Operating Systems Lab Sheet 9

Jack Carroll

T00194823

**(Exercise 1) Exercise 1: Checking to see where you currently are in the file system using *pwd***

You are using what is called the **Bash shell** for your work today. We will talk more about shells later – for now it is enough for you to know that a shell is capable of interpreting the command you supply at its prompt. The Bash shell is regarded as being one of the best for interactive type work. Remember that you can normally tell immediately what directory you are currently in when using the Bash shell via the **terminal window prompt**. But it doesn’t tell you what the **absolute path** is to this directory. Worse still, for some other shells, no directory information whatsoever may be transparently available by default and in order to find out the absolute path to the current working directory you’ll need to type:

pwd

which **p**rints (displays) the current **w**orking **d**irectory to the screen.

What is the current (default) working directory?

/home2/t00194823

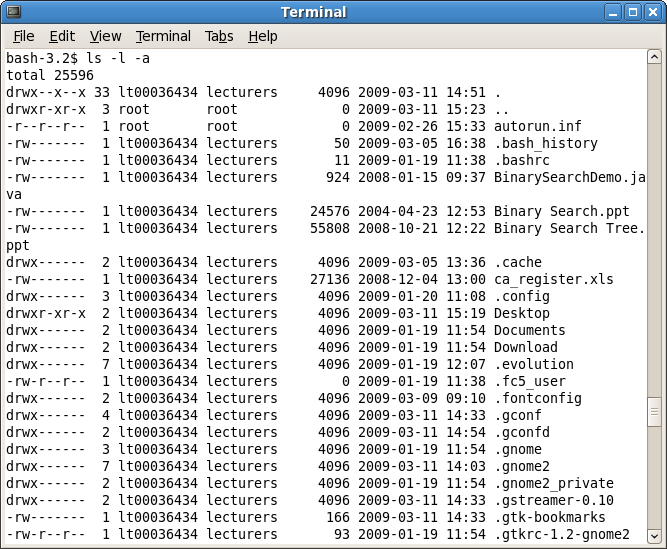
### (Exercise 2) Exercise 3: Display lots of information on files and folders (directories) using *ls -a -l*

Like most Linux commands, ls has a number of **options** (switches) associated with it, which can be used for more specific listings of files/directories.

e.g. if you type

ls –a -l

at the prompt, you will get a **long** (-l option) listing of **all** (-a option) files which means that what are called **hidden files** (often **system files** created automatically at the time your account is created) will also be displayed. You can always recognise hidden files in Linux since they begin with a **full stop** e.g. **.bash\_history**



Referring to the line in the screenshot above:

**drwxr-xr-x 2 lt000036434 lecturers 4096 2009-03-11 13:05 Desktop**

the **long-listing** means that you get a lot of extra information, from left to right these are:

**file type**, indicated by the first **d** above

**permission modes** on the files/directories, indicted by the **rwxr-xr-x** above

**number of links to the file**, indicated by the **2** above

**file** **owner**, indicated by the **lt00036434**

**group owner**, **lecturers** above

**size in bytes**, indicated by the **4096**

**time of last modification**, indicated by the date and time given **2009** etc

**file name**, indicated by the **Desktop** (a directory)

***file type symbols***

Each file has a **symbol** to indicate its **type**. These are:

**-** for regular files

**d** for directories

**l** for links

**c** for special files

**s** for sockets

**p** for pipes

So in the screenshot above you can see that my home directory, and yours also most likely, only contains **regular files** and **directories**.

***permission modes***

Any file you create yourself, you are guaranteed to be the owner of and the group you belong to depends on how the system administrator has things set up (you could be part of the CPMM100, CPGD100 or CPSW100 group for example). Permissions are a **security feature** for files and such permission modes also exist under Windows XP. Permission modes in Linux consist of 3 bits of information for each of 3 groups (owner, group and others) in that order, making **9 bits of information in total**. Each 3 bits can have rwx

r allow read access

w allow write access

x allow execution of file

– signifying that that access is not allowed.

In my case above you can read the permission modes on the file called **.bash\_history** as follows:

**rw-------**

“This file can be **r**ead or **w**ritten (modified/moved/deleted) by the **owner** of the file (lt00036434), it cannot be executed by the owner (-). The file cannot be read/written/executed by users in the same **group** as the owner or by **others**.”

With reference to the file **.gtkrc-1.2-gnome2** in the screenshot earlier, answer the following questions:

1. What type of file is this? A hidden file
2. How many links does this file have? 1 link
3. What is the size of this file in bytes? 93 bytes
4. Who owns this file? lt00000036434
5. Who is the group owner of this file? lecturers
6. What permissions does the owner of the file have? Read and write
7. What permissions do all others have on the file? Permission to read
8. What permissions does the group have on the file? Permission to read

Naturally it is **possible to alter permission modes** on a particular file and we shall look at this in a later practical.

### N.B. multiple options can be combined so that ls –a –l is equivalent to ls –al. Prove this to yourself now.

**(Exercise 3) Exercise 4: Changing into a directory using *cd***

The command for changing into a directory is exactly the same in Linux as it is for DOS and is one of the more important Linux commands. In order to go into the **Desktop** directory, which should exist in your **home directory** /home2/t00012345 (where you are currently by default) you just type

cd Desktop

You should see that just before the $ prompt,there has been a change from **~** , which is **shorthand** for “the home directory”, to “**Desktop**”.

Now use the command **ls -a** to list the files and directories in this directory. Check that the entries that are listed refer to something on the desktop. Some of these entries refer to the shortcuts that you have created on your desktop for the terminal and OpenOffice Writer application. However, you should also see references to . and ..

. means the **current working directory** (cwd) – it is just **shorthand** for it. This notation is very handy when you need, for example, to copy the contents of another directory to the directory you currently find yourself in (i.e. the cwd). Instead of having to supply the possibly long-winded **absolute path** to the cwd you can just supply . to refer to it. You will see this shorthand notation for cwd used later.

So, what do you think the effect of the following command would be? Change the directory to the Desktop which will remain at the Desktop.

cd .

Try it out to see if you are right. Write down what you think is going on here. The directory is already at the Desktop folder/directory where this directory is currently working. This leads to no change where you are at the current Desktop folder.

.. means the **directory that lies immediately above the cwd** (often called the **parent** directory). Again this is just **shorthand** notation to refer to that directory

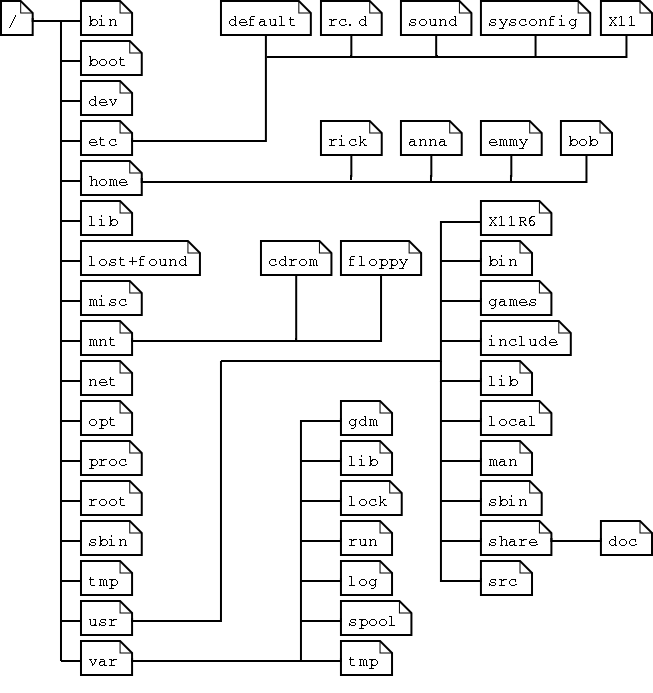
So, what do you think the effect of the following command will be? The command will go one level back to the home folder/directory of the student’s folder which is labelled ~.

cd ..

Try it out to see if you are right. Write down what you think is going on here. The command moves down one level to change from the Desktop folder to the home folder/directory of the user that is currently using the computer.

**(Exercise 4) Exercise 5: The Linux File System**

The file system in Linux appears like a **tree structure**. Depending on the exact installation, certain directories may be added or removed but your file system will definitely look similar to the graphic below. The **top** of the tree is the **root directory**, indicated by the **/** symbol.



You should now change into the root directory as follows:

cd /

You should now list all the files and directories in your root directory using **ls**.

There should be about 20 directories listed (how can you tell they are directories?). Notice that there are often no regular files or other file types listed in the root directory.

Name 3 directories that you have found in the / directory

1. boot/

2. dev/

3. misc/

The directories directly beneath the root directory contain certain types of information and break the overall file system into well-defined logical groupings. Some of these directories and what they contain are as follows:

**/bin** – common programs (binaries) shared by the system, administrator and regular users.

**/boot** – the startup files and the kernel itself, **vmlinuz**, Also GRUB boot loader data.

**/dev** – references to all the peripheral devices and disks

**/etc** – the most system configuration files, similar to “Control Panel” in Windows

**/home** – the home directories of the regular users (yours is actually in /home2 in ITT)

**/lib** – library files, including files for various programs needed by the system and users

**/mnt** – traditionally the standard mount point for external file systems such as memory sticks and external hard drives but in your case these are located under **/media**.

**/net** – the standard mount point for remote (network) file systems

**/opt** – often used to store 3rd party software

**/proc** – a virtual file system containing various information about system resources and processes

**/root** – the home directory of the administrative user root

**/sbin** – programs used by the system and administrator

**/usr** – programs, libraries and documentation used by user-related programs

**/var** – storage for temporary files created by users e.g. Log files, mail queue, temporary storage of files downloaded from the Internet, keep a temporary image of a CD before burning it.

Using the **cd** command on its own with no arguments will **change you into your home directory** from whatever directory you are currently working from. Try this now by typing

cd

The presence of the **~** before the $ prompt confirms that you are back to your home directory.

#### (Exercise 5) Exercise 6: Creating a new directory using *mkdir*

mkdir allows you to create a new directory. Assuming you are in your **home directory**, create a directory called *yourname* (so if your name is Betty Brown create the directory BettyBrown etc) as follows:

So type something like

mkdir BettyBrown

This creates a new directory in your home directory in this case.

Now check that this directory has definitely been created. What command did you use for this check? pwd

You should now change into this newly created directory and then create a new directory called OSlabstuff. So now you should have something similar to the following directory structure:

/

home2

t000\*\*\*\*\*

BettyBrown

OSlabstuff

Once you have the Oslabstuff directory created, check to ensure it is definitely there and then change into it.