

Overview

UNIX administration is not normally achieved using the graphical user interface (GUI). The reasons for this are several folds.

- Many systems have a 'stripped-down' operating system with no GUI.
- It is common for systems to be held on "dark sites" (s i t e s where no-one goes in the normal pattern of things) and the systems are managed remotely using telnet, ssh etc.
- GUI's limit the power of UNIX, you can do a lot more on a command line than you can do through a GUI. GUI's are getting better but still the average UNIX expert finds the GUI limiting for all but a few tasks.

The user of the machine must know how to interact with it by using UNIX command that are typed in next to the command prompt.

The aim of this workshop session is to get you familiar with basic UNIX commands so that you are able to interact with a UNIX machine from a simple command line prompt instead of by using a friendly and colorful GUI.

Logging in and out of the system.

To use a UNIX machine a user needs a *user account*. This comprises of:

- A user account name.
- A user account password.
- A directory that the user is automatically 'placed in' when he/she logs onto the system. This directory, referred to as the user's home directory, is where the user keeps his/her work.
- A number of administration files placed in the home directory that configure the user's working environment.
- Information entries for the user in certain system files (e.g. /etc/passwd)

Each UNIX system has a "superuser account" called **root**. People charged with managing UNIX systems are usually referred to as UNIX systems administrators. One obvious UNIX system administration task is to create a user account for each user of the system.

System directory hierarchy

The directory structure of a UNIX system can be viewed as a tree. At the base of the tree is 'the root directory' (which is the home directory of the user root) this directory will contain a number of files and other directories, each of these directories will contain additional files and other directories and on it goes it is a hierarchical directory structure. Don't get to lost in the tree analogy, the root directory is generally considered to be at the top of the hierarchy and all other directories are siblings of it.

The following example shows the top-level directory structure for Ubuntu-

Top level	Second level	Purpose
/		The root directory
	bin	Contains all of the code that constitutes the operating system
	boot	Contains the boot sector
	dev	Contains ' device files '. These are special files that the system uses to interact with hardware devices.
	etc	The etc directory contains files that are changed to configure the operating system
	home	Users' home directories go under here. Each home directory is named with the same name as the user account name. E.g. /home/os
	initrd	
	lib	
	lost+found	If directory tree gets corrupted and UNIX finds any lost files, it sticks them in here so the administrator can decide what to do with them.
	misc	
	mnt	
	opt	
	proc	
	root	
	sbin	
	tftpboot	
	tmp	Contains temporary files - all programs written for UNIX should use this directory to store temporary files. It sorts of keeps things tidy, many UNIX variants will delete all files in tmp when rebooted
	usr	

	var	Contains files and directories that change size quite a lot. For example, print queues and e-mails waiting to be sent are held in var
--	-----	---

Notice that the top level directory is / , **that** is root.

Issuing commands at the command prompt

The 'shell' is an interface between the user and the UNIX operating system. Commands are typing in at the command prompt and the shell interprets them and sends instructions to the operating system to perform the appropriate actions.

There are many different types of shell, such as, Bourne, Korn, C, Bash. Ubuntu by default uses the Bash shell which provides a convenient means of repeating previously entered commands...

- Use the up and down arrows to view and select a previously entered command.
- Use double pling (!!) to repeat the last command entered.
- Enter the **history** command to see a full list of previously entered commands. Choose a previously entered command from the history list by using its displayed index number prefixed with a pling character (!)

UNIX commands that give information

	Command	Description
1	hostname	Displays the name of the machine. Each machine on the network is given a unique name by the system administrator.
2	who	Displays who is logged onto the machine.
3	date	Displays the current date and time.
4	cal	Displays a calendar.
5	env	Displays a list of environment variables and their values. We will meet these again later
6	uptime	Displays the time duration that the machine has been turned on.
7	uname	Displays information about the computer system.
8	man uname	The man command gives information on the command that follows it.

Exercise 1 - Using UNIX commands that give information

Try out the following commands and make observation notes as you go along ...

	Command	Notes
1	Hostname	
2	who	
3	date	
4	cal cal 2019 cal 11 2019	
5	Env	Scroll up and down the screen and make a note of the values of the following Environment variables: HOSTNAME = SHELL= TERM= USER= PATH=
6	Uptime	What pieces of information are presented by this command? You may need to find out more about the uptime command using the man pages
7	uname -a	What pieces of information does this command display.

UNIX commands relating to directories

	Command	Description
1	Pwd	P rint (on the screen) the pathname of the w orking d irectory (i.e. the directory you are presently in).
2	ls	L ist contents of directories.
3	cd	C hange working d irectory.
4	mkdir	M ake a d irectory.
5	rmdir	R emove d irectory.
6	mv	M ove a directory (also used to rename a directory or file).

Exercise 2 - Using UNIX commands relating to directories.

	Command	Notes
1	pwd	Write down the pathname of your current directory. (The directory you are automatically placed in after logging on is referred to as your ' home ' directory.)

2	ls	List the contents of your current directory (should be empty at this stage).
3	mkdir testdir	Make a subdirectory named 'testdir'.
4	cd testdir	Change directory to the new directory.
5	pwd	Write down the pathname of your current directory.
6	ls	List the contents of your current working directory (should be empty at this stage).
7	cd	The change directory command, cd , on its own will move you back to your 'home directory'. Another way to return to your home directory is to use the command cd ~
8	pwd	Write down the pathname of your current directory (it should be your home directory)
9	ls	List the contents of your current directory (this should now show the name of the subdirectory named testdir)
10	cd / pwd ls ls -p	Change to the top most level directory (i.e. the root directory) Confirm where you are using the pwd command. List the contents of the current directory List the contents of the current directory and put a forward slash at the end of any entry that is a directory.
11	(Task)	Check that the names of the directories under the root directory agree with those listed in the theory page. If not, make a note of any differences.
12	cd pwd ls	Change to your home directory Confirm where you are using the pwd command. List the contents of the current directory
13	cd testdir	Change to the directory named testdir
14	cd .. pwd ls	Change to the directory one level up (as specified by the use of two dots). Confirm where you are using the pwd command. List the contents of the current directory
15	pwd ls mkdir testdir/play ls cd testdir/play pwd ls cd .. pwd ls mv play play2 ls	Check you are in your home directory. List the contents of the current directory. Create a directory named play underneath the one named testdir. List the contents of the current directory. Change directory into the one just created. Confirm where you are. List the contents of the current directory. Change up one level (ie. back up to the testdir directory) Confirm where you are. List the contents of the current directory. Rename the directory named play to play2 . List the contents of the current directory to check that the renaming has been done.
16	Note:	The command cd testdir/play uses the name of the directory relative to the current directory. To be more precise you could have specified the command as: cd ./testdir/play where the single dot means the current directory.
17	cd pwd ls	Move back up to your home directory. Confirm where you are. List the contents of the current working directory.
18	rmdir testdir	Remove directory. You will find that this command will not work since the directory contains subdirectories. Overcome this by first removing the lower

		level directories. Later you will be shown a potentially dangerous command that will remove any directory and anything underneath it.
19	pwd ls	Check where you are. List the contents of the current directory.
20	cd mkdir testdir mkdir testdir/play	Recreate the directories again.

Ubuntu Terminal Shortcuts:

To further ease up your skill, these Ubuntu Terminal keyboard shortcuts would help.

Ubuntu Terminal Shortcuts	Function
Ctrl + Shift + T	Open new tab on current terminal
Ctrl + Shift + W	Close the current tab
Ctrl + A	Move cursor to beginning of line
Ctrl + E	Move cursor to end of line
Ctrl + U	Clears the entire current line
Ctrl + K	Clears the command from the cursor right
Ctrl + W	Delete the word before the cursor
Ctrl + R	Allows you to search your history for commands matching what you have typed
Ctrl + C	Kill the current process
Ctrl + Z	Suspend the current process by sending the signal SIGSTOP
Ctrl + L	Clears the terminal output
Alt + F	Move forward one word
Alt + B	Move backward one word
Ctrl + Shift + C	Copy the highlighted command to the clipboard
Ctrl + Shift + V or Shift + Insert	Paste the contents of the clipboard
Up/Down Arrow keys	To scroll through your command history, allowing you to quickly execute the same command multiple times

TAB	Used to complete the command you are typing. If more than one command is possible, you can press it multiple times to scroll through the possible completions. If a very wide number of commands are possible, it can output a list of all possible completions.
-----	--