

COMP09024 Unix System Administration Laboratory 1: Introduction



Introduction and Basic Commands

Learning Outcomes

- Virtual environment and login procedure
- Basic Commands: man, ls, cd, pwd, whoami, touch, echo

PLEASE RECORD ALL ANSWERS TO THE QUESTIONS FOR EACH LAB IN A LAB-BOOK FOR ELECTRONIC SUBMISSION. LAB STAFF WILL ADVISE YOU OF THE DETAILS

Important: **UNIX is CASE sensitive!** Be sure to follow the instructions given in the sheets exactly. You may find it helpful to turn off 'Caps-Lock', so that the standard is lower-case.

For all laboratories, we will use a web based virtual machine which run a real Linux inside of it. This web based virtual machine will remain available 24/7 for the duration of the course. The virtual machine contains the following usernames and passwords:

| Username | Password |
|-----------------|-----------------|
| root | qwerty |
| student | qwerty |

A terminal session should start within the Linux box:

```
Starting logging: OK
Initializing random number generator... done.
Starting network: OK
Starting dhcpd...
no interfaces have a carrier
forked to background, child pid 105
Starting atd: OK
Starting UWS ...

+---+---+---+ +---+---+---+ +---+---+---+ +---+---+---+ +---+---+---+
|U|N|I|X|I| |S|Y|S|T|E|M| |A|D|M|I|N|I|S|T|R|A|T|I|O|N| | - | |C|O|M|P|0|9|0|2|4|
+---+---+---+ +---+---+---+ +---+---+---+ +---+---+---+ +---+---+---+

[Lab 0]

Introduction

UWS login: root
Password:
Last login: Wed Sep 16 21:42:41 on console
[root@UWS ~]#
```

Introduce “student” as username and “qwerty” as password. You should obtain a shell:

```
[student@UWS ~]$
```

You can change all passwords at anytime and even destroy the full machine by removing all files (even the sensitive ones). Since this Linux it is being virtualized in your browser, you just need to refresh the browser to have a new fresh Linux (by pressing F5 for example). **There is no way you can damage anything so you have no excuses to play and learn!**

First Steps in System Administration

Being the system administrator (aka: 'super-user' or 'root') in a Linux environment is a very powerful (and fun) role. As this module aims to teach a generic Unix skill set, the majority of tasks will be carried out using the command line interface (CLI). Graphical User Interfaces (e.g. KDE, Gnome) may change, Unix commands however have to adhere to a standard (POSIX) and will not change in their core behaviour between different distributions. Furthermore, command line input is faster, more effective and much more versatile than any GUI interface.

So let us continue with some of the [Philosophy behind Unix commands](#):

There are many different commands that you can enter to configure and interact with Unix. The current Debian-Linux distribution has around 3000! These commands can be entered at the shell prompt, or command (line) prompt, or if you use a GUI, look for the shell program icon (as stated however, we will try to avoid the GUI-mode).

There are two types of commands, shell commands and Linux commands. Both of these are case-sensitive, like every ASCII-character in Unix (in contrast to ms-WInDoWs ;-)

Shell Commands

Shell commands are part of the shell program. There are several different shells (bash shell, bourne shell etc), and each will have a variation of the shell commands built in. The commands differ between shells, but are the same even across different Linux distributions for each shell.

Linux Commands

Linux commands are not part of the shell. Each one is a separate executable program, probably written in the C programming language. These executables are stored in various directories set up for binary files, such as `/bin` and `/usr/bin`. The location of these directories can be defined with the `$PATH` variable so the shells know where to find them. These commands vary between different Linux distributions, and remain the same whichever shell you are using. Though they can vary between different shells, the variations are generally of a marginal nature. You could think of it is different regional dialects of the same language.

The Command Prompt

What your prompt looks like will depend on the shell you are using (e.g. bash, borne, csh, tcsh) and the settings that you have chosen for this particular shell on your specific system. In our Lab we normally use a bash shell, and have the prompt set up to look like this...

```
[root@UWS ~]#
```

This shows that I am logged in as root (# prompt) on a computer called `UWS`, and I am in a folder called `root` (`~`) . TIP: To see the full path of the folder 'root', use the **pwd** command.

Your prompt will look a little different as you should currently be logged in as a normal (non-privileged user), indicated by the \$ prompt.

Linux Commands

Some Linux/Unix commands are very simple and comprise nothing more than the command itself.

For example the command **ls** (list) will simply list the names of the directories and files in the current directory...

```
[student@UWS ~]$ ls
Desktop
[student@UWS ~]$
```

For illustration we will consider only one file, a directory called Desktop. You can often add options or parameters, which are usually added to the command after a '-' , which is known as the **command line qualifier**. Always leave a space between the command itself and the qualifier.

Adding the qualifier **-l** (it is -l, the letter, not -1, the number) will give you a LONG listing which includes the permissions, ownership, size, date/time, and name of the files and directories... (details regarding the listing will be discussed in a later laboratory session)

```
[student@UWS ~]$ ls -l
drwxr-xr-x    2 root  root 4096 Aug 22 10:31 Desktop
```

Another example would be adding the option **-a** (**ls -a**) to list ALL of the files in the current directory, including hidden files.

```
[student@UWS ~]$ ls -a
. . . .bashrc Desktop
```

The **-a** shows hidden files (hidden files have a '.' before the filename) in this' directory. Their meaning will be explained in a later session.

You can use more than one option at a time...

```
[student@UWS ~]$ ls -la
```

would produce a LONG listing format of ALL (including hidden) directories and files in the current working directory.

There are many options for some Linux commands, and the best place to find out about them is the '**man**' pages. The man pages show you how to format the command, and what options and parameters may be used with each command. For example:

```
[student@UWS ~]$ man ls
```

The above command will show informatino about the *ls* command usage.

Below is a list of four commonly used commands with explanations aimed at Linux command prompt beginners.

| Command | Description |
|------------------------|--------------------------------|
| cd | Change directory |
| locate | Find files from a database |
| ls | List information about file(s) |
| pwd | Print Working Directory |

Let us first login as super-user or root in the system. To do so, we use the command **su** (switch user)

```
[student@UWS ~]$ su -
```

The system will ask you for a password. Again this is: **qwerty**

Now, the prompt will end with the hash sign (as the specifics of this log-in prompt will differ from machine to machine we will refer to it simply as

```
[root@UWS ~]#
```

from now on. (The hash sign actually represents the 'bash-shell')

As a first task, please check the man-pages of some of the commands given above by typing:

```
[root@UWS ~]# man pwd
```

(You may exit the man display by the <Control-Z> key sequence or the 'q' key. It is also possible to move through the manual pages using the arrow keys and page-up/page-down keys.)

```
[root@UWS ~]# man ls
```

Question

- **Q1.1)** Name and explain ONE qualifier selected by you for each of the three commands

Question

- **Q1.2)** Who are you (= what is your user name) in this setup?

Please use the **man** pages to confirm the action and meaning of the command **whoami**

```
[root@UWS ~]# man whoami
```

Now use the command:

```
[root@UWS ~]# df -k
```

Question

- **Q1.3)** What type of information does the command **df** display?
- **Q1.4)** What are the names of the different filesystems that are displayed? What is the mount-point for the filesystem source beginning **/dev/root**?
- **Q1.5)** What do the *Used* and *Available* columns stand for? (you can always call the Linux help for the df command: **man df** , man is an abbreviation for manual pages)

Linux Environment

The aim of the next group of tasks is to provide a brief introduction to creating directories and files and navigating a Unix system (using the **cd** command). The **pwd** command is introduced to verify the current directory. First of all set into the student directory by typing:

```
[root@UWS ~]# cd /home/student
```

```
[root@UWS student]# pwd
```

Now let us create a new directory, to be called **subdir**:

```
[root@UWS student]# mkdir subdir
```

Then move in to this new directory by typing:

```
[root@UWS student]# cd subdir
```

To confirm that you are actually within this directory type:

```
[root@UWS subdir]# pwd
```

Question

- **Q1.6)** What directory name is displayed by the **pwd** command? Give the full name including the leading / (=forward-slash).

To confirm the directory is empty type the **ls** command:

```
[root@UWS subdir]# ls
```

Now we will create two (dummy) files in our new directory. A useful command that achieves this is the **touch** command, which just creates an empty file.

```
[root@UWS subdir]# touch file1
```

Another useful command to instantly create small files is the **echo** command, type:

```
[root@UWS subdir]# echo 'Hello this is file2' > file2
```

Now type **ls** again, this time with a qualifier:

```
[root@UWS subdir]# ls -l
```

Question

- **Q1.7)** Use the **man** command **man ls** to find out what information is given by **ls -l**? Try to figure out which column shows file size (Tip: file1 created by the **touch** command should be empty, therefore taking up no space!). What is the size of the newly created file, file2?

Now type:

```
[root@UWS subdir]# cd ..
```

(yes, there are two dots ('..') after the **cd** command. Take care not to omit the space between the **cd** command and the two dots!

Question

- **Q1.8)** Use the **pwd** command to find out what happened and in which directory you currently reside? What is the meaning of the double dot: **..** if used in conjunction with the **cd** command? (Tip: the command prompt may give you a hint as well)

Now type:

```
[root@UWS student]# cd ./subdir
```

Question

- **Q1.9)** Use the **pwd** command to find out what happened and your current directory.

Now type the **cd** command without any arguments:

```
[root@UWS subdir]# cd
```

Question

- **Q1.10)** In which directory did the command: **cd** just move you?

Last but not least, we will go to the highest order directory (root-directory) of the filesystem via:

```
[root@UWS ~]# cd /
```


Questions

- **Q1.11)** Are there any hidden files in the root directory called ``/``?
- **Q1.12)** How many filesystems are there in total listed in the root directory called ``/``?

If you have time left, please check whether a connection to the Internet is established on your system. To do so open the firefox web browser.

So, that's all for the first lab. Please remember to put all of your answers in an appropriate Log-book, of acceptable format (Details will be explained during the first module session). Hope you enjoyed this Unix taster...

- END OF LAB -