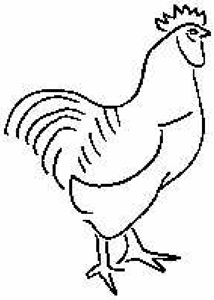
CLTC Documentation Sheet 1/B:

Basic Installation of the CLTC's Server System



Developed by The 'Free Range' Community-Linux Training Centre Project -
Version 1.0, January 2003.   http://www.fraw.org.uk/cltc/

"*You have 0.000012 milliseconds in which to perform this basic operation. If you fail we will have no choice but to transplant your vital organs into more adaptable specimens and to recycle your residue through thermite crucibles.*"
Edward Abbey, *The Monkey Wrench Gang*

This sheet looks at the basic installation and configuration of the CLTC's server. The configuration of services and the management of the server will be dealt with later in individual sheets. This sheet only considers the installation of the Linux distribution onto a laptop, and post-installation tasks to facilitate the installation of network services.

The objectives of the server

Installing the CLTC's server is not meant to be an exercise in developing the most professional and secure network server possible. Quite the opposite in fact.

The method of configuration for the CLTC server has to be simple so that non-experts can undertake the installation in order to learn. It also *has* to be insecure, so that by fixing insecurities they learn to make the server system itself secure. In developing this system we have not tried to be particular stylish, fastidiously correct, or technically logical. What we've sought to do a basic install as other non-professionals (like us!) might do. Then, simply and easily, with the least effort, hack around with the configuration of the system until 'it works'.

The other aspect to installing the server is documentation. This is as boring as hell for the non-professional user, but there are good reasons to document each step of the installation (as we have done to create these briefing sheets). For the maintenance of a server used by a group of people, the best reason to document is that it allows others, who may not have the same level of experience as the primary technician, to reinstall or maintain a server when the primary technician is not available. Another good reason is that by documenting the system you have a 'prescription' for installing the server. Then, as a learning exercise, you give someone the installation disks, the prescription, and tell them to get on with it.

Installing a Linux system is dependent upon the computer you are using and the distribution you select for your server. For the CLTC we chose *Red Hat 7.1 Professional. Please note that this information deals only with the configuration of the system under Red Hat 7.1 – other distributions, as well as different versions of Red Hat, have slightly different systems of configuration.*

We chose Red Hat because it allows 'low level configuration', without the use of graphical tools, so that people can learn the mechanics of server installation. Other distributions, such as SuSE, have easy to use graphical interfaces to configure the whole server system. But in our experience graphical systems limit you ability to hack around with the configuration of the server when you need it most. For example, installing a server on an old '486 or Pentium-I PC when you pretty much have to use text mode rather than hi-res graphics a your interface.

Also, for teaching purposes, it's better to illustrate the concepts behind server configuration from the command line and text files than the simple point and click interfaces of GUI programs. Therefore we use Red Hat for the synthesis of quick and easy graphical tools, with the ability to resort to command line and text-file configuration when required. Conversely, the CLTC's client systems can use a variety of distributions because, on the client systems, other distributions give a better range of multimedia applications than Red Hat.

Restrictions on installation

On the CLTC, because of the laptops we're using, there are the restrictions on what is possible when installing the server:

Firstly, it's running on a laptop. This may require a little lateral thinking if you have problems with combines CD-ROM/floppy drives that don't swap, or problems with initialisation of hardware.

Secondly, these older laptops only have 4 gigabyte hard drives. Therefore we need to install the minimum necessary to produce a workable server, with additional client-side programs to allow it to used for demonstration purposes (if required).

The installation procedure is fairly standard, following the procedure outlined in the Red Hat installation guide. However, as noted in *Documentation Sheet 1A*, using laptops creates a problem when selecting the drivers for the video card and display. As we were using older laptops the drivers for the video card were available, and by selecting an appropriate type of display from the list of monitors installation was not complex. Even so, it was helpful to consult some of the many 'laptop installation' web sites to get other people's tips on installing Linux on a particular model.

The restrictions of the hard disk meant that we can't spare any disk space for unused programs – and so we had to minimise the number of packages installed.

We also have to consider what this server will be used for. We may want to reinstall from scratch to teach server installation. We may even want to let people attempt to 'crack' the server over the local network in order to teach system security. Therefore we don't need to consider security as part of the method by which the hard disk is partitioned and installed. We therefore installed everything on a single partition.

All PC-like hard disks use a standard partitioning system. A hard disk can have up to four 'primary' partitions. But by allocating one of the primary partitions as an 'extended partition' that contains the bulk of the drive space, we can allocate many more smaller 'drives'. Usually multiple partitions are used to increase security. For example by allocating a /home partition you can undertake system maintenance, or even a complete reinstall, without the system users' personal data being affected. You can also partition 'active' directories within the system (like /home or /var) from the other fairly inactive directories in order to improve system monitoring and intrusion detection.

The CLTC's server need not be configured for reliability or security – so we're doing none of that here. In the process we save a lot of disk space that would normally be committed to providing a 'head' of free space on each partition/drive.

Linux installations require a fixed area of the disk to be a 'swap' partition. As a general (if conservative) rule, you allocate twice the memory capacity on the disk as a swap file. Therefore we allocate 256 megabytes (twice the 128 megabyte memory capacity) as a swap file, and all the rest as a single 'root' partition (in actuality, we allocated a round 3,850MB as root, and then the remaining 274MB as swap, just because it was easier to use a round number for the amount of root storage).

The installation process

The first part of the installation process involves changing the BIOS settings of the computer. We have to make the machine's CD-ROM the primary boot device. This enables us to boot from the installation CD directly.

You can boot from a floppy to install Linux. However there is little point when you would have to change the same setting to do so, and most Linux distributions provide a bootable CD that removes the need to use a floppy drive.

On the Dell CPi laptops, the BIOS supports different levels of password. The 'configuration' password locks up the system preventing anyone from changing the settings, changing the BIOS passwords, or booting from a CD or floppy drive. The configuration password needs to be disabled before the boot configuration can be changed.

After booting from the CD you are presented with a list of options. Pressing 'return' will lead you to the standard graphical installation program. If you know that your system has problems with the graphical installation you have to type in another option, such as 'low res'.

The next stages are fairly easy – selecting the installation language, keyboard and mouse type, and the class of installation you require ('laptop', 'server', etc.). Things become complicated when you arrive at the partitioning screen. `fdisk` is a program that allows you do directly manipulate the partition table of the hard disk. This is useful for fixing damaged partition tables to recover disk space. But normally you just proceed using 'Disk Druid'.

To begin, unless you just want to wipe the previous installation, you delete all the partitions by selecting a particular partition and clicking the 'Delete' button. If you just want to reinstall, select a partition and press 'Edit'. If you wipe, you need to press 'Add'.

After clicking Edit or Add, you get the same dialog box. This allows you to set the 'mount point' (the directory name) for the partition, and the file system type. For the `/root` partition you select the type of file system you wish to use. For the 'swap' partition just select 'swap'.

When you have finished you should have two partitions, root and swap, filling all the space on the hard drive.

Next you select your boot loader. Boot loaders control the loading of an operating system – for example, on dual boot systems allowing you to select Linux or Windows. There is only one OS in the server, for you can just select the default settings.

Next you configure the network address for the server. This is automatically configured to 'DHCP'. Uncheck this box, and then enter the numbering details that you have devised for the network. You should have planned these before embarking on installation of the server.

Firstly, enter the 'IP address'. If you now click on the 'netmask', 'network' and 'broadcast' boxes the numbers here will be completed for you automatically, as will the primary DNS box. Then enter the host name. This section is now complete.

Next you select the firewall options. Here just make sure that you check 'eth0' as a trusted device, and that you check all the protocols to be accepted by the server. We'll disable this later, but we just give a basic configuration to that we can reactivate and play with it later.

Next you select the languages that you wish to use with the system. For the server we select 'English (UK)' only. More languages take up extra disk space, and the server does not need additional languages to function on behalf of the clients (which will have the space to install all the available languages provided with the distribution).

Next you set the location of the system ('Europe/London') and the time zone (UTC). You should also select the UTC offset to 'UTC'.

The next stage is a little more laboured. You have to setup the user accounts for the server. You can do this later, but as you're here it's just as easy to do it now.

The CLTC server has one user account for each of the four main clients. This allows them to use File Transfer Protocol (FTP), Telnet and Secure-Shell (SSH) over the network. There are four more accounts that correspond to each of the virtual servers on the system. This allows each client to access the server via FTP to upload files directly to the virtual server – in a similar way to if they were running their own web site via a service provider. There's is also an 'epsilon' account (because you should work as 'root' all the time on the server), a 'guest' account that is used for networking, and a 'mobbsey' account for remote system administration (this account is common to all the laptops). Finally, there's the 'root' account for critical system administration. In total then you have to configure 11 user accounts, plus the mandatory 'root' account.

Root is easy. Just enter the password in the box provided, then enter it again below to confirm you've got it right.

Ordinary user accounts are controlled via the user account dialog. If you make a mistake you can also delete user accounts. For each user you must enter a user name, their real name, and a password. You have to repeat this process for each user you wish to create.

After completing the user account configuration you now enter package selection. This stage is critical because you need to include everything you need to operate both the laptop (laptops need extra drivers) as well as the network services. For this reason it's easier to click on *custom installation* at the beginning of the installation process, and then click on the *laptop support* option in additional and the various server packages at this stage.

You could have clicked *laptop install* earlier. The problem is then that the many packages required to run server functions would need to be installed manually.

For now can select everything except *dos/windows compatibility* and *kernel development* (if you don't use emacs as your main text editor, you can also forget the *emacs* option).

Packages are arranged into various classifications. Red Hat is fairly minimal about this. Other distributions, like SuSE, have far more categories in their package selection lists.

Each category contains packages that implement specific functions. For example, *System/Daemons* contain the packages that control most of the network services, like the Apache web server. The purpose of working through all the categories is not only to add packages. We can also deselect packages to save space. For example, do you really need the KDE *KOffice* suite on a server system? – deselecting it saves you about 15 megabytes.

Whilst selecting or deselecting packages, be sure to keep an eye on the 'install size' display. You need at least a gigabyte (1,024 megabytes) of free disk space following installation. This is to install other programs, set-up and run network services, and store data on the system for people to use. If you go below this limit, consider deselecting a lot more of the unessential packages.

When you finish package selection the installation program will check for "dependencies". Trying to install packages at a later date, after installation, can be a nightmare because of the problem of *unresolved dependencies* – programs that are supplied by other packages, and that have not been installed. These will block installation of the package you want to install. This is especially true of large programs, such as network services, which have hooks into many different parts of the operating system.

By installing everything that you need at this stage the installation program takes care of dependencies for you. It knows which packages depend upon which, and installs them all for you automatically. This simplifies the process greatly. For example, all you have to do is select the 'Apache' web server program, and the installation program will make sure that all its dependencies are installed for you.

When you have OKay'd the installation of extra packages to satisfy dependencies you move onto configure the graphics card for your computer. You can read this information from the text displayed when you first boot-up the computer. But the program will usually select the right driver program. If not, you need to go online and find a site that lists the complete specifications for the computer that you are using. You can then select the right driver manually... if the distribution has one.

Next, you are asked to OK the installation of the operating system. After clicking OK, the installation program goes about its business for a long period of time. The 266MHz CLTC server system takes around an hour and a half to install a little over two gigabytes of packages. Faster computers may take a little less. But during this period of time you have to be around to change the installation CD in the CD-ROM drive. Red Hat 7.1 has two installation CDs, with about 40 minutes between the start of installation and the change of CD. On the CLTC system, this time can be usefully employed by beginning the installation of some of the client machines.

When package installation completes, you are next asked if you wish to create a boot disk. This would be useful. But with the CLTC's laptops, the CD-ROM/floppy drive bay is not 'hot swappable' and so we can't do this. For this reason we skip this section. If you get into difficulty, and have no boot disk, you can always boot from the first installation CD in 'rescue' mode.

After this you go through 'X Configuration'. This is when you must select the monitor used with your computer, and set the correct resolution and colour depth for it. The problem with this stage is that laptops do not readily tell you what type of monitor screen the system uses.

The easiest option here is to try different monitor types until you get one that seems to fit. You do this by selecting the monitor, going to the next page and selecting the resolution and colour depth, and then hit the 'Test' button. If all is well you'll see a test screen beautifully displayed in the right dimensions and colour. If not, try again.

Note also that the page on which you select the resolution and colour depth controls the login installation. By default this is set to 'graphical'. The problem with a graphical login on the server is that the 'X Server' system that generates the pretty graphics uses a lot of processor time and memory. Even if you don't login, and just leave the server running, the fact you've started the X Server robs the server of the use of these resources. For this reason check the 'text' option. This will give you a command line (text) login that uses far less system resources.

You can also select the default desktop when you do start up the X Server. Red Hat defaults to 'Gnome'. Most other distributions default to the more function 'KDE'. Which you use is really a matter of taste and personal preference. But KDE uses more system resources, and is more bloated, than Gnome. For this reason you may prefer to use Gnome. In fact, if you installed other desktops, such as FVWM2, these are even more lightweight than Gnome but are still very functional. However, the installation program doesn't recognise them. So you select the 'text' option, and then when you login use the `switchdesk` command from the command line to change the default desktop.

For the Dell CPi's there was no information on the monitor type, so we just tried all the 'Dell' manufactures laptop displays until we found one that worked. If this doesn't work on your laptops, your last resort is to try to create a frame-buffer display. This is a special driver that tries to interface with the display at the generic level. But often the results are poor, blocky, and not very pretty.

If the monitor selection works, you reach the least screen when you can finish installation and reboot the system.

Post-installation configuration

Don't smile, you haven't finished yet. Once installation is complete you have to reset the BIOS boot setting back to the 'internal hard drive'. You can then boot up your new installation. The first thing to do is enter the root account because there are various changes to be made before we can proceed onto the actual installation of network services.

First of all we modify the configuration of the Ethernet card to accept requests from multiple IP addresses. To make life easy, there's a program called *Network Configuration* in the *Programs/System* menu.

Network Configuration controls the settings for all network devices. When you first run the program you should just see the eth0 device. To add more IP addresses we have to add an 'alias' for eth0. Pressing the 'Alias' button brings up the 'select device' dialog. Click 'next' and you get the 'select Ethernet device' dialog. Click '*next*' and you will go to the 'edit ethernet/bus interface' dialog. You will see that in the 'IP' box most of the IP number has been completed for you – all you have to do is enter the last number. Make sure that the 'activate interface at boot time' option is checked, and then press 'Done'.

You will notice that there is a new device in the list called eth0:0 – the first alias of the device eth0. You can now repeat the 'alias' process a further four times to produce the second, third, fourth and fifth aliases of eth0, each with an incremented IP address. The first four of these will be used for the virtual servers. The last as an 'experimental' IP address to practice setting up servers and firewalling. When they have all been created click on the 'Activate' button. This stores the new settings and activates the new aliases on eth0.

Next, we need to install programs from the 'applications' CD that comes with Red Hat Professional. This contains Adobe Acrobat, and the Adobe Acrobat plug-in for the web browsers (useful if you need to browse Acrobat files on the server machine for some reason)

When this is done, we not need to edit the file /etc/inittab. This file controls what happens when you press the Ctrl-Alt-Del buttons. By default this causes the system to reboot – we want it to cause a system shutdown.

Open the /etc/inittab file using a text editor, such as *gedit* or *KWrite*, and change the following lines:

```
 # Trap CTRL-ALT-DELETE
 ca::ctrlaltdel:/sbin/shutdown-t3-r now
```

to:

```
 # Trap CTRL-ALT-DELETE
 ca::ctrlaltdel:/sbin/shutdown -h now
```

Now save the *inittab* file. When you next boot up, pressing Ctrl-Alt-Del will force a system shutdown. We use this method because if you're in a hurry to shut everything down, pressing Ctrl-Alt-Del on all the laptops to force a shutdown saves time.

The next task is to reassign the home directories of the four web server accounts – *alphaweb*, *betaweb*, *gammaweb*, and *deltaweb* – to point to directories within the Apache web server's storage area. This involves two operations: setting up four directories in the Apache web server's data directory, and giving these directories the correct permissions; then changing the user accounts to point to these new directories.

Firstly, we create the four directories and assign the correct ownership to them:

```
 cd /var/www
 mkdir alphaweb
 chown -R alphaweb.users alphaweb
 mkdir betaweb
 chown -R betaweb.users betaweb
 mkdir gammaweb
 chown -R gammaweb.users gammaweb
 mkdir deltaweb
 chown -R deltaweb.users deltaweb
```

To modify the user account information we use the *User Manager* program. This is located in the *Programs/System* menu. We select each of the web server accounts in turn, changing the home directory of each to point at the corresponding hope directory that we have created.

Finally, we set full access permissions on the /var/www directory:

```
 chmod -R 777 /var/www
```

Next we need to create the directory that will be exported to the client systems as a 'share' on the networked file system. We then need to change it's permissions to allow everyone to access it. We do this with the commands:

```
   mkdir /home/export
   chmod 777 /home/export
```

The last task to to completely disable the firewall of the server system. We do this by disabling the two processes that filter data flowing in and out of the system. Using a program called ntsysv, we deselect the programs iptables and ipchains.

A word of caution! You must be extremely careful when using this program. You have the potential to turn on or off important daemons.

You may also turn on daemons that are improperly configured and which could prevent the system from booting. So make sure you only change the selection of those programs that you want. Otherwise, select 'cancel' to prevent saving any changes.

After running *ntsysv*, you are given a scrolling list of all the daemons that are installed on the system. Those that are enabled are marked with an asterisk (*). You can move up/down using the arrow keys. Pressing the space bar selects/deselects a daemon.

Move down until you find *ipchains* and then hit the space bar to deselect it. Then, move down to find *iptables*, and deselect it. You can now press the tab button to jump to the 'OK' box, and then press the space bar to save and exit the changes. Pressing tab again will successively move you to the 'Cancel' option, and then back to the daemon list.

You've now completed the basic installation of the server. We now proceed on to set up each of the services we wish the server to provide.

Free Documentation License:

Note: This report has been produced entirely using open source/free software using the Linux OS.