National University of Computer and Emerging Sciences Operating System Lab - 02

Lab Manual

Objective

This lab is all about running commands in Ubuntu Terminal and compiling C program in Ubuntu

Table of Content

Objective 1

Table of Content.....1

Shell 2

Commands in Linux 2

Patterns and Wildcards.....11

Pipe in Linux..... 12

Compile C program in Linux 12

Lab Activity 13

Shell

Fortunately, or unfortunately, a computer can only understand binary language and humans can easily understand English language or equivalent high level language and therefore it is difficult to interpret and understand with the computer system. In order to ward off this difficulty every operating system has got an inbuilt interpreter(Shell). A shell accepts instructions or commands fed by user in user understandable language and translate it to binary language which a computer can easily understand. So in short a Shell is a language translator and in this lab is all about introducing Shell of the Linux and the commands that are most commonly used.

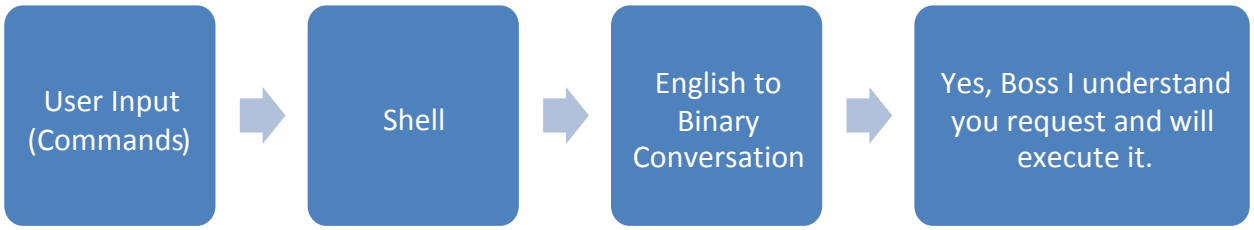


Figure 1 Shell - A diagrammatic representation

Commands in Linux

From here the reader is exposed to the basic Linux commands. All the commands have to be tried in the terminal. Throughout the lab manuals Ubuntu will be used for explaining the concepts.

NOTE: *All Linux commands are case sensitive i.e. 'cp' is not equivalent to 'CP'. Also, all the files and directories in linux are case sensitive so for example '/etc/hosts' is not equivalent to '/etc/Hosts' and so hosts and Hosts are two different files in the same directory.*

Command	Switch	Description	Example	Output
BASIC COMMANDS				
Manual/Help for any command				
man	None	Gives manual for the specified command	man mkdir	Opens manual in terminal, press 'h' for help or 'q' to quit and get back to terminal

Command	Switch	Description	Example	Output
Date and Time Commands				
date	None	Displays the system date	date	Sun Jan 19 22:11:00 MST 2014

	-s	Sets the date specified by the string	date –s “20 JAN 2014 18:00:00”	Mon Jan 20 18:00:00 GMT 2014
	+%d	Displays the day	date +%d	20
	+%m	Displays the month	date +%m	01
	+%y	Displays the year	date +%y	2014
	+%D	Displays the date in mm/dd/yyyy format	date +%D	20/01/2014
	+%H	Displays the hour in 24 hour format	date +%H	18
	+%M	Displays the minute	date +%M	36
	+%S	Displays the second	date +%S	40
	+%T	Displays the time in HH:MM:SS in 24 hour format	date +%T	15:20:20
	+%a	Displays the abbreviated weekday	date +%a	Mon
	+%A	Displays the full weekday name	date +%A	Monday
	+%b	Displays the abbreviated month	date +%b	Jan
	+%B	Displays the full month name	date +%B	January
Managing Users and Groups in Linux (root user only)				
useradd	None	Creates a new user profile or update existing user information	useradd abc	User Created
addgroup	None	Add a group to the system	addgroup example	Adding group `example' (GID 1003) ... Done.
adduser	None	Creates a user account that can be used for login	adduser username	Ask for password and some data along with confirmation and creates the account
	--ingroup	Creates user account and add that user in a group specified	adduser --ingroup sudo abc	Same as adduser <username> and also it adds the user to the group
usermod	-a -G	Modify an existing user	Usermod –a –G sudo abc	Add already existing user to already existing group

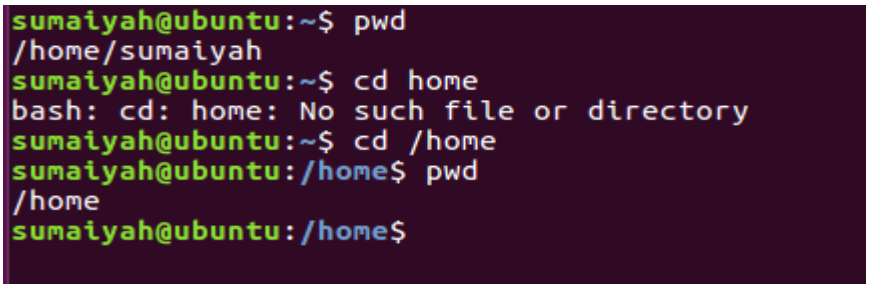
Command	Switch	Description	Example	Output
deluser	None	Deletes the user from the system	deluser abc	Removing user `abc' ... Done.
	--group	Deletes the group from the system	deluser --group example	Removing group `example' ... Done.
	--removehome	Removes the user along with its home folder directory	deluser --removehome abc	Removing files ... Removing user `abc' ... Done.
passwd	None	Change password of the current logged in user or user specified	passwd OR passwd abc	passwd: password updated successfully
Shutdown or Restart System				
shutdown	None	Power off the computer	shutdown now	The system will shutdown now for maintenance
	-r	Reboots after shutdown	shutdown –r now	None
poweroff	none	Shutdowns computer	sudo poweroff	None
reboot	none	Restarts computer	sudo reboot	None

Files and Directories in Linux

In generalized Linux file system, the basic unit is a file. It contains data about the file, essential metadata and non-essential metadata and some information. In Linux everything is a file. A directory is a special kind of the file.

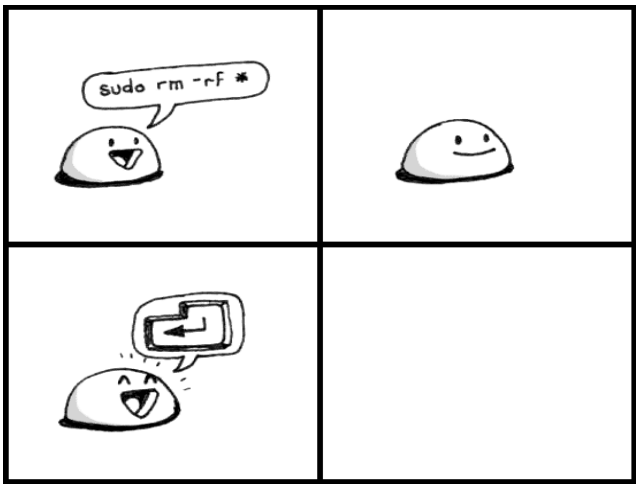
Relative and Absolute Paths

In Linux file system, when you type a path starting with a slash (/), then the root of the file tree is assumed. If you don't start your path with a slash, then the current directory is the assumed starting point. The screenshot below summarizes all.



MANAGING FILES AND DIRECTORIES IN LINUX				
File Basics				
touch	None	Creates a file	touch file1	File Created
	-t	Creates a file with given timestamp	touch –t 130207111630 BigBattle	File created
file	None	Determines file type	file HelloWorld.c	HelloWorld.c: C source, ASCII text
			file /dev/sda	/dev/sda: block special (8/0)
ln	None	Creates link of the file	ln file1 link1	None
	-s	Creates shortcut link of the file or directory	ln -s file1 slink1	None
			ln -s dir1 dirslink1	None
Displaying Contents of a File				
cat	none	Displays contents of file in the terminal	cat file1	<contents of file1>
head	none	Displays first 10 lines in terminal	head file1	<first 10 lines of the file content>
	-[number]	Displays first specified number of lines in terminal	head -20 file1	<first 20 lines of the file contents>
tail	none	Displays last 10 lines in terminal	tail file1	<last 10 lines of the file content>
	-[number]	Displays last specified number of lines in terminal	tail -17 file1	<last 17 lines of the file contents>
Copy, Move, Rename or Remove Files or Directory				
cp	None	Copies a file	cp fileA fileB	None
	-r	Copies a directory	cp -r dir1 dir2	None
	-i	Copies files but prevents overwrites	cp -I a.c b.c	None
mv	none	Moves/renames files and directories	mv fileA ~/fileB	None
			Mv dirA dirB	None
rm	none	Removes a file	rm file1	None

	-r	Removes a directory	rm -r dir1	None
	-rf	For removal of write-protected files	rm -rf dir1	None



Directory Basics				
pwd	None	Determines the current path	Pwd	/home/alishah/Desktop
mkdir	None	Creates a directory in current or specified directory	mkdir dir1	None
			sudo mkdir /home/dir1	None
	-p	Creates directory or directories in tree hierarchy manner	mkdir -p dir1/subdir/subsubdir	None
	-v	Prints info about the directory being created	mkdir dir1	mkdir: created directory 'dir1'
ls	None	Displays the content of current directory or specified directory	ls	<content of current directory>
			ls /etc	<content of /etc directory>
	-l	Displays the content in long format and with detail	ls -l	<contents with detail like owner, creator etc>
	-a	Displays the content along with hidden content of current or specified directory	ls -a	<content of current directory>
	-h	Displays the content in human readable form	ls -h	<contents>

	-R	Displays the content in recursive order (it list file and directories along with files and subdirectories of subdirectories and so on)	ls -R	<contents>

File/Directory Permissions and Ownerships

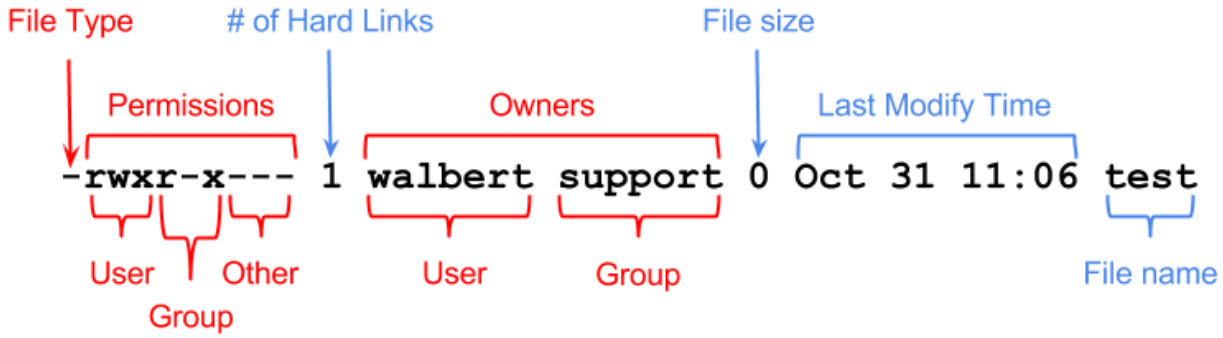
Every file created in file system has an owner and permissions associated with it. There are basically three kinds of user available in Linux

1. Owner (User who created the file/directory)
2. Group
3. Other Users/Groups

Each of the above-mentioned user will have access permissions. Following are the three permissions associated with all the files.

1. Read (Denoted by r)
2. Write (Denoted by w)
3. Execute (Denoted by x)

These permissions can be visualized by 'ls -l <file/directory name>'



Let us examine '-rwxr-x---' the first '-' represent that it's a file 'd' would represent that it's a directory, the next 3 characters 'rwx' are the rights for the owner, next three are the permissions of the group and last three characters are the permissions for the other users/group.

The third column represents states the user who is the owner of the file. Now the question is: can I change the permission or ownership of a file or directory. The answer is 'yes!'

Chmod can be issued in two different ways, First method is 4 2 1 code in digital electronics

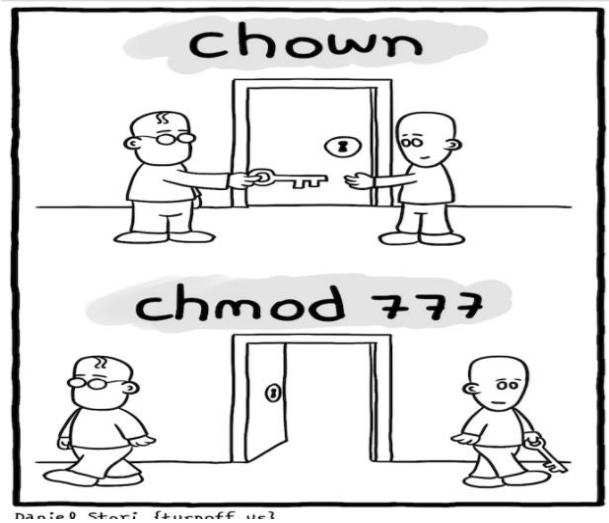
4	2	1
r	w	X
1 or 0	1 or 0	1 or 0

This is really simple, if a user has to be assign with all permission (Read, Write and Execute), 1 has to be applied in all the permissions that are required: 1(r) + 1(w) + 1(x) = 1(4) + 1(2) + 1(1) = 7 so 7 is the number that will fetch all the permissions for that file or folder.

Owner			Group			Other		
4	2	1	4	2	1	4	2	1
R	w	x	r	w	x	R	w	X

Assuming that all the users get rwx permission so 4+2+1 = 7 will get mathematically 777. Below table shows the syntax and example of using chmod command and also how to change the owner of the file i.e. chown command.

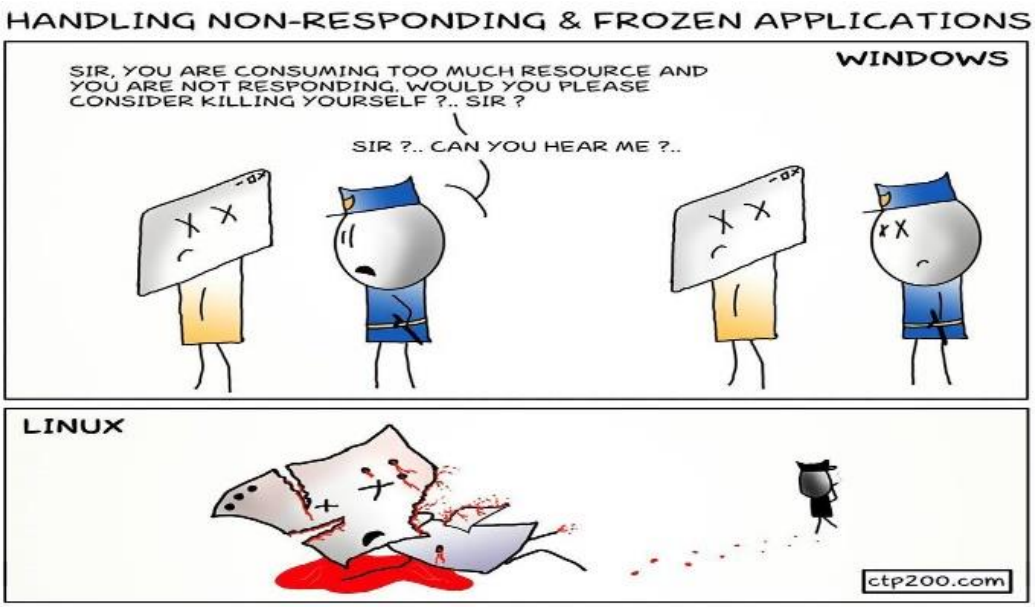
Command	Switch	Description	Example	Output
chmod	None	Changes permissions of a file	chmod 700 file1	None
	-v	Output a diagnostic for every file processed	chmod -v 650 file1	mode of 'file1' changed from 0700 (rwx------) to 0650 (rw-r-x---)
	-R	Changes permissions files and directories recursively	chmod -R 760 dir1	None
chown	None	Change the ownership of a file	chown username filename	None
	-R	Change the ownership files and directories recursively	chown -R user dir1	None



Other Useful Commands in Linux

Command	Switch	Description	Example	Output
grep	None	Search pattern in a given file	grep 'hellow' file1.txt	<search results>
	-i	Search given pattern in a file ignoring case	Grep -i 'hEllow' file1.txt	<search results>
	-r	Search given pattern in all the files in a directory recursively	grep -r 'helllow' dir1	<search results>
	-w	Search words only not strings	grep –w hello cricket.txt	<search results>
	-c	Show match count for pattern	grep -c hello cricket.txt	<search results>
cal	None	Get the calender of the current or specified month and year (only month will not do)	Cal	<calendar of current month and year>
			cal 9 2020	<calendar of September 2020>
			cal 2020	<calendar of year 2020>
whatis	None	Gives a brief description of command	whatis ls	ls (1) - list directory contents
whereis	None	Gives the path of the Command	whereis ls	ls: /bin/ls /usr/share/man/man1/ls.1.gz
ifconfig	-a	To know the status and configurations of network interfaces	ifconfig -a	<output>
finger	None	To know about user account in Linux Users	finger username	<output about username>
ps	None	Show snapshot of running processes	Ps	<process with PID output>

	-A	Show all the processes	ps -A	<processes with PID output>
kill	None	Kills the process with specified process id	kill 1434	None
alias	None	Renames a command	alias l='ls -al'	None
unalias	None	Undo renaming a command	unalias l	None
df	None	Shows detail of disk usage. df works by examining a directory entry, which generally are updated only when a file is closed.	Df	Filesystem 1K-blocks Used Available Use% Mounted on …
du	None	Estimates file space usage. Output the summary of disk usages of every file hierarchically i.e. recursively	Du	<output>
mount	None	It is use to mount a file system that do not mound itself	mount /dev/sda5 or mount /dev/usb	None
sudo	None	Runs the command as root/super user/administrator	sudo cp ~/Desktop/file /usr	None
	-i	Login as root user	sudo -i	<ask for password>
su	None	Change username or become a super user	su username	<logs in to username>



Patterns and Wildcards

Patterns aka regular expression uses wildcards to represent unknown values. Wildcards helps the user to perform certain operations with specifying filename or text pattern. There are three special characters basically made available for this purpose.

1. * - will match against none or one or a string of more than a character
2. ? - can be used to match one character
3. [] - matches one specified character out of a group of characters

WildCard '*'

- '$ ls file*' - list all the files in current directory starting with filename 'file'.
- '$ ls *2.txt' - list all the files in current directory ending with '2.txt'

WildCard '?'

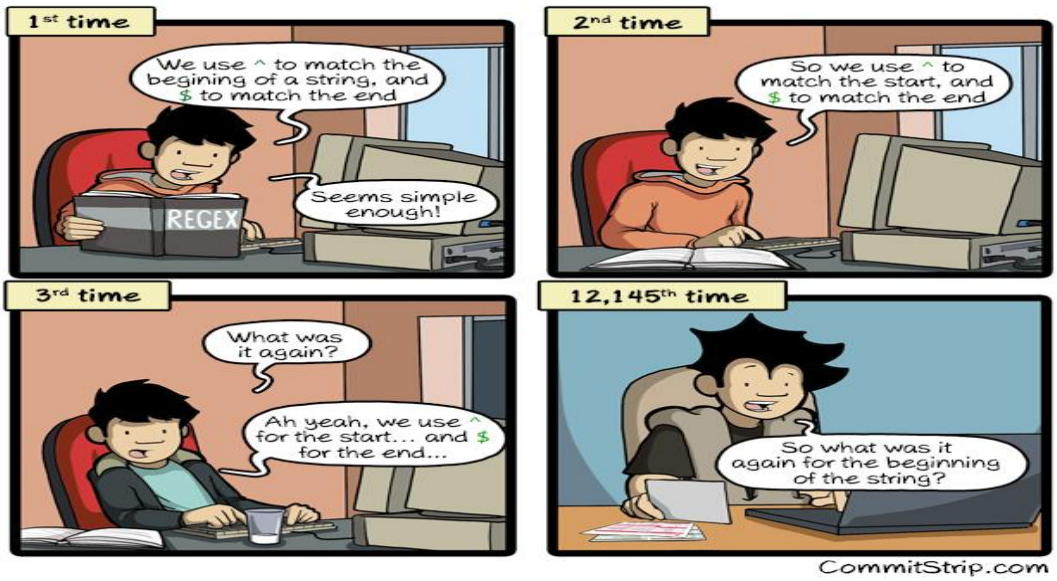
- '$ ls file.tx?' - list all the files that begins with 'file.tx'

WildCard '[]'

- '$ ls rmt[12345]' - list all the file that begins with 'rmt' and has a 1,2,3,4 or 5 after it.

```
sumaiyah@ubuntu:~$ cat sumi
Hello World
fastian
sumaiyah@ubuntu:~$ grep ^H sumi
Hello World
sumaiyah@ubuntu:~$ grep ^f sumi
fastian
sumaiyah@ubuntu:~$ grep d$ sumi
Hello World
```

```
student@OSLAB-VM:~$ ls | grep '^[A-Z]'
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
```



Pipe in Linux

If a user in Linux likes to combine two or more commands, pipes is the option. Pipe is represented by the symbol '|'. Let us look at the example below:

$ cat file1.txt | grep 'world populations'

$ ls -al | grep 'mp3'

$ ls | grep 'mp3' | sort -r

Compile C program in Linux

This session will show how to write a C program, compile the program and how to execute it using terminal.

1. Open the terminal and create a file with 'c' extension.

```
$ nano hello.c
```

2. Write the following text to the file:

```
#include<stdio.h>
int main() {
printf("hellow world from Cprogram");
return 0;
}
```

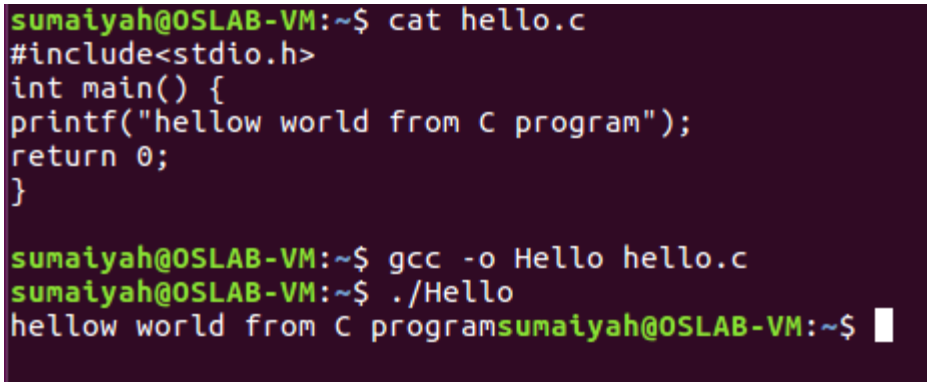
3. Compile the file and create an executable object file

```
$ gcc -o Hello hello.c
```

4. Run the newly created object file

```
$ ./Hello
```

The snapshot of the terminal as below:



Lab Activity

1) User Account
    a. Create a group name ‘OSLAB02’
    b. Create a user account ‘OSUser1’ and ‘OSUser2’ and add it to the group which is created in ‘a’
    c. Also add the newly created user to group ‘sudo’
    d. Login in to that user using terminal

2) Create the following directories with one command. dirOSLAB -> subDir -> subsubdir -> OSLAB2

3) Write 2 C program one prints “I love Operating System” and other prints “I love Linux”. Compile and Run both programs and print the output to two different files. After then combine both the files in one new file using a single command.

4) Perform the following activity
    a. Create user ‘abc’
    b. Create a file ‘file1.txt’
    c. Change the owner of the file to newly created user “abc”
    d. Rename a file ‘file1.txt’
    e. Create a file with timestamp
    f. Make a copy of /proc directory
    g. Write a command to delete non-empty directories.
    h. Create a dummy file and then change the ownership of the dummy file.
    i. Determine the process id of the user from which you are logged in and then terminate that process. What happens after terminating the particular process id?
    j. List all files in system having string ‘lab’ in their filenames.
    k. Determine the storage capacity utilized in system.

5) Perform the following
    a. List the files in the directory "/bin" that end in "sh".
    b. On one line, use the "cd" command to first go to your home directory then to the "<rollnumber>" subdirectory. [Ans: cd /home; cd <rollnumber>]
    c. What command lists the files in the current directory that begin with upper case letters?
    d. If they do not already exist, create three new directories "Letters", "Programs", and "Misc" using a single command
    e. Copy all files in the current directory whose names contain the character string "let" into the subdirectory "Letters".

f. Copy all files in the current directory whose names end in ".c" or ".h" into the subdirectory "Programs".
g. Copy all files in the current directory whose names contain the character strings "notes" or "misc" into the subdirectory "Misc".
h. Copy all files which begin with "copy.me" into the "OS" subdirectory. Move all files which begin with "move.me" into the "OS" subdirectory.
i. Delete all files which contain the sequence "del".

Coursera

- The Unix Workbench (cover Week 1 and 2 only for Lab 2)

https://www.coursera.org/learn/unix
https://seankross.com/the-unix-workbench/