configuration file **grub.cfg** in **/media/casper-rw/boot/grub** and place in it the following text (substitute 'ubuntu' for kubuntu if you are using ubuntu).

```
set timeout=10
set default=0

menuentry "Run Kubuntu 10.04" {
loopback loop (hd0,1)/boot/isos/kubuntu-10.04-desktop-i386.iso
set root=(loop)
linux /casper/vmlinuz boot=casper iso-scan/filename=/boot/isos/kubuntu-10.04-desktop-i386.iso splash --
initrd /casper/initrd.lz
persistent
}
```

The information on the line following the command 'linux' are called *kernel parameter boot prompts*, *boot time parameters*, *boot prompts* or *cheat codes* they are passed to the kernel at boot time to modify the

kernel installation. Some common boot prompts can be seen using the terminal command 'man bootparam'

You are now ready to boot Kubuntu/Ubuntu from your casper-rw.

If your computer is like me and my computer a little on the ancient side, it may not have a BIOS option to 'Boot from USB' if that is the case get a copy of Plop boot CD. Boot into plop on a CD and that will give you the option to boot from USB.

**Plop usb boot cd**

**<http://www.plop.at/en/bootmanager.html>**