Lab 1: Linux/Unix practice

Log in, using the login name and password you've been given.

You might be asked an obscure question to do with GNOME. If so, stick with GNOME.

Select Applications - System Tools – Terminal

This should give you a little window, or terminal, with a prompt and a cursor.

You are on the command line.

The prompt will be something like

bash-3.2$

bash stand for Bourne Again SHell. Stephen Bourne is a computer scientist (UK born) who, in the 70s, wrote a shell (for Unix), known as the Bourne shell. bash was based on it (and other shells such as the Korn shell and the C shell) for the GNU project. It's probably the easiest shell to use.

What shells are like

They carry out commands. (This is not true, but will do to be getting on with.)

1. You enter a command
2. If it can, the shell carries it out, at least to its own satisfaction
3. It gives you another prompt

Note that there is no positive reinforcement for you. It doesn't tell you that it's done anything, and offers no confirmation that it's done what you wanted.

So, in a shell, the habit you should adopt is

1. Enter a command
2. Check that it's worked in the way you wanted. (You need, of course, to enter another command to do this.)

For instance,

1. Copy a file
2. Check that it's been copied to the right place

If the shell can't carry out a command (because, say, you entered one in a way it disapproves of), it will send you an error message, and return you to the prompt.

Shells are case-sensitive. (So that filenames such as LIST, List, and list would be distinct and different.)
Commands are, as a rule, in lower case.

pwd

First find out where you are in the system. Enter

pwd

This stands for present working directory. The answer will give you the path of your home directory.

ls

See what's there. Enter

ls

ls lists files.

mkdir

Make two directories. Let's say they are for Java and Praxis. Enter

mkdir java

Enter

mkdir praxis

You can, of course, use any case you like for the names of directories, but many of us find it a little simpler to keep them in lower case.

Enter (it's short for list)

ls

and check that the directories have been made.

cd

This stands for change directory. It enables you to move around the file system. It's generally best to get to the directory you want to work in before doing any work. You need to tell the system, of course, which directory you wish to change to. The basic form is cd directoryName. Enter

cd java

Check where you are. Enter

pwd

Your home directory is the parent directory or your java directory. The parent is represented by two dots .. To get back to your home directory, enter

cd ..

There should be a space between the command (cd) and the dots.

Check where you are. You should be back in your home directory. Change into your praxis directory:

cd praxis

Make a directory there called lab1:

mkdir lab1

Change into your lab1 directory. Check you're in the right place.

More than one terminal

It's often useful to have more than one terminal open at a time. For instance, if we enter a command that takes control of a terminal, we often need another terminal where we can keep track of things. To show you another way of opening a terminal, select

Applications - Accessories - Terminal

You should now have two terminals on your screen. We shall use one to run an editor for us, and keep the other for Linux commands.

Editors

We now shall make a little text file. To do this, we shall need an editor. There is a range of editors available. The most straightforward is called gedit. Enter

gedit

This should give you a simple text editor. Because you were in your lab1 directory when you called gedit, gedit will save files in lab1 by default.

This is an instance of a simple rule that many users follow: before doing anything in Linux, get into the right directory first. Type (it's a quote from Samuel Beckett):

Ever tried. Ever failed. No matter. Try again. Fail again. Fail better.

Save the file. Call it something simple like sam.txt.

Get back into your other terminal. Check the file has been saved. (In lab1, type ls).

. and .. and /

The full stop refers to your current directory: the directory you're in now.

Two full stops (no space between them) refer to the parent directory.

Enter

pwd

to check where you are now.

Enter

cd ..

(There needs to be a space before the two dots.)

Enter

pwd

If you were in lab1 before, you should now be back in your praxis directory.

The backslash refers to the root of the network system. Enter

cd /

(again with a space before the /)

Enter

pwd

and at the root, referred to simply as /

To get back into your home directory, you can always enter

cd

by itself. This works wherever you are in the system. It's always easy to get back to your own workspace, your home directory.

Paths

You should be in your workspace. You can get back down to your lab1 directory in one command. Enter

cd praxis/lab1

Check where you are. Now get back to your home directory in one command.

Switches

You add a switch to a command to make it behave in a particular way. A switch is what's called an argument to the command. You always need a space between the command and the first switch. For example, enter

ls –al

The minus sign tells the shell that this is a switch (and not the name of a directory, say). a is short for all, and l is short for long listing format.

Permissions

You should see, on the left hand side of the listing, a series of letters for each file.

drwxr-xr-x 2 hh117 staff 4096 Jun 29 14:15 .
drwxr-xr-x 3 hh117 staff 4096 Jun 29 14:08 ..
-rw-r--r-- 1 hh117 staff 77 Jun 8 16:17 sam.txt

. is the current directory (hence the d as the letter before the permissions)

.. is the parent directory

sam.txt is the little text file we've made.

A file has 9 permissions, in 3 groups of 3.

The first group gives the permissions of the owner.

The second group gives the permissions of other people in the same group (in my case, staff)

The third group gives the permissions of everybody else.

The three permissions in each group are

r, which means read-access

w, which means write-access

x, which means executable

So, I can read or write (alter) my sam.txt. Anyone else can read it, but can't change it.

chmod

The command is short for "change mode".

You can change the permissions on a file you own. For example, you might sometimes need to make a script (a file which is a sequence of commands) executable before you can run it. Let's say we have written a script called "start" to start us up in sql and want to make it executable. We would enter

chmod u+x start

u is short for user.

man

Things are getting a little fiddly. We often need help with commands. The word for this is "man", short for manual. To get help with the switches for chmod, enter

man chmod

This will give you a screen's worth of the manual. You can page down through the text. If you've seen enough, you might want to escape. Hold down the Ctrl key and press Z. In other words, press

Ctrl-Z

This usually stops a process in Linux and returns you to the shell prompt.

You might prefer to use one of the many sites on the web for help with commands, for instance

http://www.zzee.com/solutions/chmod-help.shtml

Copying files

Get back into your praxis directory (from your lab1 directory).

To copy a file, use cp. This command needs two arguments. The first is the name (and path) of the file to be copied. The second is the destination. (As we've said above, the full stop represents the current directory.) To copy sam.txt into the praxis directory, enter

cp lab1/sam.txt .

Wildcards

We can use * as a wildcard, to stand for everything in a directory

Renaming files

One command is used to move or rename files: mv

To rename sam.txt, enter

mv sam.txt ever.txt

Check that the command has worked.

To move ever.txt to lab1

mv ever.txt lab1/

Deleting files

The command is rm (for remove).

Change into your lab1 directory.

You can only delete files for which you have write permission.

To delete ever.txt, enter

rm ever.txt

Check that the command has worked.

That's a start

Lets make a signature for your email.

Bring up a browser and go to https://webmail.hw.ac.uk/exchange

Log into your email. Go to Options on the left hand side. Type in your signature in the appropriate box. At minimum include your name, course and student number. If you want to be more adventurous go ahead but remember this is going out with every email both to friends and staff so be professional.

That's a start

Further learning of Linux is really up to you.

If you have time left. Get familiar with TurnItIn on Vision.

We should always leave things tidy, for the benefit of subsequent users. Don't just leave the machine without logging out.

Close gedit (File Quit).

Close your terminals (File Close window)

Log out (System Log out yourUsername)