
RMIT School of Computer Science

COSC1131/1133 Unix Systems Administration

Lab Session 1 - Basic Linux Installation

Contents:

1. [Partitioning the hard disk](#)
2. [Installing Slackware 8.0](#)
3. [Exploring your new Linux system](#)
4. [APPENDIX: The PC Hard Disk Partitioning System](#)

Introduction

In this lab, you will be installing Linux on a computer, replacing anything that may already be installed on the machine. You will be installing the Slackware 8.0 Linux distribution. You will need a Slackware 8.0 installation CD.

Turn on the computer and monitor, and place the CD in the CD-ROM drive. The machine should boot from the CD, and give a boot prompt. You do not need to enter anything at this prompt; press ENTER. When the system prompts you for a keyboard map, press ENTER also, to use the default map. Finally, the system will present you with a login prompt. Log in as `root` (the superuser); there is no password.

You will be presented with a UNIX shell prompt. From this prompt, you can run commands needed to install Linux on the PC.

Partitioning the Hard Disk

Firstly, you will need to partition the hard disk, creating the disk partitions where Linux will be installed. You can do this with the `fdisk` command. Enter "`fdisk /dev/hda`" at the prompt and press ENTER. (On a Linux system `hda`, `hdb`, etc refer to IDE hard disks. If the system had SCSI disks they would be `sda`, `sdb`, ...)

`fdisk` presents you with its own prompt, at which you can enter `fdisk` commands. To see the list of available commands, enter the `'m'` command.

Firstly, you will need to delete the old partitions, if any, on the computer. Use the `'p'` command to list the partitions that exist. Then use the `'d'` command to delete them, one by one. It is best to start deleting at the largest partition number and work backwards to 1)Y

Now you will need to create the following partitions:

1. a root (/) partition, of 200Mb. This will be a primary partition. (hda1)
2. An extended partition encompassing the rest of the disk, and containing the following logical partitions: (hda2)
 1. a /usr partition of 2Gb (2048Mb). (hda5)
 2. a /home partition of 1Gb (1024Mb). (hda6)
 3. a /usr/local partition of 2Gb (hda7)
 4. a /var partition of 100Mb (hda8)
 5. a /tmp partition of 100Mb (hda9)
 6. a swap partition of 128Mb (hda10)

To create the root partition, enter the 'n' command, and select 'p'. Give it the partition number 1. Press ENTER when prompted for the first cylinder to go with the default value. For a size, enter '+200M' for 200 megabytes. Then enter the 'p' command and look at your handiwork.

Then create the extended partition (command 'n', action 'e'; default first cylinder and last cylinder; this will give it the rest of the disk

The other partitions will be logical partitions. Create them and note down which device names/numbers correspond to which partitions.

When creating the swap partition, you will need to change its type to 82, with the 't' command, to tell the system that it is a swap partition. Enter 't', the number of your intended swap partition, and 'L' to list codes. Then enter 82, which you will notice means "Linux swap". Note that this swap partition is too small for a "real" production system. We keep it small in the lab because creating the swap system takes a while later in the install process. For a real system on these machines a swap of between 256Mb and 512Mb would be about right. (But it would take nearly 10 minutes at the later part of the install)

When you're finished, enter the 'p' command and look at the partition list. What do you notice about the device names of the partitions? Then enter the 'w' command to write the partition table and exit fdisk.

Installing Slackware

In the next step, start the installation program with the `setup` command. This will present a menu of actions that can be performed to install Linux on a computer. Use the cursor keys to navigate the menu. Firstly, you'll want to set up your newly created swap partition; select `ADDSWAP` from the menu and press ENTER. If you created the partition correctly, it will be displayed on screen, asking whether you wish to use it. Say yes. (If it doesn't appear, go back to fdisk and make sure that you have a swap partition, and it is of type 82.)

The installer then goes on to configuring your other partitions. Firstly, you will need to choose a root (/) partition. Select the first (primary) partition you created for this. From the format options menu, select "quick format", with no bad block checking. (If you're installing on a hard disk for the first time, checking for bad blocks can be a good idea; however, it takes a while, and so we will omit it in order to fit the lab into 2 hours.) From the inode density menu, select the default option; for most cases, you will not need to change this.

Now go on to do the same for the remaining partitions, formatting them and assigning them to their mount points.

Next you will install the packages which comprise the Linux system. Go on to the "SOURCE selection" screen, and select "install from Slackware CD-ROM". You will then be presented with a menu of application categories, from which you can select packages. If you don't want to install anything from a category, you can unselect that category at this stage. Make sure that the Emacs,

GTK, KDE, TeX, Xfree86, XView, XApps and kernel sources categories are not selected.

After you hit enter an additional menu is displayed offering a variety of installation styles. Select the "menu" installation mode. This will present you with a menu of all the individual packages in each category, allowing you to choose which to install.

Then proceed to select and install the packages, following these guidelines:

- From series A, select the "Linux 2.4.5 with no SCSI" option. (This is the more recent kernel, and the lab machines do not have SCSI devices.) Make sure that you accept all of the rest of the defaults.
- From series AP, make sure that `man`, `manpages` and `texinfo` are selected; these tools are useful for looking up documentation on commands.
- From series N (network-related), install only the following:
`autofs`, `tcpip1`, `dhcp`, `iptables`, `lynx`, `openssh`, `openssl`.
(Installing unnecessary network services can sometimes introduce security problems in a production environment.)
- In general, the safest option is to accept the default packages selected in each group that you install. The reason for this is that Slackware does not check for package dependencies.

Once you have installed packages, the installer will ask whether you wish to make a boot floppy; say no. Then proceed to the final stage of the setup. The computer does not have a modem, and the mouse is a PS/2 bus mouse. The hardware clock is set to UTC (Coordinated Universal Time); this is useful, as it allows daylight saving to be handled automatically. The time zone is "Australia/Melbourne". Skip the networking configuration stage; assume that the machine will not be on a network.

The installer will ask how you wish to install lilo, the Linux boot loader. lilo is the system which is responsible for booting a Linux system from the hard disk. As this is a Linux-only system, install lilo to the master boot record (MBR). Do not be concerned that it says "Possibly dangerous". The reason for this is that if more than one operating system is installed, the non-Linux system may not boot correctly if lilo is on the MBR.

Finally, you will need to enter a root password. The root password controls access to the root account. When you are logged in as root, you have absolute power over the Linux system; you use the root account for administering the system. In the lab, *but only in the lab*, a good choice is "password". For the root password on a production system you would need a really good password. The key attributes of a good password are:

1. Absolutely not in a dictionary or a "made-up" word from common Science Fiction or Fantasy novels or films.
2. No names. (Above comments about fiction apply)
3. Not a variation on your name or your username
4. Should have a combination of UPPER and lower case letters, numbers, and punctuation and/or special characters(#@%|~).

How do you do this and remember the password? My suggestion is to take a line from a song, poem, play or a familiar quotation. Then use the first letters of the words, maintain the case, and punctuation. If the word for or four is in the line, then substitute 4. If to, too, or two are in the line, then replace with 2. Or can be replaced with | and not with ! (from C programming). What this leaves you with is a password that is absolutely unguessable, but which you can easily remember because the line you remember is meaningful. As an example, "To be or not to be, that is the question -" becomes "2b|2b, titq-". This is a really famous phrase, so it probably is not a really great password.

Finally, when the installation is completed, remove the CD and reboot the system. If everything

worked, it should boot into Linux, and present a login prompt. You can then log in as `root`, with the password you set.

Exploring your new Linux system

Log in as `root` and change to the root directory (`cd /`). Get a listing of the files there (`ls`). Look in some of the directories.

Look in the `/proc` directory, and examine the contents of some of the 'files'. What do the files there tell you about the computer and the Linux system it is running?

Now enter `/etc` and look at some of the files. `/etc` contains configuration files which control how parts of the Linux system operate. You can use the `less` command to look at files, or `vi` to edit them. Have a look at the file `/etc/fstab`. What does it do?

Examine the contents of `/etc/passwd`. What do the lines there correspond to? Also look at the file `/etc/shadow`. Can you see the relationship?

Edit the file `/etc/motd` and enter a welcome message, which will be displayed. Write the file and exit the editor. **NOTE:** the changes made to the `motd` will not be there after a reboot. We will discuss why it works that way.

Next you should create a user account for yourself. Because the `root` user is so powerful, using it too often can be dangerous; there is nothing stopping you from destroying the system or making it unusable with a careless command. Which is why you want a regular user account without those powers. You will use this account for everything that does not require root privilege.

To create a new user, use the `adduser` command. Enter a user name (up to 8 characters, no spaces) when prompted, and then enter other details. For many of the fields, you can just press ENTER to accept the default. Finally, give your new account a password. (NOTE that `adduser` is a script, not a binary file.)

Once the account is created, press ALT and F2 on the keyboard. This will take you to a new virtual terminal. Log in as your user account to make sure that it works. When you log in, your welcome message should be displayed.

Press Alt and F1. You will be returned to the first terminal, where you are `root`. What do you notice about the two shell prompts? How can you tell if you are logged in as `root`?

(Linux has up to 10 virtual terminals accessible from the console; you can switch between them with ALT and the function keys. This allows you to have multiple logins and shells on one console.)

Now you will create an account for a new user. For this you will use the `useradd` command. (Unlike `adduser`, `useradd` is non-interactive, taking options on the command line. This is often useful in scripting.) To get the manual page for `useradd`, type `man useradd`.

Call the user 'testuser'. After you do this, you will need to give them a password. Look at the manual page for the `passwd` command. Once you have done that, go to another console (Alt+F3) and log in as `testuser`. Where is `testuser`'s home directory located?

Then try the 'w' command. What does it tell you?

Now, go back to your root shell, and enter the command 'mount'. What does this show you? (Hint: if you don't understand how a command works, you can look at the manual page for it with `man`.) Which filesystems are mounted on your Linux system? What do they do?

Finally, when you are finished, go to a root shell and shut down the system with the 'halt' command.

APPENDIX: The PC hard disk partitioning scheme

The scheme used for dividing a PC hard disk into partitions originated in the 1980s with the IBM PC. Originally, this scheme made provisions for dividing a hard disk into 4 partitions, known as *primary partitions*. However, this became too limiting, so the *extended partition* was invented. An extended partition is a primary partition, which can contain other partitions. These are known as *logical partitions*. You can only boot off a primary partition, which is why your Linux root partition (/) should be a primary partition.

Under Linux, partitions 1 through to 4 (hda1...hda4 on the first hard disk) are primary partitions, one of which may be an extended partition. Partitions from 5 upwards are logical partitions.

