# Linux Installation Guide

©2015 Norman Carver  
Computer Science Department, SIUC

## Contents

1 Background: BIOS/EFI, Partitions, Filesystems, Bootloaders  
   1.1 BIOS vs. UEFI ........................................ 2
   1.2 Advanced Format Harddrives .......................... 2
   1.3 Traditional “MBR” Disk Partitioning .................. 2
   1.4 GUID Partition Tables (GPTs) .......................... 3
   1.5 How Disks and Partitions are Identified .............. 3
   1.6 Linux Disk Partitioning Schemes ..................... 4
   1.7 Physical Filesystems ................................ 6
   1.8 Logical Volume Manager (LVM) ........................ 6
   1.9 RAID and Encryption ................................. 6
   1.10 Bootloaders (MBR) .................................... 7
   1.11 UEFI Bootloaders .................................... 8
   1.12 UEFI Secure Boot .................................... 9

2 Prior to Installation .................................. 10
   2.1 Obtain Information about Current Partitions and OS Installations .............. 10
   2.2 Determine Desired Disk Partition Layout ................ 13
   2.3 Optional: Partition Disks Prior to Installation .................. 14
   2.4 Determine Desired Bootloader Configuration ............... 15
   2.5 Obtain Network Parameters ............................ 16
2.6 Optional: Gather Hardware Information .............................................. 16

3 The Installation Process ................................................................. 18
  3.1 Partition Selection and Formatting .............................................. 18
  3.2 Bootloader Selection and Installation ....................................... 18
  3.3 Software Selection .................................................................. 19
  3.4 Root and User Accounts .......................................................... 19
  3.5 Graphics Configuration ............................................................ 20
  3.6 Network Configuration .............................................................. 21
  3.7 Servers and Services ................................................................. 21
  3.8 Firewall and Security Settings .................................................. 22

4 Post-Installation Verification and Configuration .............................. 23
  4.1 Graphics Troubleshooting .......................................................... 23
  4.2 Verify Correct Partitions are Mounted ....................................... 23
  4.3 Secure Servers and Services ...................................................... 24
    4.3.1 Verify Firewall ................................................................. 24
    4.3.2 Verify that Only Desired Servers are Running ....................... 24
    4.3.3 Limit Access to Servers .................................................... 24
  4.4 Verify Network Connection(s) ................................................... 25
    4.4.1 Networking Troubleshooting .............................................. 26
1 Background: BIOS/EFI, Partitions, Filesystems, Bootloaders

1.1 BIOS vs. UEFI

In a standard “PC” type computer, the BIOS (Basic Input Output System), is responsible for basic hardware testing, initiating the boot up process, and various interactions between an OS kernel and PC hardware. The BIOS initiates the boot process by locating bootloader machine code in the Master Boot Record (MBR), and running that code (see Section 1.10). We will concentrate on these sorts of systems in this document, since they are still common. However, the traditional PC BIOS is being replaced by a new system known as the Unified Extensible Firmware Interface (UEFI or often just EFI). The UEFI standard was initiated by Intel, with a goal of replacing the traditional PC BIOS system. Most new PCs are UEFI capable, since Microsoft has mandated its use for Windows 8.

The main importance of UEFI when installing Linux on a UEFI machine is that UEFI changes the way bootloading must be set up. Booting an OS with UEFI requires that there be a separate partition that contains an UEFI-compatible bootloader program. See Section 1.11 for information on UEFI booting. UEFI machines also include a feature referred to as secure boot, which prevents the loading of bootloaders or drivers unless they have been appropriately signed. While many Linux distros can now deal with secure boot, not all can. Use of secure boot is mandated by Microsoft for Windows 8 on. See Section 1.12 for information on UEFI secure boot.

Another difference with UEFI systems is that UEFI supports a new partitioning scheme for harddrives: GUID Partition Table (GPT). The traditional PC partitioning scheme—now referred to as the MBR scheme—has a number of limitations that can make it difficult to handle large harddrives. While some UEFI systems can deal with both MBR and GPT disks, BIOS systems can handle only MBR. See Section 1.4 for information on GPT.

1.2 Advanced Format Harddrives

Harddrives have traditionally used sectors of 512 bytes, where sectors are essentially the smallest data units that can be addressed. As harddrives have increased in size, the use of 512 byte sectors has become a problem, so the Advanced Format standard was developed. AF harddrives use 4kB (4096B) sectors (or larger). Many current large (>2TB) harddrives are AF drives. One must be careful when partitioning AF harddrives, because if partition boundaries are not properly set, performance can be degraded. The problem will occur if the 4kB data blocks being read by the OS cross the 4kB sector disk boundaries, since then two disk accesses will be required instead of one. This problem arises because many AF harddrives present 512 byte sectors to the OS even though they are physically using 4 kB sectors. In this case, it is important that the sector number that starts a partition be divisible by eight. The latest versions of most partition tools have been updated to adhere to the 4k boundary requirement, but older versions do not, so it is a good idea to verify that you partitions are set up properly. Note that some AF drives report their sector size as 4k, and then alignment is not an issue.

1.3 Traditional “MBR” Disk Partitioning

The standard PC harddrive partitioning scheme that originated with the IBM PC is now often referred to as the MBR partitioning scheme/format. It supports a maximum of four primary partitions, one of which can be replaced with an extended partition. The extended partition can be subdivided into any number of logical partitions (though many OS’s place limits on the number of partitions allowed on a single disk).
The primary/extended partitions are numbered from 1 to 4, while logical partitions are numbered starting at 5.

The MBR scheme’s limitation of four primary partitions can be a problem for Linux. It is not uncommon for Windows PCs to come with the disk already partitioned with four partitions, meaning that even if the partitions are resized to make free space, it is not easy to add additional partitions for Linux (one cannot simply change a primary partition into an extended partition). On such machines, installing Linux will generally require destroying one of the less important partitions (recovery, test utilities, etc.).

An even more critical issue is the rapidly increasing capacity of harddrives. The MBR scheme uses 32-bit words to store the partition sector boundaries. This limits it to handling harddrives with a maximum capacity of around 2 TB (when the disk presents 512 byte sectors). Handling such drives requires use of the newer GPT partitioning scheme—see Section 1.4. (Note that the MBR scheme can handle >2 TB drives if the drive presents 4K sectors—see Section 1.2.)

1.4 GUID Partition Tables (GPTs)

One element of UEFI is a new disk partitioning scheme, called the GUID Partition Table (GPT). The GPT scheme can be used in place of the MBR scheme on machines that support UEFI. GPT does not have the four primary partition limit. Instead, the standard requires that at least 128 partitions can be supported. With the MBR scheme a disk might be partitioned as follows: two primary partitions (partitions 1 and 2), one extended partition (partition 3), and four logical partitions (partitions 5 through 8). With the GPT scheme we could similarly partition the disk, but simply numbered the six partitions 1 through 6.

Some recent Linux distros will give the option of using the GPT scheme to partition the disk. Be aware, though, that the traditional fdisk family of partitioning tools do not work with GPT. Instead, one can use gdisk (GPT fdisk) or parted (and its derivatives). Recent Linux kernels can all boot from GPT disks, whether the machine has UEFI or traditional BIOS. See Wikipedia for more information on how GPT is structured.

Note that unlike the two-hex-char partition type codes used in the MBR partition scheme (see Sections 2.1 and 2.3), GPT uses Globally Unique Identifiers (GUIDs) to denote partition types. E.g., the GUID for a Linux filesystem partition is: 0FC63DAF-8483-4772-8E79-3D69D8477DE4.

1.5 How Disks and Partitions are Identified

Most of the time you will probably see disks and partition identified by their device names. Most computers now have SATA (serial ATA) harddrives. The naming scheme for SATA drives and partitions is as follows: /dev/sd xn, where x is a letter that denotes the drive order (“a” for the first drive, “b” second, etc.), and n is an integer that denotes the partition number. Numbers 1–4 are used for the possible four primary/extended partitions. Numbers 5 and up are used for logical partitions (inside of an extended partition). So, /dev/sda is the first drive, /dev/sda1 is a primary/extended partition on the first drive, and /dev/sda5 is the first logical partition on the first drive. GPT partitions are simply sequentially numbered starting with 1, so partitions might be /dev/sda1, /dev/sda2, ... /dev/sda4, /dev/sda5, ...

The naming scheme for SCSI drives is exactly the same as for SATA drives. The naming scheme for older IDE/ATA drives and partitions is similar, but uses the prefix hd instead of sd. So the first IDE/ATA drive is /dev/ hda, the second /dev/hdb, etc., with the partition numbering scheme being identical to SATA drives. One additional thing to note is that while IDE/ATA CDROM/DVDROM drives use the same naming scheme as for harddrives, SATA/SCSI CDROM/DVDROM drives are named /dev/sr0, /dev/sr1, etc.
As an alternative to device names, you may also see partitions identified by their UUIDs or by their disk labels. UUIDs (Universally Unique Identifiers) provide a mechanism for giving a partition/drive a unique and unchanging identifier. Consider that if you were to accidentally swap the disk controllers your drives were plugged into while working on your computer, you could easily have /dev/sda become /dev/sdb and vice versa. If the device names were used in various configuration files, your machine likely would no longer boot. By contrast, each disk’s UUID would stay constant and allow a particular drive to be identified. The downside of UUIDs is that they are hard to read and remember: “A UUID is a 16 byte (128-bit) number. The number of theoretically possible UUIDs is therefore about 3 × 10^12. In its canonical form, a UUID consists of 32 hexadecimal digits, displayed in 5 groups separated by hyphens, in the form 8-4-4-12 for a total of 36 characters (32 digits and 4 hyphens). For example: 550e8400-e29b-41d4-a716-446655400000” [Wikipedia]

A disk label is a symbolic name that can be inserted into a filesystem and used in some contexts to identify/specify a partition. They can be particularly helpful in determining what is installed on a partition. For example, you could label a partition holding a Mageia 2 root filesystem as $MA2\_root$, while another partition holding a complete Ubuntu could be labeled $Ubuntu$. Note that different filesystems have different constraints on label length and allowed characters, so check what is allowed before choosing a label. ext filesystems can be labeled with tune2fs and the -L option.

### 1.6 Linux Disk Partitioning Schemes

In order to install Linux, you will need at least one free disk partition, for the root filesystem—i.e., the physical filesystem containing / (the root directory). Typically, at least one additional partition will be needed to provide swap space. Thus, a Linux installation generally requires there be at least two free partitions (of appropriate sizes). It is possible to reduce this requirement to a single partition, since Linux can use a swap file within the root filesystem (like Windows does) rather than a separate swap partition. This has always been considerably slower so was virtually never done, but according to the “Paging” entry on Wikipedia, with modern Linux kernels there is little/no speed difference between a swap partition and a swap file (assuming the partition is on the same drive as the root partition), and files allow more flexibility in allocating swap space. Nonetheless, it is unlikely there are any Linux installers that can setup swap files, so his would have to be done manually/post-installation. Note that on UEFI systems, a separate (third) partition will also be required for the bootloader code (see Section 1.11).

There are a variety of theories about optimal size for a swap partition, but you should remember that the maximum (in use) address space for all of your running programs will be limited to the sum of your RAM and your swap space. If this amount is too small, programs may be unable to be started when many programs are already running. These days, one should have at least 3-4 GB in RAM plus swap, and even more is desirable if you leave multiple programs running for extended periods. However, if too much of the address space is swap, your system can become very slow (e.g., 1 GB of RAM and 7 GB of swap will not produce performance even remotely similar to a system with 4 GB of RAM and 4 GB of swap). Common suggestions for swap size are to make it 1-2 times the amount of RAM you have. As the amount of RAM on systems increases, however, this makes less sense. In fact, if you have 4 GB or more of RAM, you may need minimal (or even no) swap.

If there is going to be a Windows installation on the machine, it generally needs to be in the first primary partition on the first harddrive (i.e., /dev/sda1 or /dev/sda1). Since Linux can very happily boot and run from any disk and from logical partitions, it makes sense to create an extended partition and place any Linux partitions as logical partitions inside of the extended partition. This avoids problems with the MBR scheme limit of four primary/extended partitions. So a typical install using a single root partition plus swap partition would have the swap partition be, say, sdx5, and the root partition be, say, sdx6 (where x denotes the disk with space for use with Linux).
While Linux can be installed using a single root partition (plus swap partition), there are a number of situations in which it is better to break the (logical) filesystem into multiple partitions. If one searches on the Internet, one will find many different suggested partitioning schemes. How the disk(s) should best be partitioned ultimately depends on what OS’s are going to be installed and how the machine is going to be used (server vs. workstation, single user vs. multiuser, etc.). If a Linux installation is to be done into a single partition, this partition will be mounted as / (root) and contain the entire (logical) filesystem. Other partitioning schemes have a / (root) partition, but then add additional partitions for other subtrees of the (logical) filesystem.

Typical subtrees to be placed in their own partitions are:

- **/home** — allows easier backups and reuse of personal files in multiple Linux installs, plus helps avoid problems filling the root filesystem;
- **/var** — important in multiuser system, to protect the root filesystem from being filled by user or log files and crashing the system;
- **/boot** — contains the kernel images and bootloader files, there are several reasons why it can be advantageous to use a separate partition (see discussion below).

Other subtrees may generally be placed in their own partitions as desired. However, note that placing /usr on a separate partition from / can potentially cause problems with some setups, so this is not a good idea.

Generally 5 GB for / is about the minimum that might work for a Linux installation, and 10 GB will provide enough room for a fairly complete system. The best size for a separate /home partition will depend on how many users and what sort of files they want to store; prior to users storing files, /home will be very small. A /var partition can probably be as small as 1 GB, and /boot partitions are generally 200-250 MBs.

Having /boot partition was once critical due to the inability of bootloaders to boot from kernel images beyond the 1024 cylinder boundary on harddrives, but this has not an issue for been many, many years (though you still see it mentioned). However, using a separate /boot partition may be required when using LVM (Section 1.8) RAID or encrypted filesystems (Section 1.9), or non-ext filesystems, since bootloaders may not be able to read such filesystems. Some distros (E.g., Redhat and Fedora) routinely create /boot partitions.

Remember that whenever you divide the disk into multiple partitions (and sub-filesystems), you reduce flexibility in using disk space. For instance, the /home partition/filesystem could become filled even when there is plenty of space on the root filesystem. Thus, you should avoid breaking your installation filesystem into too many partitions unless you have a good reason for doing this and can reliably predict disk usage. This is particularly true since it is usually difficult to modify disk partitions without disturbing existing filesystems. On the other hand, if a filesystem should become filled while there is a larger partition free on a disk, it is relatively easy with Linux to move the filesystem to the larger partition.

If you intend to install a number of Linux distros on your disk, it can be a good idea to divide the disk into a set of identically sized logical partitions. This makes it easier to backup partitions (using dd or partimage), and gives more flexibility when experimenting with multiple Linux versions. A single swap partition can be shared by all of the installed distros.

There are some hardware-related speed considerations one may want to consider when partitioning. On most harddrives, the first/inner space on the disk can be accessed somewhat faster than that toward the end/outside of the disk. This suggests that the swap partition(s) should be placed in lower numbered cylinders (i.e., one of the first partitions). If your system has multiple harddrives, it also makes sense to have your swap partition(s) on a different disk(s) from the OS, and you will get even better performance if the swap drive(s) are connected to separate controllers from the OS drive(s). Linux can use multiple swap partitions, and the order they are used is based on the order they are listed in /etc/fstab.
1.7 Physical Filesystems

Creating partitions is only the first step in getting a disk ready for the installation. Partitions must be formatted before they can store files. Formatting involves setting up a physical filesystem. Linux supports a large number of physical filesystem types, and different distros will use different default filesystem types. At this time, the standard Linux filesystem used by default by most installers is ext4. An older alternative is ext3 (there is definitely no reason to use the even older, non-journalled ext2). ext4 has many advantages for large and multimedia files, and Google began switching all their systems to it from ext3 in early 2010. Most distros will also offer the alternative Linux filesystems XFS, JFS, and ReiserFS. Some will even offer the still experimental btrfs. Reportedly, Google benchmarked ext4, XFS, and JFS, and found ext4 and XFS to be equivalent in performance, but considered ext4 an easier upgrade. Certainly the "ext" filesystems are the safest to use in booting, since the "drivers" are built into the kernel. Other filesystem types may require use of a ramdisk (though this is usually set up automatically). A further consideration is whether one intends to be able to mount one Linux' filesystem on a different Linux install. Older, pre-ext4 Linuxes will not be able to mount ext4 filesystems, and older GRUBs (see Section 1.10) may not be able to read it.

1.8 Logical Volume Manager (LVM)

One option to deal with uncertainty about appropriate partition sizes is to use Linux' Logical Volume Manager (LVM). The LVM allows one to combine multiple disk partitions (across multiple disks even) into a single volume group (VG), which can then be divided into multiple logical volumes (LVs), which are effectively like partitions. Unlike standard disk partitions, however, LVs can be dynamically resized within the VG, allowing an administrator to reallocate disk space among filesystem subtrees. Many modern Linux installers are able to set a disk up to use LVM, and some will now do this by default.

It is also important to understand that while it is much easier to resize an LV than a disk partition, resizing an LV containing a filesystem requires resizing the filesystem, and that is neither quick nor risk-free. Thus, it is important to allocate space among filesystems such that resizing is unlikely to be required.

Another drawback of using LVs rather than partitions is that the /boot subtree cannot reside on an LVM volume because the GRUB Legacy boot loader cannot read it. This means that installers will generally set up multiple partitions, one of which is a /boot partition and one that is an LVM volume. GRUB2 have the ability to deal with LVM (though an appropriate module), but it may still be advisable to have a separate /boot partition if using LVM.

1.9 RAID and Encryption

The Linux kernel supports most RAID configurations without the use of special hardware. This is known as software RAID (vs. hardware RAID with special controllers). A number of Linux distro installers can create RAID arrays from multiple disks or partitions. RAID arrays can also be setup prior to installation or post installation. If you use software RAID for /, it is a good idea to have a separate /boot partition that is not RAID. While GRUB2 (with appropriate modules) can potentially boot from software RAID, getting it to work properly may be problematic.

Another option offered by some installers is to encrypt one or more of the partitions. There are multiple encryption tools for Linux, and different distro installers may use different tools. One of the most popular is Linux Unified Key Setup (LUKS), which offers encryption at the device level rather than the filesystem level. LUKS is implemented via the kernel device mapper subsystem module dm-crypt. If encrypting your installation, you will need a separate, unencrypted /boot partition.
1.10 Bootloaders (MBR)

A boot loader (or bootstrap loader) is code that is run by a PC’s BIOS to start the loading and execution of an OS kernel—i.e., it boots an OS. In this section we will concentrate on bootloading on BIOS systems; UEFI booting is covered below in Section 1.11.

All Linux installers can install a bootloader that is capable of booting multiple OS’s. Most distros use GNU’s GRUB (G rand Unified Bootloader) by default, though a few may still allow the older LILO (L inux Loader) to be installed. GRUB has many advantages over LILO if things go wrong with a system, so it is generally the better choice. Note, however, that GRUB relies on being able to read its configuration file from a filesystem during bootup, so this file may need to be in an ext filesystem (though some other filesystems may be supported by particular versions of GRUB). This requirement may cause one to need a separate partition for /boot. LILO does not have this requirement, since it finds necessary files from their positions on the disk (which is why you must reinstall LILO whenever any changes are made to kernels, etc.).

Actually there are two main versions of GRUB in current use. The older version, now known as GRUB Legacy (versions up to 0.97), and the newer version, known as GRUB2 (version numbers 1.98 and up). Currently, some distros use GRUB Legacy while others use GRUB2. While each should be able to chainload the other (see below), in practice there have been issues with various GRUB versions working together.

To understand bootloader installation options, one must know where bootloaders can be installed on hard-drives and how bootloaders are organized. The first sector of a hard drive is called the master boot record (MBR) or master boot sector. Harddrive sectors have historically been 512 bytes (though see about AF drives in Section 1.2). With standard BIOS-based PCs, in order to boot, the BIOS looks for and runs code found in the MBR to be able to boot the computer, there must be a bootloader installed in the MBR of the first hard drive. Since the MBR has traditionally been only 512 bytes, the code in the MBR must be very simple, but 512 bytes is not enough for a complete bootloader program. Because of this, bootloaders are typically divided into two (or more) components or stages. The first stage is simple code (≤512 bytes) that merely finds and loads the second stage. The second stage is what initiates the actual booting of an OS kernel. (Note that GRUB Legacy may use a stage between what it terms stage 1 and stage 2, which is termed stage 1.5.)

The code for a bootloader’s second stage may be relatively large, since it may have to have the ability to read multiple filesystems in order to get boot configuration information from various OS’ installations. The second stage of Linux bootloaders can be installed in either of two places: (1) between the MBR and the start of the first partition; (2) inside of an OS’ filesystem. The downside of installing stage 2 inside a filesystem is that the physical disk location of the stage code must be stored in stage 1 (since stage 1 cannot read filesystems), and some filesystem operations might move the stage 2 file, making the system unbootable. The GRUB team recommends option (1), but this requires that the first partition start at a sector well beyond 1. Starting the first partition at sector 63 leaves 31 kB of space for stage 2. This is generally sufficient, though that depends on the capabilities compiled into stage 2. The need for bootloader space is why partitioning tools (e.g., fdisk) often default to starting the first partition at sector 63 or even higher. Given the size of modern harddrives, it may make sense to leave around a megabyte for bootloaders. Thus, it is also common to see partitioning tools starting the first partition at (512-byte) sector 2048.

Linux bootloaders may automatically boot one OS, but more commonly they display a menu from which the user can select the OS (or OS version/kernel) to boot. Linux bootloaders can be setup to directly boot any of multiple Linuxes, but they can also boot any OS (including Windows) through a process called chain loading. Chain loading of an OS relies on the other OS having installed its own bootloader, which the primary bootloader simply transfers control to. Note that the first sector of a partition is is called the partition boot sector or partition boot record. To allow chain loading of an OS, one typically installs the OS’ bootloader into the boot sector of the OS’ “root partition” (the root partition means the partition that will hold the / filesystem). Selecting where to install the bootloader is one of the key
installation decisions that must be made. Unfortunately, while virtually all Linux installers will allow you to select to install the bootloader under the root partition of the installation, not all do this properly, so you can end up with an install that cannot be booted via chain loading! The usual problem is that the GRUB2 installation function will refuse to complete the install if its stage 2 (core.img) is inside a filesystem. This issue results in a message such as: “Embedding is not possible. GRUB can only be installed in this setup by using block lists.” Block lists refers to using physical disk addresses.

GRUB Legacy and GRUB2 have their configuration (and any other needed files) installed under /boot/grub (or maybe /boot/grub2 for GRUB2). The name of the default configuration file for GRUB Legacy varies among distributions, but is generally either /boot/grub/menu.lst or /boot/grub/grub.conf. This file is easy to understand and easy to edit by hand. GRUB2 uses multiple configuration files. The main configuration file is /boot/grub/grub.cfg, but unlike with GRUB Legacy, this file is not intended to be edited by hand. Rather, it should be generated using the command grub-mkconfig. The user can add menu entries through files in /etc/grub.d, typically 40_custom. An alternative is creating /boot/grub/custom.cfg. The file /etc/default/grub controls the operation of grub-mkconfig, and is normally a sequence of “KEY=value” lines.

On systems with harddrives partitioned using GPT, it is possible to reserve a partition for GRUB2, called the **BIOS Boot Partition**. GRUB2 can be installed into that partition without the risk of being overwritten or other filesystem issues. When creating a BIOS Boot Partition on a GPT system, it must be at least 31 kB in size, but it is good to make it larger. You must also make sure that it has the proper **partition type GUID**. The GPT GUID for a (BIOS) boot partition is: 21686148-6449-6eef-74e658545b49. Most partitioning tools will provide some type abbreviation: e.g., parted uses “bios_grub” while gdisk uses “0xEF02”.

### 1.11 UEFI Bootloaders

Bootloading on UEFI systems is quite different than with BIOS systems, and Linux support for UEFI booting remains incomplete and confusing. Not all distros are currently able to be installed on UEFI enabled systems. Those that can support UEFI booting may use differing approaches to accomplish this, and installers may not be able to completely automatically setup UEFI booting. Nonetheless, if you are installing Linux on a new machine, it makes sense to try to use UEFI booting. Since you cannot mix UEFI and MBR bootloaders, it makes sense to start a new machine off with what is going to be used in the future. Also, Windows 8 and on require UEFI booting.

With UEFI, the primary bootloader must be an UEFI-compatible program. It will be stored in a special **UEFI System Partition** (ESP). This partition will have GPT GUID of C12A7328-F81F-11D2-BA4B-00A0C93EC93B, MBR ID of 0xEF, and must be formatted with a FAT (vfat) filesystem. The partition must contain the directory EFI at toplevel. Within this directory are vendor-specific directories, such as: e.g., redhat, ubuntu, mageia (case appears not to matter). The UEFI system will look for bootloader code within these vendor directories.

There are a variety of Linux bootloaders that have been modified to be UEFI compatible. These include GRUB Legacy and GRUB2, plus: **ELILO** (from HP), **EFILINUX** (from Intel), and **rEFInd** (a fork of the rEFIt Boot Manager used in Intel Macs, by the author of gdisk). Different distros use different bootloaders for UEFI installs at this point.

The UEFI bootloader partition generally gets mounted during boot, typically under /boot/efi or /boot/EFI. This means that the UEFI bootloader files appear in the filesystem tree under a directory like /boot/efi/EFI/redhat.
1.12 UEFI Secure Boot

UEFI adds a feature known as secure boot, which aims to enhance PC security by preventing the loading of drivers or boot loaders that are not signed with an acceptable digital signature. Use of secure boot is mandated for Windows 8 and later.

It is important to be aware of secure boot because only a limited set of Linux distros are currently able to be installed and work with it. Those that can include: Fedora, Ubuntu, Mint, and OpenSuse. The exact approach used by different distros to handle secure boot can vary. Other distros may have only experimental support for secure boot or may require users to disable secure boot in the UEFI firmware in order to be able to boot the distro!

With Windows 8, manufacturers were required by Microsoft to include a method to disable secure boot on x86 systems (but not for ARM systems). Systems are also supposed to be able to allow users to add additional signing keys. With Windows 10, the ability to disable secure boot becomes optional. Once Windows 10 PCs become available, if secure boot cannot be disabled, this will drastically limit the Linux distros that can be installed on a box, so you must be aware of this!
2 Prior to Installation

Before beginning to install a Linux distro by booting its installer, it is a good idea to make certain one has the information that will be required during the install process. Otherwise, one risks wasting time by finding that the installation has to be aborted at some point so that necessary information can be acquired. In addition, it is sometimes better to set up the desired disk partitions prior to running the installer (e.g., to gain more precise control).

The key activities one should typically do prior to installation are:

1. Obtain information about current partitions and OS installations;
2. Determine the desired partition layout;
3. Partition the disks (optional);
4. Determine the desired bootloader configuration;
5. Acquire the network parameters;
6. Gather hardware information (optional);

2.1 Obtain Information about Current Partitions and OS Installations

One of the more critical decisions you will have to make during installation is what space/partitions on your disk(s) you want to install the new Linux to. If you are going to be installing Linux on a machine that already has one or more installed OS’s that you want to keep intact, it is extremely important to examine the current disk setup prior to installation, to avoid damaging existing installationa and losing data. This will require that you determine how the disks are already partitioned and what is on each partition (if anything).

If you already have a running Linux, there are at least six sources of information:

- Basic Partitioning commands/tools such as fdisk or gdisk.
- More sophisticated partitioning tools such as GNU Parted.
- Distribution-specific disk manipulation tools.
- The directory /dev/disk, which contains subdirectories that list disk partitions by various naming schemes: by-id, by-label, by-path, by-uuid.
- The commands df and mount, which will identify the partitions that are currently mounted and where they are mounted (e.g., / or /home).
- The file /etc/fstab, which contains a list of the disk partitions that are mountable from the particular Linux, along with their mount points.
- The bootloader configuration file (/boot/grub/menu.lst, /boot/grub/grub.conf, /boot/grub/grub.cfg or /etc/lilo.conf), which will identify the root partitions for different OS versions that can be booted by the bootloader installed by that Linux.

The most basic partitioning command/tool is fdisk. The command “fdisk -l” will list all of the partitions on all of the disks/devices. It is a good idea to become familiar with fdisk, as it may be the only partitioning tool available on a Linux rescue disk. Most (full) Linux systems also have the similar tools sfdisk and cfdisk as well. sfdisk can be useful as it can print partitions out in a format that it can also read in, making it easy to identically partition multiple drives. cfdisk is a GUI (CURSES) program. The commands gdisk, sgdisk, and cgdisk are comparable, but support GPT disks.
The output from “fdisk -l” will look something like:

```
Disk /dev/sda: 250.0 GB, 250000000000 bytes
255 heads, 63 sectors/track, 30394 cylinders, total 488281250 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x50000000

Device Boot Start   End     Blocks   Id  System
/dev/sda1   63     112454   56196    3  Dell Utility
/dev/sda2   112455  16900379 8393962+ c  W95 FAT32 (LBA)
/dev/sda3   * 16900380  17302004  200812+ 83 Linux
/dev/sda4   17302005 488279609 235488802+ 5 Extended
/dev/sda5   17302068 488279609 235488877+ 8e Linux LVM
```

```
Disk /dev/sdb: 750.2 GB, 750156374016 bytes
255 heads, 63 sectors/track, 91201 cylinders, total 1465149168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa2ab77d5

Device Boot Start   End     Blocks   Id  System
/dev/sdb1   63     1465144064 732572000+ 5 Extended
/dev/sdb5   63     8418059  4208966+  82 Linux swap / Solaris
/dev/sdb6   8418123  50347709 20964793+  83 Linux
/dev/sdb7   50347773  92277359 20964793+  83 Linux
/dev/sdb8   92277423 134207009 20964793+  83 Linux
/dev/sdb9   134207073 176136659 20964793+  83 Linux
/dev/sdb10  176136723 218066309 20964793+  83 Linux
/dev/sdb11  218066373 259995959 20964793+  83 Linux
/dev/sdb12  259996023 301925609 20964793+  83 Linux
/dev/sdb13  301925673 1465144064 581609196  83 Linux
```

`fdisk` now shows the Start/End values for partitions in terms of the disk sectors rather than cylinders (as it used to). Since all modern harddrives use Logical Block Addressing (LBA), cylinders are no longer relevant (though some tools may still complain if partitions are not aligned to cylinder boundaries). The Block values give the size of each partition in terms of 1024 bytes (1k) blocks. Thus, 4,208,966 is about four million 1k byte blocks, or around 4 GB. Partition types are identified with a two hex digit code shown under the Id column, and written in text under System. While this should give one an idea of the purpose of the partition, this is not completely enforced (so one could potentially install an ext3 filesystem in the “W95 FAT32” partition). The code used for Linux filesystem partitions is 83, and 82 is used for Linux swap partitions.

GNU Parted provides the partitioning capabilities of the “fdisk programs” as well as the ability to understand and manipulate several filesystems. It can be accessed at the command line as `parted`, or via various GUIs such as `gparted` (GNOME Parted). The command “parted -1” will list all partitions on all disks/devices. It differs from “fdisk -l” in that `parted` understands filesystems, so can tell you what filesystems are on partitions rather than just what the partition type codes are.
The output from “parted -l” will look something like:

Model: ATA ST3250310AS (scsi)
Disk /dev/sda: 250GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

<table>
<thead>
<tr>
<th>Number</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
<th>Type</th>
<th>File system</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32.3KB</td>
<td>57.6MB</td>
<td>57.5MB</td>
<td>primary</td>
<td>fat16</td>
<td>diag</td>
</tr>
<tr>
<td>2</td>
<td>57.6MB</td>
<td>8653MB</td>
<td>8595MB</td>
<td>primary</td>
<td>fat32</td>
<td>lba</td>
</tr>
<tr>
<td>3</td>
<td>8653MB</td>
<td>8859MB</td>
<td>206MB</td>
<td>primary</td>
<td>ext3</td>
<td>boot</td>
</tr>
<tr>
<td>4</td>
<td>8859MB</td>
<td>250GB</td>
<td>241GB</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8859MB</td>
<td>250GB</td>
<td>241GB</td>
<td>logical</td>
<td>lvm</td>
<td></td>
</tr>
</tbody>
</table>

Model: ATA ST3750330AS (scsi)
Disk /dev/sdb: 750GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

<table>
<thead>
<tr>
<th>Number</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
<th>Type</th>
<th>File system</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32.3KB</td>
<td>750GB</td>
<td>750GB</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32.3KB</td>
<td>4310MB</td>
<td>4310MB</td>
<td>logical</td>
<td>linux-swap(v1)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4310MB</td>
<td>25.8GB</td>
<td>21.5GB</td>
<td>logical</td>
<td>ext3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>25.8GB</td>
<td>47.2GB</td>
<td>21.5GB</td>
<td>logical</td>
<td>ext3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>47.2GB</td>
<td>68.7GB</td>
<td>21.5GB</td>
<td>logical</td>
<td>ext4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>68.7GB</td>
<td>90.2GB</td>
<td>21.5GB</td>
<td>logical</td>
<td>ext4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>90.2GB</td>
<td>112GB</td>
<td>21.5GB</td>
<td>logical</td>
<td>ext4</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>112GB</td>
<td>133GB</td>
<td>21.5GB</td>
<td>logical</td>
<td>ext4</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>133GB</td>
<td>155GB</td>
<td>21.5GB</td>
<td>logical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>155GB</td>
<td>750GB</td>
<td>596GB</td>
<td>logical</td>
<td>ext3</td>
<td></td>
</tr>
</tbody>
</table>

One of the main advantages parted has over fdisk is that is shows the actual filesystems that are installed in the partitions. For example, with sdb we can see that some of the Linux partitions have ext3, some have ext4, and one (sdb12) does not contain a filesystem.

If you do not know what (if anything) is in some partitions, you can mount each partition one by one, and see what is there. For example, if you mount a partition, do ls on it, and see something like “bin boot dev etc...” then you have a / (root) partition. If you see something like “carver lost+found” then you have a /home partition. If you get a vfat or ntfs partition, when you obviously have a Windows partition. Mount errors involving unknown filetype generally indicate an unused (unformatted) partition. When you find a / (root) partition and are not certain what Linux it is, you can generally check by looking for a file in /etc that tells. Typical filenames will include the substring “release” or “version” as in: /etc/mandriva-release or /etc/debian_version. Linuxes that adhere to the LSB should have a file: /etc/lsb-release.

To mount and examine a partition do:

1. Create a mount point: mdir /mnt/disk
2. Mount the partition: mount /dev/sda6 /mnt/disk
3. See what is on it: e.g., ls -dl /mnt/disk
4. Unmount it when done: umount /mnt/disk
Free space on a disk (i.e., space that has not been placed into a partition), can be found from the information supplied by the `fdisk` and `parted` commands, though it can require looking carefully at sector start/end values. `parted` and most distro GUI disk tools will make free space more obvious.

If you do not have a (running) Linux already installed on the machine, you can do much of what has been described above from a live Linux CD or from the “rescue mode” or “console mode” provided by most Linux installer discs. You can also use a running Windows installation to get information about which partitions Windows is using and whether there is free space on the disk, from Windows’ *Disk Management* tool. Since this tool knows nothing about non-Windows filesystems, it is of limited value with disks that already include non-Windows OS's.

Note that if you must resize an installation in order to make free space, this requires first shrinking the filesystem on the partition and then shrinking the partition. Before one can shrink a filesystem, the amount of space on the filesystem that is already in use must be determined, as this gives a lower bound on how big the resized partition will have to be (plus there will probably need to be some additional space for continued use). Automated resizing tools will generally safely limit how small you can make a partition, but manual (command line) programs can require that you figure out sizes.

## 2.2 Determine Desired Disk Partition Layout

Virtually all Linux installers now have automated partition selection routines and some can even resize existing filesystems. However, you may not end up with what is best if you simply use the automated routines.

Your primary task is to find sufficient space on your disks to install Linux. Exactly how much space is required will depend on the distro and what software you want to install. 5 GB is about the minimum that might work, while 10 GB will typically provide enough room for a fairly complete system (though not much user space). For a system you actually intend to work on (i.e., save files on), with both KDE and GNOME installed, 20 GB would be very good. You will typically also need 1–4 GB for a *swap partition*.

The space for your new installation can come from a combination of:

- a newly added disk;
- existing free space on the disk(s);
- existing unused partitions;
- resizing filesystem(s) on used and wanted partitions.

Linux can boot from any partition on any drive (unlike Windows). This allows for quite a bit of flexibility in selecting space. Often people add a second (smaller) drive and put Linux on that so that an existing Windows installation is not disturbed at all. Sufficient existing free space or unused existing partitions are also simple. The last of the above sources represents the most difficult and risky situation, but it is also the most common with a machine that currently has only Windows installed. See Section 2.3 for further information on resizing filesystems/partitions.

Once enough free space has been identified, you will need to consider how to partition it. Typical partitioning schemes are discussed in Section 1.6. If the space to be used is not contiguous, you may still be able to use it in place if you can match up the space with an appropriate partitioning scheme. For example, if there is 10 GB free in one place on the disk, and 5 GB elsewhere, one could create partitions for each, placing the root file system on the 10 GB partition and using the 5 GB partition for `/home` or similar.

It is possible that you could find available free space that is broken up into too small sizes to be useful (e.g., two 5 GB regions). In this case, partitions/filesystems will have to be moved in order to combine
the individual free space regions. While some Linux installers can do this, you probably will want to use a separate tool to prepare the disk prior to installation. Note that moving partitions obviously makes the entire installation process risker, and moving large partitions can be a very time consuming procedure. *It is imperative that all critical data be backed up before moving or resizing filesystems!*

### 2.3 Optional: Partition Disks Prior to Installation

If you find that there are appropriate partitions or free space for your installation (or if you are going to wipe a disk clean for the install), then you are probably all set, as most installers can deal with these situations. If not, you may want to prepare the disk partitions before proceeding to run the installer.

Virtually all Linux installers can do the following:
- wipe disk(s) clean and install automatically;
- use one or more existing partitions for installation;
- create new partition(s) in free space for installation;
- delete existing partitions to create free space;
- resize partitions (without saving their contents).

The most difficult situation is when you need to resize a partition with an existing filesystem on it (that you want to save) to make free space for the Linux installation. If this is the situation you find yourself in, you will want to investigate whether the installer for your chosen distro(s) can do what you want. These days many Linux installers can resize virtually any type of filesystem, but some can only resize Linux ext filesystems, and some still cannot do resizing. Be aware that resizing a filesystem is always somewhat risky, as a power failure or software bug could result in the entire filesystem/OS being unusable. Thus, whenever one is going to resize a filesystem, it *is imperative that all critical data be backed up first!*

Starting with Vista, Windows finally includes a tool for resizing its own (NTFS) partitions. This has to be considered the *preferred method for resizing Windows partitions* (Linux NTFS resizing capability comes from reverse engineering NTFS, NTFS is proprietary). Obviously, the Windows resizing tool would have to be used prior to running the Linux installer. Resizing the Windows filesystem/partition will open up free space on the disk, which can then be used by the Linux installer. Windows' resizing functionality can be accessed via the *Disk Management tool*. Unfortunately, if you look at comments online, you will find that Windows sometimes refuses to resize its partitions as people desire. You may want to read about disabling the Windows “page file” (swap file) prior to resizing. It is also generally necessary/wise to *defragment* a Windows filesystem before attempting to resize it.

Even when the installer for your distro(s) can resize your partitionss/filesystems, you may have more control over partitioning if you do it ahead of time using other tools. For example, if you want a set of identically-sized partitions or want specific use of primary vs. logical partitions, this can be difficult to achieve with many installer partitioning tools. There are a range of tools available for partitioning disks prior to running an installer depending on whether you are going to be running them from a different Linux installation, a live Linux disc, a Linux installer’s “rescue mode,” or some other rescue disc, and depending on whether you want to use command line or GUI tools.

The *fdisk* (and *gdisk*) command is the tool most likely to be available (perhaps with *sfdisk* and even *cfdisk*) under virtually any of the above scenarios, so it is important for systems administrators to be familiar with its use. *fdisk* gives you great control over partitioning, but also requires that you understand a fair amount about how disks are partitioned. It deals with only partitioning of disks, so cannot be used to resize filesystems along with partitions. To modify the partitions on a disk, one uses a command like “*fdisk /dev/sda*” to start editing the *partition table* for disk /dev/sda. The list of (one letter) commands can be found by typing “m” (for menu).
These tools will generally set the type of primary and logical partitions to 83, meaning it is to contain a Linux filesystem. It is important to change the ID/type code for the swap partition to 82, as this is what installers will detect to determine what partition to use for swap.

If one is limited to simple command line tools such as fdisk, resizing of filesystems will have to be done using the appropriate filesystem-specific tool. For ext filesystems, that command/tool is resize2fs. Using this command is somewhat complicated due to the need to resize a filesystem first and then resize its containing partition appropriately, so look up howtos online. There are similar Linux tools/commands for the other Linux filesystems, as well as FAT and NTFS.

GNU Parted (parted) or one of its GUI versions, such as GNOME Parted (gparted), provide integrated partitioning and resizing capabilities. (Note, though, the resizing relies on the same filesystem-specific tools listed above, so these must be installed/available.) Some Linux distros provide their own tools similar in function to GParted. For example, the Mandriva/Mageia installer's disk partitioning tool is available as a standalone program: drakdisk. To get full control, you need to select “Custom Disk Partitioning.” From here, you can modify partitions (create, delete, resize), and create/resize filesystems on the partitions.

Note: you cannot (or at least generally should not) modify partitions/filesystems containing an OS you are running (or have mounted). Use a rescue/live disc or an installation program to do partitioning/resizing.

### 2.4 Determine Desired Bootloader Configuration

Virtually all installers will setup a bootloader and install it into the MBR by default. If the newly installed Linux will be your only Linux, then this is almost certainly what you want. Most Linux installers scan existing partitions for installed OS’s and will create bootloader menu entries for all they find—including Windows. The situation can become more complicated when you are installing an additional Linux and want to continue to be able to boot some of the already installed Linuxes. Unless you make appropriate choices in the installer, your existing MBR bootloader will probably be overwritten by the installer. Most installers will setup the new bootloader to include a menu entry for existing Linuxes they find, but typically they will try to use chainloading to boot the other Linuxes (some installers now do setup GRUB configurations to directly boot other Linux installs If the previous Linux was booted via the MBR only, chainloading will fail.

There are two possible solutions. One solution is to have the new Linux install its bootloader into its root partition rather than the MBR, then post install, boot into the older Linux (that had written the MBR bootloader) and edit its configuration file to be able to chain load the new Linux. The second solution is to have the new Linux overwrite the MBR, make sure its bootloader is set up to be able to chainload the old Linux’s root partition, and before the install, make certain the old Linux’s bootloader is duplicated in its root partition (install its bootloader into its root partition as well as the MBR). (Actually, with GRUB, it is usually quite easy to do this step post install, from the new Linux.)

The bottom line is that before you run an installer you need to determine where the installer should place the bootloader for the new Linux: (1) the MBR, or (2) its root partition. If you are going to overwrite an existing Linux’s bootloader in the MBR, you should duplicate that bootloader into the Linux’ root partition prior to starting the install of the new Linux. You should also think about what entries you want in the new Linux’ boot menu, as you may be offered a chance to edit these selections.
2.5 Obtain Network Parameters

These days most Linux installers will automatically set up whatever network hardware they can, but they will set it up to automatically get network parameters via DHCP. This may not be the correct or best setup for your network. You need to ask the network/system administrator how you should configure your machine. This is particularly important to do, as incorrect network settings can wreak havoc on a network, making the administrator very unhappy with you!

The first thing one needs to know to configure a network interface is whether it should use DHCP to automatically/dynamically obtain network parameters or whether it should use a static network configuration. Some Linux installers will simply set up network interfaces to use DHCP and if you need a static you will have to do it post install. Others will give you the option to set up the correct static parameters as part of the install.

If the configuration is to be static, you will need to know the following five pieces of information in order to complete the configuration:

- *hostname* for the machine;
- IP address for the machine;
- IP address of the *gateway* machine;
- *netmask*;
- IP address(es) for 1–3 DNS servers.

For wireless networking, you will need to know the *SSID*, type of security (encryption) being used, and the encryption key/password.

2.6 Optional: Gather Hardware Information

Before doing an install, it is a good idea to gather some information about the hardware on your system. Modern Linux installers generally do an excellent job dealing with most hardware, but sometimes there can be questions that arise about the best drivers or settings.

If you have documentation that came with the system, that is one source of information. If you already have Windows running on the machine, you can use its Device Manager to gather such information. If you have another Linux running, you can use several tools to gather information. Often a distribution will have a GUI tool to list hardware. Command line programs of interest include: lshw, lspci, and xdpyinfo.

The most important hardware to know about:

- graphics card;
- monitor;
- network interfaces (NIC, wireless, modem);
- sound card/onsboard sound.

If you have problems with graphics/monitor setup during the installation, you may not be able to come up in the GUI/X11, making it harder to fix things (since the GUI tools will not be able to be used). For the graphics card, you should know the manufacturer, model and possibly the amount of memory on the card (less important now). For the monitor, if it is an LCD, all you probably need to know is its native resolution (LCDs all work fine with refresh rates from 50–60 Hz). If the monitor is a CRT, you should ideally know the manufacturer, model, desired resolution, and refresh rate information. Luckily X11 tends to automatically produce an at least minimally usable setup these days. However, post-install effort can still be required—particularly if you wish to use *proprietary* graphics drivers (see Section 3.5).
Most NICs (i.e., wired network interface cards) will be automatically detected and set up by modern Linuxes. *Wireless cards can still cause problems,* however. First, there are simply not Linux drivers for every wireless card, and second, even if there are drivers, many cards require that *firmware* be loaded. The problem with firmware is that it is often under licenses that prohibit distribution by free Linux distros. Thus, *if you are going to be relying on wireless networking, it is critical to investigate whether it will be able to function with the distribution you intend to install.* Many distros have hardware databases that list known working hardware, and many distros also have live CDROM/DVDROM versions that you can use to test your hardware. If your wireless card requires firmware that cannot be distributed as part of the installer, most distros will have online repositories that will allow you to get the required firmware package. Of course if your only Internet access is via the unworking wireless card, you will be in trouble. If you will not have access to a wired network interface, you will need to try to figure out what package contains the required firmware, and download it to a USB stick or similar prior to installation. If you are unfortunate to have a wireless card that lacks a working Linux driver, you may be able to use the Windows driver via something called *ndiswrapper.* Many Linux installers and/or network configuration tools can set this up, but you will need the Windows driver file to be available.

Most “onboard” sound devices adhere to the *AC’97* standard and will work fine with Linux. Separate sound cards can be an issue due to lack of drivers, so it is important to check. Most Linux distros use *ALSA* (Advanced Linux Sound Architecture), so the first place to check for support is the ALSA website.

Modems are another problematic type of hardware under Linux, as many are designed to work only with Windows. These are the so called “Winmodems” or “softmodems.” In fact, you could argue that these are not actually “modems” at all, but merely interfaces to a phone line, with all the “modem” processing being done via a Windows-specific driver. This is why they are so cheap—they do almost nothing in hardware. By contrast, most true hardware modems, when you can find them, work just fine in Linux. Few new computers come with modems anymore, but one can still be useful for faxing.
3 The Installation Process

This section will cover the primary issues that may be encountered during an installation. Linux installers can vary a great deal from one another. Some are highly automated and simplified, giving you very little control over how the installation is configured, so they may not allow you to make decisions about all of the issues discussed below. Others may allow you a great deal of control, but only if you choose “advanced” or “expert” options in certain steps. Still others always make you decide on nearly every option. If an installer does not allow you to configure some component as desired, you can always use appropriate tools post installation.

One thing to consider during the installation is whether you want the machine connected to a network or not during the installation (most of the time this must be a wired connection, though some “live” install discs can support WiFi connections). The advantage of having a connection is that some installers can test your network connection and even use it to install updates or additional software during the installation process. The disadvantage is that many installers do not leave a machine fully secured, so having the machine connected to the Internet could potentially allow the machine to be attacked before you get a chance to secure it! If your machine is going to be directly connected to the Internet (i.e., not behind a hardware firewall), it is a better idea to perform installations without a network connection. You can easily carry out any of the network-related actions that might be done during an install once the machine has been secured post install.

3.1 Partition Selection and Formatting

*Setting up partitions on a machine with existing OS’s that you want to save is one of the most critical steps in the installation—so be careful!* Make certain that you have at least read Sections 1.6 and 2.2, and have decided on a partition and filesystem layout in advance. The partitioning component of an installer will generally be encountered fairly early in the installation process.

Since there can be a great deal of variation in how disks are set up, all installers will give you some control over what partitions and/or disk space gets used for the installation. Most will also include automated options that involve wiping out all existing OS’s or using existing partitions. With some installers, taking control of the process will require selecting an advanced/expert/custom mode. Many installers probe the disk(s) for existing filesystems/OS’s, and offer you options based on what is found on your particular disk.

3.2 Bootloader Selection and Installation

*Mistakes installing the bootloader can result in some of your OS’s being unable to be booted* (including the new Linux), so care is needed in this stage. Unlike partitioning errors that can destroy an OS and your data, however, bootloader errors can generally be corrected post-install (if you have enough technical knowledge). Still, you should read Section 2.4 and have decided on a bootloader layout prior to starting an install.

There is considerable variation in bootloader configuration options presented by various Linux installers. Nearly all distros now use GRUB (Legacy or GRUB2) by default, but a few of these will still allow installation of LILO. As noted earlier, GRUB is nearly always a better choice than LILO.

The bootloader install location should be either: (1) the MBR, or (2) the new install’s root partition. Virtually all installers will place the bootloader into the MBR by default. Unfortunately, many make it unnecessarily complicated to override this and select the root partition. Often, the ability to change the default requires clicking on an advanced/expert button someplace on one of the screens associated with
bootloader configuration, and it can be easy to miss the button if you are not familiar with the installer. Another possible complication is that some installers will let you pick from all legal disk partitions for the installation location, so you must be careful to pick the root partition that you are installing to. Finally, instead of using standard Linux device path notation to specify the install partition, some installers will make you use GRUB partition notation:

<table>
<thead>
<tr>
<th>Device Name</th>
<th>GRUB Notation</th>
<th>GRUB Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda</td>
<td>(hd0)</td>
<td>both</td>
</tr>
<tr>
<td>/dev/sdb</td>
<td>(hd1)</td>
<td>both</td>
</tr>
<tr>
<td>/dev/sda1</td>
<td>(hd0,0)</td>
<td>GRUB Legacy</td>
</tr>
<tr>
<td>/dev/sda2</td>
<td>(hd0,1)</td>
<td>GRUB2</td>
</tr>
<tr>
<td>/dev/sd5</td>
<td>(hd0,4)</td>
<td>GRUB Legacy</td>
</tr>
<tr>
<td>/dev/sd5</td>
<td>(hd0,5)</td>
<td>GRUB2</td>
</tr>
</tbody>
</table>

Many installers will scan your disks for existing OS’s and make boot menu entries for them (often along with “safe mode” or other entries for the Linux being installed). Some installers can give you the option to edit the menu, if you do not want it so cluttered. This is also easy to change post install.

### 3.3 Software Selection

Each installer has a different approach to selecting the software to be installed with the system. True install disks generally provide some control over what will be installed, such as KDE vs. GNOME, or the ability to select classes of optional packages (e.g., development software). Some allow you to view and select among basically every available package if desired (e.g., Mageia). By constrast, the “live” installers are often offered in one “flavor” (e.g., KDE or GNOME) and simply install all software on the CD. Most of these live disks install a fairly limited selection of software, so you are unlikely to end up with GCC and other development tools. Virtually all distros include GUI tools that can be used after the install to access online repositories and get additional software that is found to be needed. You may need a fast Internet connection if you want to add something substantial like GNOME or KDE, however.

From a security standpoint, many installers leave something to be desired because they make it difficult/impossible to determine—let alone control—what server software gets installed. For example, many distros install servers such as SSH, NFS, and Sendmail without explicit selection by the user and without any indication that this is being done. If these servers are set to be started at boot and are not firewalled, the newly installed system may be immediately vulnerable to attack. See the Section 3.7 below for further information.

### 3.4 Root and User Accounts

For security, you should never be logged into the GUI/desktop as root. Thus, it is important to create at least one non-root user account during the installation process, and virtually all installers will allow this. Most installers also have you set up root’s (the administrator’s) password as well (except for distros like Ubuntu that disallow root logins).

Be sure to pick strong passwords—particularly for root. Some installers check the strength of passwords and others do not. It is also important that you remember and/or write down the passwords that you choose, as resetting lost passwords takes some effort.

The other user-related option that you may see in installers is the choice of whether to automatically login a “primary” non-root user account when the GUI starts. While this is a convenience, it reduces the security of your system because no password will be required to reach a desktop, so anyone that has access to the
machine can access this account. Thus, this option is not a good idea in general.

### 3.5 Graphics Configuration

Only a very few distros these days will allow you to set up a system without the GUI/X11 being installed and set to start at boot. The graphics system can be one of the most problematic aspects of an installation, though this has become fairly rare (at least with desktop computers). If not properly configured, when you reboot you will end up in a console, unable to run any of the distros GUI configuration tools. Most Linux installers do a pretty good job of probing graphics cards and monitors, and often the graphics and display will be set up perfectly without any input from you. However, since things sometimes do not work perfectly, it is useful to have some understanding about the graphics hardware on your system as well as X11.

One of the most common issues one may face is that the default resolution that is chosen may not be optimal, so that should definitely be known ahead of time. The driver will affect the screen resolutions that can be chosen, as well as the possible bit depth. The bit depth should be 24 or 16 (avoid 8 unless nothing else is an option, and then fix after reboot). The monitor will affect the resolutions that should be chosen and the refresh rates. CRT monitors can handle a range of resolutions and still look sharp, but will have a maximum resolution. LCD monitors should generally be run at their so-called “native resolution” or they will look very poor. Most CRT monitors have a 4:3 aspect ratio, so should be run at a resolution like 1024x768 or 1280x960. LCD monitors can vary a great deal in aspect ratio, so are harder to predict. 1280x1024 is a common resolution for 17–19” (non-widescreen) LCD monitors. Many monitors (e.g., 24”) now have 1920x1080 resolutions since that is the size of HDTV frames. CRT’s require a refresh rate of at least 70 Hz or they will flicker, but if too high a rate is used they will fail to display anything. LCD monitors all now use refresh rates of 50–60 Hz.

If the graphics card is not recognized or there isn’t a driver for it, most installers will setup the generic VESA driver. This driver is supposed to work for all cards, though resolutions will be limited and there will be no hardware acceleration. If you see this as the selected/default option, you can either (1) leave it until after the install is complete, but then try to change it, or (2) change it during the install if you know the card (and there is a driver for it).

Both Nvidia and ATI/AMD produce their own “proprietary” Linux graphics drivers. Because these drivers are closed-source, distributed only in binary form, and with restrictive licensing requirements, they are not generally included with free versions of distros. Instead, the installer will setup an X.org open source driver will be installed (e.g., “nouveau” for Nvidia cards or “radeon” for ATI/AMD cards). These drivers generally do not provide the same level of graphics performance since they do not typically support hardware acceleration. For many distros, there are either official or unofficial/semi-official online repositories from which you can download and install the proprietary drivers post install using the distros package system. If a distro does not make the proprietary drivers available from repositories, you will have to download and install the proprietary drivers manually post install. You can find and download the Nvidia and ATI/AMD Linux drivers from these sites:

- [www.nvidia.com/object/unix.html](http://www.nvidia.com/object/unix.html)

Read and follow the instructions for installation on the sites. Note that Nvidia has four different video drivers for different age ranges of their cards! Both companies provide executables that must be run as root, in a **console**, without the GUI/X11 running. (The GUI/X11 can be stopped with a command like “init 3” in Linuxes using `runlevels`, or “/etc/init.d/gdm stop” in Linuxes without runlevels.) The Nvidia installer will compile and install custom drivers for your kernel. Once the drivers have been created, the new driver will need to be specified in the X11 configuration file, usually `/etc/X11/xorg.conf`. This can be done manually or using an Nvidia utility. What must be changed is the “Driver” option under the “Device”
section, (e.g., from “nouveau”, to “nvidia”). After that, restart the GUI/X11 (e.g., switch to runlevel 5) or reboot. The ATI installer is a similar, executable download.

It is important to be aware that because installing these drivers can require compilation, “development software” (like the GCC compiler) will have to be installed, and also kernel header and/or source packages. A few distros install development software by default, but most do not (particularly with the “live” installers). Virtually none install kernel headers/source by default. Thus, if you are planning on installing a commercial graphics driver, you will want to consider how to get the required Linux software installed.

3.6 Network Configuration

Most wired Ethernet cards, NICs, will work perfectly with Linux, and will be automatically setup to use DHCP. Thus, if you will use a wired network connection with DHCP, there is generally very little you need to know or do for the installation. If your machine is to use a static IP address, then you will need to know the following pieces of information prior to setting up networking: (1) IP address, (2) hostname, (3) gateway IP address, (4) netmask, and (5) IP addresses of DNS servers (should have 2-3). You will need to select the static setup options and enter this information. Not all installers allow you to setup static addressing, so you may have to do this post install.

With laptops, wireless/WiFi networking is often what is desired. Unfortunately, as discussed in Section 2.6, wireless cards often may not be properly set up by Linux installers due to licensing restrictions for firmware or even because of a lack of drivers. As noted, if your wireless card will require a driver/firmware that is not included with the installer—and you will not have access to a wired Internet connection—it is critical that you obtain the required file(s) ahead of time (or have a second OS on the machine that can be used to obtain the files). The best chance of getting wireless network running during an install is to use one of the distros “live” installers (if available).

3.7 Servers and Services

One of the most important issues in securing a Linux system against network attack (and even local attack with a multiuser system) is limiting the servers and services that are run. As noted above in Section 3.3, many distros automatically install servers that are unnecessary for many or even most users. The most common examples are: SSH, NFS, Sendmail, and Samba. If these servers are setup to start and boot, but not firewalled, your brand new installation is potentially vulnerable to attack. This is why it may be advisable not to be connected to the Internet until after you have performed the post install checking described in Section 4.

There are a few installers that will install server software only if you ask for it, and there are a very few that will even give you a chance to see just what servers and services are installed and setup to be started at boot. Probably the best example is the Mandriva installer, which allows you to view and disable all servers and services as part of the install. One danger is that it can be hard to determine if some of the services are really required, and disabling them at install might cause the system to not work properly when booted up. Often, it is a good idea to leave them enabled at install, then disable and test only once you know the machine is functioning properly post install.
3.8 Firewall and Security Settings

Many installers now have some security-related options that you can configure. One common option is whether or not to set up a firewall at installation time and what services to allow. Generally it is a good idea to set up the firewall at install time. The only reason not to is there is a slight possibility that it could limit network access after the install, making it difficult to use the machine or even test network connectivity properly. Of course, any configuration options can always be addressed post install as well. If an installer offers to set up a firewall, it will typically give you some options for determining what servers should be accessible via the network connection. Initially, you may want to limit this to nothing or maybe just SSH until other servers are known to be properly configured. You certainly want to allow access to only those servers that you really intend to use!

Some installers also have options for setting up the general security level of the system (e.g., Mandriva). This affects a number of configuration settings, such as permissions. Be aware that if this is set too high, some programs or servers may not function properly. Often it is good to set this to something like a standard/medium value, test that programs/servers work, and then experiment with higher (more secure) settings to see what if anything breaks.

Some distros (e.g., Fedora) also offer the option of using the security subsystems like SELinux (security enhanced Linux). There is little reason to use such systems on desktop machines and still rather limited support for configuring them, so probably best to avoid.
4 Post-Installation Verification and Configuration

If your new Linux boots properly following installation, the main purpose of the post-installation process is to make certain the system is secure. Obviously, if your new Linux does not boot properly, you will first have to try to diagnose what is wrong. The most common issues are going to be a problem with the graphics configuration, an error in the bootloader installation, or networking not functioning. Ideally you should reboot with the network cable still unplugged and first make certain things are secure before plugging it in. If your network interface was set up by the installer to use DHCP but you need static settings, this can cause the first boot to pause until the DHCP requests time out.

4.1 Graphics Troubleshooting

Serious graphics problems will typically cause you to end up in a console, a non-GUI shell window that will generally appear as a black window with text. Note that there are no scroll bars in consoles, and your mouse won’t function. “Redhat-family” systems will generally get dropped to ruled level 3 (non-GUI). If the GUI/X11 is not functioning, you obviously cannot use the distro’s GUI configuration tools. A few distros have command-line versions as well, perhaps using CURSES graphics.

It is difficult to give a complete guide to troubleshooting graphics issues since many problems could occur. On Linuxes using the Xorg X server, the configuration file will be: /etc/X11/xorg.conf. This is one place to start, by looking to see if an appropriate driver was selected and appropriate resolution and refresh rates were configured. You should also examine the Xorg log file, typically /var/log/Xorg.0.log to look for indications of errors when the server attempted to start. Some distros may also have graphics-related configuration information in their own special files, for example under the /etc/sysconfig directory.

To try to fix graphics problems manually, you will generally have to edit /etc/X11/xorg.conf and change driver, resolution, modules, etc. Be certain to make a backup copy of the original file before you do any editing. Once you have made changes, you can try restarting the X server (change runlevel using “init 5” or use the command startx). Note that many recent distros no longer create an explicit xorg.conf file because the X server can now generally automatically configure itself as it starts. You can generate an explicit xorg.conf file using the command “xorg –configure” or by using xorgcfg or xorgconfig (though these tools don’t always work as desired). If worst comes to worst, you can always find example xorg.conf files on the Internet, and simply create the file with a text editor.

4.2 Verify Correct Partitions are Mounted

You should do a brief check that those partitions/filesystems that you want mounted—and only those—are what is actually being mounted. Use the df command to list the mounted filesystems. You may find, for example, that the Windows partitions/filesystems are being automatically mounted, but that you want them mounted only as needed (i.e., you will mount them only when required). Fixing this requires modifying the /etc/fstab file. Again, most distributions generally have a GUI configuration tool that can help with this. However, you can always simply manually edit the /etc/fstab file (be sure to back it up as something like /etc/fstab.orig first!). Just open up your favorite editor, find the Windows partitions lines, and either delete them or better yet, make to they are available for manual mounting but are not automatically mounted at boot time. For example, you may find an original line like this in the fstab file:

/dev/hda2 /mnt/windows vfat umask=0,iocharset=iso8859-1,codepage=850 0 0

If you want to do not want this partition automatically mounted, add “noauto” right before the “umask=0” entry, and if you also want to allow non-root users to mount this partition also add “users”. More information can be found in the man pages for the mount command. If you wish to unmount some partitions that are
mounted but that should not have been, you can use the `umount /mnt/windows` command.

### 4.3 Secure Servers and Services

#### 4.3.1 Verify Firewall

If the distro’s installer did not allow you to configure the firewall setup, you need to immediately find the distro’s firewall configuration tool and use it to enable the firewall and allow access to only those servers that you want there to be access to. Making certain that a firewall is set up properly is a fair amount of work, so you will probably want to assume that the installer did it right if it did it at all. You should, however, check that it did something. You can see whether firewall rules have been loaded into the Linux kernel with this command: `iptables -L INPUT`. If you see no rules listed (just the headings), then immediately figure out what is wrong with the firewall before you plug your machine into the network. Ultimately, you should run a port-scanning program against your machine to verify that the combination of firewall and listening servers is what you desire. However, this cannot be done from the machine itself, it must be done from another machine (because the firewall will generally handle traffic from the local machine to itself differently from the traffic coming to the machine from the network interface).

#### 4.3.2 Verify that Only Desired Servers are Running

Verifying what servers are running can be somewhat difficult to do. Among the possible methods are:

- Use a GUI configuration tool provided by the distribution.
- Examine the running processes.
- Look at what programs are “listening” for connections.

If the distro has a GUI tool to examine servers and services this is often the easiest way to see what is running. One problem is that these tools do not usually distinguish between services that involve servers and those that do not. Many Linux “services” are simply scripts run at boot time that do things like start the network interfaces or run periodic jobs that were missed. Since they do not start servers that listen for connections, they are generally not a security threat.

If you use a command like `ps aux` to see what processes are running, you can look for server programs such as sshd, apache, portmap, etc. Of course, you may not know the name of every server that might be running, so it is easy to miss one.

The standard command for checking for listening servers is `netstat`. The command `netstat -lpt` will list the listening TCP sockets along with their associated programs/processes, while `netstat -lnpu` will do the same for listening UDP sockets. These two commands should show you all of the servers that are potentially accessible from the network. If you see programs that you do not want running, you need to stop them and make certain they will not restart on the next reboot. There are several ways to do this, which we will discuss in a handout on startup configuration.

#### 4.3.3 Limit Access to Servers

Even when you want to have certain servers running, it is not always the case that you want anyone on the Internet to be able to reach them. Some servers have their own configuration files that control access, so you will have to read their documentation. However, there is a generic “hosts access” system that is used
by many servers. Setting this up requires configuring two files, which were originally part of what is often referred to as *topw*  

topw*  
were used by servers started via inetd or xinetd, but are now also used by many servers to limit access to the server by IP address or domain. The two files are `/etc/hosts.allow` and `/etc/hosts/deny.

By default, most distributions provide empty hosts.* files. You will nearly always want to add the following line to `/etc/hosts.deny`:

```
ALL: ALL
```

This sets up a default policy of denying access to all servers, from all IP addresses. Access must then be granted to each server for appropriate addresses. For example, adding following line to `/etc/hosts.allow` will grant access to the SSH server from all siu.edu machines:

```
sshd: .siu.edu
```

or even better, use IP addresses:

```
sshd: 131.230
```

Use the command "man hosts.allow" to get detailed information.

You may also need to look at the configuration files for *xinetd*. The xinetd configuration info is found in the file `/etc/xinetd.conf` and the directory `/etc/xinetd.d`.

The firewall offers another mechanism for limiting access to servers. However, GUI configuration tools cannot deal with this level of sophistication, so you will probably have to read up on *Netfilter* and *iptables*.

### 4.4 Verify Network Connection(s)

After you have made certain that any servers are secure (as explained above), you should plug in the network cable or initiate wireless networking and either check that networking is functioning or configure and test it. Since the ability to configure networking varies greatly among Linux installers, different Linuxes can be in very different states post install. Most Linux installers these days will automatically configure wired network interfaces (NICs) and set them up to use DHCP to automatically obtain network parameters. After plugging in the network cable most Linuxes now will initiate DHCP, but if yours does not, you will have to restart the network services (or reboot if you don’t know how to do this). On “RedHat-family” systems you can restart the network interface using the command: "service network restart". If you configured a static IP setup during the installation, then networking should just begin working once you plug in the network cable.

If you do not want to use DHCP on your wired connection and you did not configure a static connection during the install, you will need to do so at this point. Most distros will provide a GUI network configuration tool, but most will also set up KDE and GNOME with network applets that can be used to configure network devices. These applets will also tell you if your network interface is working. Networking can also be configured manually, but it considerably more complicated, and varies from distro to distro.

Unless you explicitly configured a wireless interface, it will not be fully configured during the installation process. Some installers will select an appropriate driver and begin the configuration to use DHCP, but parameters like SSID, encryption type, and encryption key have to be supplied manually. Use the distro’s GUI network configuration tool or network applet.

Once you believe the network interface is properly configured, you need to verify your ability connect to the Internet. While the network applets may suffice, you should also open up a web browser and surf some sites. You can also use the ping command, though you will need to know the hostname/IP address of a machine that will definitely respond to pings, as many machines do not these days. You should generally be able to ping the gateway machine and DNS servers. If you know both the hostname and the IP address of the
gateway machine, this should be an excellent way to determine both connectivity and whether problems are due to DNS or not.

To ping the gateway machine, issue a command like:

```
ping -c5 192.168.1.1
```

You should see something like this:

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=1 ttl=255 time=0.461 ms
64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=2 ttl=255 time=0.565 ms
64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=3 ttl=255 time=0.442 ms
64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=4 ttl=255 time=0.443 ms
64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=5 ttl=255 time=0.446 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 0.442/0.471/0.565/0.051 ms
```

If you there is a network connectivity problem, you will see something like:

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
From 192.168.1.30 icmp_seq=2 Destination Host Unreachable
From 192.168.1.30 icmp_seq=3 Destination Host Unreachable
From 192.168.1.30 icmp_seq=4 Destination Host Unreachable
From 192.168.1.30 icmp_seq=5 Destination Host Unreachable

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3999ms
```

If the network connection is OK but the machine does not respond to pings, you should see something like:

```
PING 131.230.133.82 (131.230.133.82) 56(84) bytes of data.
... ...
--- 131.230.133.82 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 3998ms
```

### 4.4.1 Networking Troubleshooting

If you find that you lack network connectivity, you have have to troubleshoot the problem. There are various configuration settings that need to be checked, each of which will be addressed below.

**IP Address:**

You need to see if an *IP address* has been assigned to your network interface (either via successful DHCP or via a static setting). Run the `ifconfig` command (or use the newer and more general `ip`). You should see an entry for the *loopback device* (lo) and at least one more interface device for your ethernet card/NIC or wireless/WiFi card. Wired NICs have always had device names of the form `ethn`, e.g., `eth0`. However, when there are multiple NICs on a machine, consistent naming has not been guaranteed between boots, so a new naming scheme is beginning to be used. This sheme, called *Consistent Network Device Naming*.
assigns names starting with em1 to onboard NICs, and names like p5p1 to PCI NICs (where 5 represents the
PCI slot and 1 is the Ethernet “port”). Wireless/WiFi network cards have device names of the form wlan
, e.g., wlan0.

Make certain the second line for your active interface device is something like:

```
inet addr:192.168.1.31 Bcast:192.168.1.255 Mask:255.255.255.0
```

This indicates that an IP address has been assigned to the card. You should also check that the IP address
and netmask are correct. These should either be what you set or what was sent by the DHCP server. If the
DHCP process failed, you will not see a valid IP address with the inet addr entry.

Routing:
Check that the gateway machine has been properly configured. Use the command “route -n” to check the
packet routing table. There must be at least two entries, looking something like:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Genmask</th>
<th>Flags</th>
<th>Metric</th>
<th>Ref</th>
<th>Use</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td>192.168.1.1</td>
<td>0.0.0.0</td>
<td>UG</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
</tbody>
</table>

This indicates that packets with local IP addresses (192.168.1.n) will be sent directly to the indicated machine,
while all others will be sent to the default gateway machine (here 192.168.1.1).

The routing table may contain other entries as well, such as:

```
127.0.0.0     0.0.0.0     255.0.0.0   U    0    0    0    lo
```

and/or

```
169.254.0.0   0.0.0.0     255.255.0.0 U    0    0    0    eth0
```

DNS:
Check that DNS servers have been properly listed (either via successful DHCP or static settings). This
information is stored in the file /etc/resolv.conf and would either have been put there by the installer
(static IP) or by the DHCP client. Make certain that there is at least one line like:

```
nameserver 131.230.133.5
```

There can be up to three nameserver entries in this file, and they will be tried in order until a response
is found. If you have a static IP, you can change/add DNS servers by simply editing this file (and then
restarting any applications you want to use the new servers), but be aware that this file is rewritten by the
DHCP client every time the IP address lease is renewed, so you cannot practically edit this file if using DHCP.

hosts File:
The file etc/hosts is used to map hostnames to IP addresses without having to contact a DNS server. This
file should generally have one or more mappings for the local machine, like:

```
127.0.0.1  localhost
```

and if you have a static IP address, like:

```
131.230.133.110  cspc5.cs.siu.edu cspc5
```

If you want to definitely be able to contact machines using hostnames even if your DNS service is down, you
can add entries to this file using an editor (make a backup copy of the original file first!).

Distro-Specific Configuration Files:
Many distros have specific network configuration files that are used to load network parameters such as the
gateway machine and netmask. If you are unable to use a distro’s network tool, you will have to find and edit
these files. In “Redhat-family” systems, the configuration files will be found under /etc/sysconfig. There
may be a file network with certain parameters, and often a network-scripts with files like ifcfg-eth0
that contain detailed parameters for an interface. In “Debian-family” systems, there will generally be a
directory `/etc/network` containing files like `interfaces` and `options` that define network parameters.

**Wireless Tools:**
There are several special command-line utilities for wireless cards:

- **iwconfig** — manipulate basic wireless interface parameters;
- **iwlist** — perform scanning and list frequencies, bit-rates, encryption keys...;
- **iwspy** — get per node link quality.