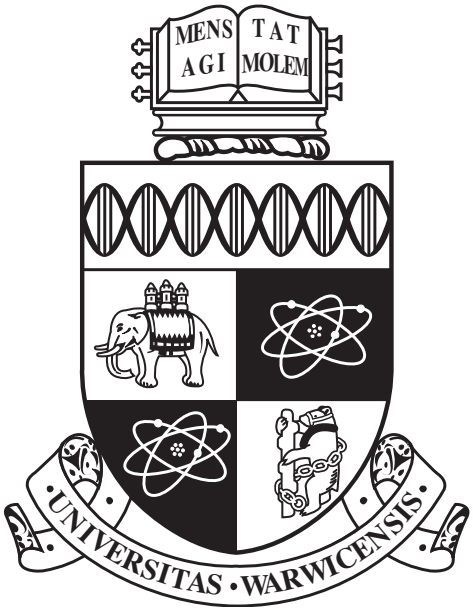
University of Warwick
Department of Computer Science

CS133

Professional Skills



Cem Yilmaz
December 17, 2021

1 Linux

1.1 What is Linux?

Linux is a free version of UNIX developed by a developer called Linus. They're compose of three components:

1. Kernel;
2. Shell: interface to kernel;
3. Utilities: word editors etc.

Definition 1.1. Shell

A shell can be seen as different things:

1. Command interpreter;
2. Friendly hooks to kernel;
3. Programming language

Some examples of shells include bash, Gnome, csh etc.

2 Linux Concepts

2.1 Files

Files are used to store data, it can contain anything. Data are sequences of bytes.
These files can be classified as:

1. Regular: text, binary code, HTML, Java, etc...;
2. Director;
3. Device

Each "file" has a unique ID, called inode. This is usually a number. All fires are in fact tree hierarchies. However, it is possible that directories can lead to the same files. This is when the also share inodes.

2.2 Basic commands

2.2.1 Terminal Commands

```
cd \\ current directory
pwd \\ print working director
mkdir \\ make a directory
rm \\ remove. Can also be used to do -d and -R to delete dir or recursively.
rmdir \\ same as rm -d but only works on empty dirs
cp \\ copy, can be also be used to do -R.
mv \\ move, moves a fily. Can be used to rename
ls \\ list, lists all files. You can use it to view vermissions with -l. Secret files w
```

2.2.2 Symbols

```
. \\ current working directory
.. \\ parent directory
~ \\ home directory
/ \\ root
```

2.3 Permissions

Permissions are set up with binary code, specifically 111.

Definition 2.1. Permissions

A permission is defined by

$$\underbrace{-}_{\text{Type}}\ \underbrace{---}_{\text{User}}\ \underbrace{---}_{\text{Group}}\ \underbrace{---}_{\text{Other}} \tag{1}$$

where $---$ represents permissions, listing from rwx, where r is read, w is write, x is execute.

The command $chmod$ is then used to change permissions. u represents user, g represents group, o represents others and a represents all. So, for example, one can run the command

```
chmod u+rwx,g-r,o+x ~/example.txt
```

And for this, would add rwx for user, remove r for group and add x for others. Alternative ways of using chmod including converting $111, 111, 111$ system (without the ,) to base 10, e.g or instead of doing $\pm$, you can use $=$. A tip is to use the man command, which lists a list of commands that you can use in the terminal.

2.4 Process

Every process has a special id number (PID, process ID) and are divided into streams. The process is entered through a standard input stream, and the result is then given by a standard output stream. The streams are:

1. Standard Input stream - stdin, no.0
2. Standard Output stream - stdout, no.1
3. Standard Error stream - stderr, no.2
4. Other streams and I/Os...

2.5 Input

Suppose we have a command called $myScript$ and the file f. We can redirect input to $myScript$ from f using

```
myScript < f
myScript 0< f
```

You can create more streams by assigning more numbers to them like done above.

2.6 Output

Files are created implicitly as and when required.
We can overwrite the file

```
myScript > f
myScript 1> f
myScript 2> f
```

We can append to the end of the file

```
myScript >> f
myScript 1>> f
```

Current processes can be checked using the ps command. You can also use the command top to sort processes by system usage of CPU. You can use the command $kill$ to kill a process.

2.7 Path