Linux Tutorial

Info:

Within my worksheets (and many other worksheets in the uni) you will see things like "<VARIABLE>". This is to signify a place where you need to replace this with the appropriate text for your use case.

What you're going to be using Linux for

Primarily, you are going to be using some form of Linux terminal to write code and install software. Most of you are going to be using Ubuntu, but there are loads of Linux flavours out there! You'll be learning a lot more about this in TB2 Software Tools, but up until that point you're going to need a basic knowledge of how to do simple tasks. In this session, we will aim to give you the know-how to use a Linux operating system for the tasks you will need to cover in TB1. You should only need to use a terminal in the Programming in C module and so we will be focusing on that.

Interested in Linux flavours? Check them out here
https://en.wikipedia.org/wiki/List_of_Linux_distributions

Your tasks

Preferrably, you will each have your laptop with you and you can use your own linux terminal for this exercise. If you don't, then don't worry. You can skip out the installation stuff as the lab machines won't let you for security reasons.

The basics:

Using the information on this sheet, complete the following tasks:

1. Open a terminal
2. Navigate to your home directory with one command
3. create a folder called "temp", and then another folder called "files" inside temp. Can you do this with one command?
4. Navigate to the "files" directory
5. Create a file called "test" in the "files" directory
6. Move this file to the "temp" directory
7. Make a copy of test in the "files" directory (This is useful for copying makefiles)
8. Delete the "temp" directory and all of its files

The fun bit:

Learning how to use a terminal can be a bit boring, so after you have completed the above tasks, download and unzip the folder here:
https://gitlab.com/slackermedia/bashcrawl

This is a text-based adventure game designed to teach you how to use a Linux shell. It touches on a lot of stuff you won't use again until TB2 but it's a great way to get used to the commands you will be using a lot anyway.

Have a look at the README.md file and start playing!

Installing software:

Using your web browser and your terminal, install Teams on your Ubuntu system. If you already have Teams then try Discord or something you feel might be useful. You are encouraged to speak to those around you if you get stuck and see if you can find a working solution.
Also, try running "sudo apt install fortune" and then run the command "fortune"

Commands and usage

The following are a condesnsed list of useful commands. There are myriad commands for Ubuntu, some of the key ones can be found here https://cheatography.com/davechild/cheat-sheets/linux-command-line/

Openning a terminal:
Press `Ctrl+Alt+T` to open a terminal

Alternatively, you can navigate to a folder graphically using the file system and right-click and "open in teminal".

"My terminal is frozen!"
Chances are you've accidentally typed Ctrl+S in your terminal. For historical reasons, this completely stops terminal input (and is very annoying). To unfreeze the terminal press Ctrl+Q.

Force-quit an application:
Ctrl+C

Copy and paste:
Copy = Ctrl+SHIFT+C
Paste = Ctrl+SHIFT+V
Note – you may not be able to copy and paste things in and out of a virtual machine.

Run the last command again:

```
!!
```

Note – you can also cycle through your previous commands using the arrow keys

Auto-complete a command:
When typing a file or folder name, press TAB to see if you can autocomplete. This feature only works if the current argument you are typing only has one viable solution.

Clear the terminal:
Ctrl+L

Making Folders:
To create a folder, use:

```
mkdir <FOLDERNAME>
```

Change Directory:

```
cd <FODLERNAME>
```

Note: using "cd" on it's own will take you back to your home directory.

Listing contents of current directory:

```
ls
```

This will list the files and folders in the current directory – Useful if you need to remember the name of a file/folder!
"ls -a" will list the same contents as "ls" but include all of the hidden files. This is usually folders with a "." at the start. When you initialise a git repositiory in a directory, a ".git" folder will be placed there.

To view the contents of all the folders in the directory, use:

```
ls *
```

To view alll the files in a specific directory use:

```
ls <DIRECTORYNAME>
```

Print the current working directory:

```
pwd
```

Remove files:

Remove single file:

```
rm <FILENAME>
```

Remove directory and all it’s contents:

```
rm -r <DIRECTORY_NAME>
```

Moving a file:

```
mv <FILENAME> <DIRECTORYNAME>
```

Make sure the directory name is relative to the current folder. You can move multiple folders away in the followng way:

Move file back a couple of levels and then go into another directory:

```
mv ../../<DIRECORYNAME>
```

The “..” means go up a level. The closer you move to your systems root directry, the more you are moving “back” or “up” the file system, the more you layers of directories you go into, the “deeper” into the file system you are.

Copying a file:

```
cp <ORIGINAL_FILENAME> <NEW_FILENAME>
```

like moving a file, copying a file will require 2 arguments. The new filename is up to you, and you can also make a copy of a file into another directory like:

```
cp test folder/new_test
```

Installing new software

Installing new software can look pretty strange, so we're going to break it down into components and show you what each part means. Learning how to install stuff on Linux is different to Windows or Mac. Most of the time there isn't something you can just double click and have the system do it for you. The general flow of installing something is:

1. sudo <INSTALL_COMMAND> <PAKCAGE_NAME>
2. Wait for it to ask if you're sure (type 'y' and enter)
3. Watch a load of miscelanous text roll down your terminal while it installs

The most crucial thing to bear in mind when installing something in a Linux environment is that you are looking for the EASIEST solution. To illustrate this, I will walk you thorugh the following example of how to install SDL2, which you will need for Programming in C in later weeks.

For some of you, you will not have very much installed on Ubuntu at all. In some cases, not even enough to install most new software. To fix this, run:

```
sudo apt update

sudo apt upgrade
```

These commands update your machine's list of all the packages it needs to update and then updates them.

The commands to install SDL2 are:

```
sudo apt-get install codeblocks

sudo apt-get install build-essential libsdl2-dev
```

First of all, what do these commands mean?

- `Sudo` – Run the following command with "sudo" privileges. Sudo is short for "super user" and refers to the user with the highest authority. We don't run around with sudo priviledges for everything because in a Linux environment you can end up doing some serious damage to your computer.
- `Apt-get` – Goes and fetches all of the packages required for installation
- `Install` – Installs the software
- `<PACKAGE_NAME>` - The name of the package you want to install.

Second of all, why does it work for everyone else and not for me?

At some point in your Linux-using career, a piece of software will seem infuriatingly impossible to install. The above commands have worked for me, but I've seen other examples where one person just can't seem to install the software for whatever reason. This is an unavoidable part of learning software engineering and it's how you deal with this problem that is mportant!