Worksheet 1: Intro to Linux, Vim and Java

Updated: 25th February, 2020

The objectives of this practical is to:

- Provide an introduction to Linux, Vim, and Java.
- Practice terminal commands.
- Practice using Vim.
- Build your first piece of code!

Note: You must submit this practical worksheet **by the start of your next registered practical**, and have it signed off during that session.

Submissions must be done **electronically** through Blackboard.

There are 2 submissions due this week:

1: You are required to submit your `P01` directory as a gzipped tarball (`tar.gz`). It must contain all the work you completed for this practical. As well as your **`.bash_history`**.
Your tutor will go through how to do this with you in class.

2: You are required to submit your Assignment Questions (See section 7) at home.

You must also not modify the files in `PDI\P01` until after your submission has been marked. During your practical your tutor will check the modified dates of the files using `ls -l` (el) before marking your work.

To create a gzipped tarball use the following command from your `PDI` folder:

```
[user@pc]$ tar -cvzf <studentID>_P01.tar.gz P01
```

For more information on what each argument of the above command does use:

```
[user@pc]$ man tar
```

To get your **`.bash_history`** into a submittable format, first close all terminals down, using <ctrl>-d. Then open a new terminal, and type this command from anywhere:

```
[user@pc]$ history >~/Documents/PDI/P01/BashHistoryP01.txt
```

Your submission will be marked during your practical session via an interview from your tutor comprised of a few questions.

Note that the questions asked in the interview may cover the entirety of the worksheet, not just the material you submit. Your submitted work will be assessed and marks will be allocated accordingly as per the assignment question.

1. Logging onto Linux

Use your Oasis username and password to login to the lab computers. If you have any issues logging in let your tutor know.

Now open a terminal window, can you work out where it is? You can add it as a shortcut by dragging the terminal from the menu to the task bar.

In this unit you **must** use the terminal for all navigation, and manipulation. This includes using terminal based text editors.

2. Accessing Resources

> **Note:** You will need to use the API later so do not ignore this!

The University lab machines have Firefox installed for you to use. There will be a few webpages that you will be using throughout the semester and it is advised that you bookmark them for easier access.

Blackboard - `http://oasis.curtin.edu.au`
This will need to be accessed through Student OASIS, in the **My Studies** tab. This is where all of the unit content can be accessed (including this practical!). **All announcements concerning this unit will also be made here.**

Java API Documentation - `https://docs.oracle.com/javase/7/docs/api/`
This is the documentation for Java, this includes all the functions that are included with Java and can be used throughout the semester. Take a while to look over this, especially the Math and String library as these will be utilized during the semester.

StackOverflow - `http://stackoverflow.com/`
This is a community driven website that answers programming related questions. If you have a question, its guaranteed that someone has probably asked it (and had it answered) on this site. **Please do double check with a tutor the correctness of this answer** as it may be more advanced than needed for the unit, fail to comply with the *enforced* coding standards, or even be totally incorrect. Stackoverflow is often a good starting point, even if it just provides an idea of where to go next.

Wikipedia - `http://en.wikipedia.org`
A useful website when you have no clue where to start. Be aware it is a community run resource. **Curtin does not allow you to reference wikipedia**. A good idea is to use wikipedia to get you started, then go and have a look at the references. If there are no references **DO NOT TRUST IT**.

3. Introduction to Linux & Basic Directory Structure

Using *only* the terminal it is now time to construct a directory structure you can use to keep the semesters work organised. A *sample* of the unix commands available to you have been provided below, there are many more commands you will use through out the unit, the < > braces are just a convention to show something that you fill in. Note that in Unix folders are referred to as *directories*.

Important Note: The current directory is referenced by a single fullstop (`.`), the parent directory is referenced by two fullstops (`..`) and all pathways are relative to the current location.

Command	What it does	Example

`ls`	Lists all files in the current directory, if an argument is provided lists all files in the specified directory.	`ls`
		or
		`ls P01`

`cp <source> <dest>`	Copies the file from *source* to *dest*.	`cp Downloads/Test.java Documents/`

`mv <source> <dest>`	Moves the file from *source* to *dest*. If the *dest* ends in a file name it will rename the file.	`mv ../Downloads/TFA.avi Videos/Awesome`
		or
		`mv Test.java SimpleTest.java`

`pwd`	Lists the directory you are currently in (*Print Working Directory*)	pwd

`cd <directory>`	Moves to *directory* (this can be a path specification). *Change Directory*	`cd PDI`
		or
		`cd Documents/PDI/P01`

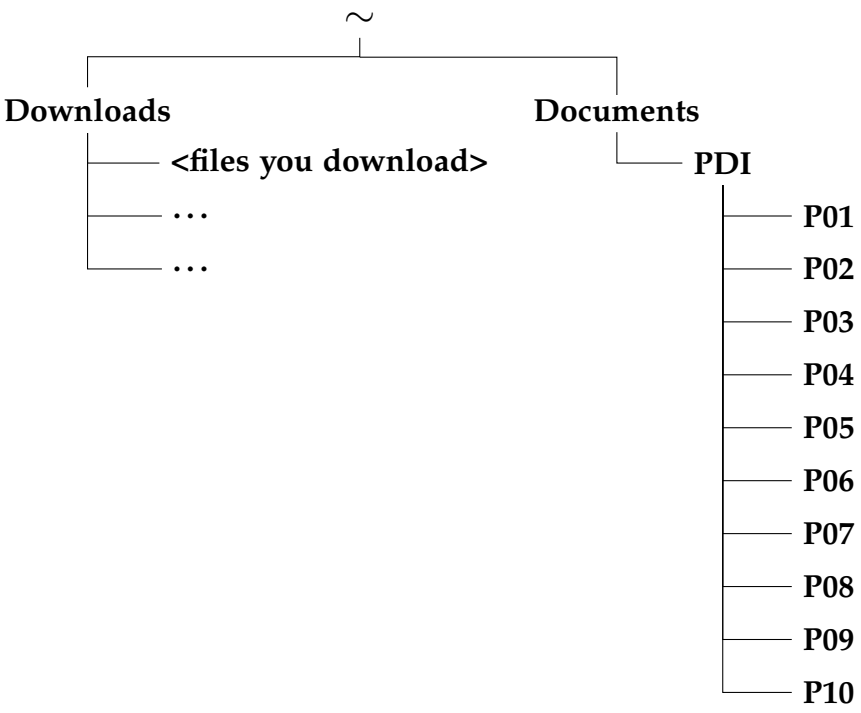
`mkdir <directoryName>`	Creates a new directory.	`mkdir PDI`
`rm <fileName>`	Removes a *file*.	`rm SimpleTest.java`
`rm -rf <folderName>`	Removes a *directory*. be **VERY** careful with this command as you are forcibly removing something and you cannot get it back.	`rm -rf PDI`

The best way to learn these commands are to regularly use the terminal. Your tutor will go through these commands with you in depth before you continue with the next step.

Using the above commands create the following directory structure, $\sim$ refers to your personal user area. An introduction to Linux has been uploaded to blackboard, it will be helpful if you get stuck.

Warning: If your tutor catches you using a GUI/mouse they will delete it and make you start again.

(You may use the mouse in your chosen internet browser.)



4. Setting Up Vim

For this unit you are required to use vim as your text editor, as per the unit outline. Vim is a highly customisable terminal based editor.

Vim has two modes, *normal/command* and *edit*. When in command mode you can use a variety of editor commands to move around and manipulate the text, you enter this mode by pressing <Esc>, and it is the default mode the program opens in. In order to add or edit the text you enter edit mode by pressing <a> or <*i*>. Here are some of the commands you will need to use in this (and other) practicals.

Command	Meaning
<Esc>	Enter command mode
a or i	Enter edit mode.
G	Moves the cursor to the last line of the document.
gg	Moves the cursor to the first line of the document
:num	Moves the cursor to the line specified by <num>
yy	Copies the current line.
dd	Cuts the current line.
p	Pastes the line.
u	Undo.
<Ctrl>+r	Redo.
/<search>	Highlights everything matching the search term. Use <n> to move to the next item and <N> to move to the previous item.
:w	Saves the document.
:q	Quits. (Use :wq to save and quit at the same time or :q! to force quit (when you don't want to save))

Note: The difference between Insert mode <i> and Append mode <a>.
<i> will allow you to type before the cursor (as the cursor itself takes up 1 character)
<a> will allow you to type after the cursor (append)

Vim has a settings file called .vimrc, we are going to modify this file in order to increase the usability of vim. Navigate back to *your* home directory (~), the easiest way to do so is to use cd with no arguments.

```
[user@pc]$ cd
```

Now you need to open the .vimrc file.

```
[user@pc]$ vim .vimrc
```

Enter edit mode by pressing <i>, then add the following text to the file.

```
1 syntax on
2 set tabstop=4
3 set softtabstop=4
4 set expandtab
5 set number
```

Once you are finished, return to command mode by pressing <Esc>. Then save your changes with the command <:w>. **Do not exit vim**.

The last step is to source the file in order ensure there are no errors. To do this, press <Esc> to enter command mode and then type.

```
[user@pc]$ :source .vimrc
```

If no errors occur then you can enter `:q` to exit vim and return to the terminal.

> **Warning:** If any errors messages appear, ask your tutor to come over as something has gone wrong!

You may also have multiple terminals open, one to edit and one to run your program. This is called split-screen. You just need to make sure that in the terminal you are editing, you save before you switch to the other one, as they are different instances.

> **Note:** You **must** use 4 spaces, not tabs in this unit.

5. Introduction to Pseudo

Now that you've had a chance to go through some terminal and vim commands, it's time to write your first piece of pseudo code!

Once you're out of vim and back in your terminal, make sure you're in your `~/Documents/PDI/P01` folder. Ask your tutor if you are unsure about where you currently are or how to get to this directory.

> **Note:** The **pwd** command may be useful to you here.

To get started on your pseudo, you must first create a `.txt` file in vim. Over the semester you will create to main types of files: `.txt` and `.java`.

In your terminal type:

```
[user@pc]$ vim Calculator.txt
```

This will create a blank text file for you in vim to type your pseudo code into. Below is some pseudo code you will need to type into this text file.

Remember: make sure you press <i> to enter edit mode.

```
START Calculator
    MAIN:
        OUTPUT "Please enter the first number: "
        INPUT numOne

        OUTPUT "Please enter the second number: "
        INPUT numTwo

        answer := numOne + numTwo

        OUTPUT "The answer is " numOne " + " numTwo " = " answer
    END MAIN
END
```

Make sure you copy everything exactly as it is above, including spacing and cases. Take note of certain words that are in caps. What do they mean? Why are they in caps?

Once you have written everything in that file, press <Esc> to enter command mode followed by <:wq> to save and exit your file.

6. Introduction to Java

Now that you've made a pseudo code file with your pseudo code written in it, it's time to write the Java equivalent. Just like the previous question, below is the Java code you are required to write. Make a new Java file with the command:

```
[user@pc]$ vim Calculator.java
```

This will create a new blank file for you to write your Java code into.

```
1  import java.util.*;
2
3  public class Calculator
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8          int numOne, numTwo, answer;
9
10         System.out.print("Please enter the first number: ");
11         numOne = sc.nextInt();
12
13         System.out.print("Please enter the second number: ");
14         numTwo = sc.nextInt();
15
16         answer = numOne + numTwo;
17
18         System.out.println("The answer is: " + numOne + " + " +
19                            numTwo + " = " + answer);
20     }
21 }
```

Just like the pseudo code, make sure you copy everything including the spacing. Don't worry if the colours don't match exactly, they're not meant to. As long as there is some syntax highlighting, that's okay.

What are the differences between the pseudo you wrote and this Java code? If it's the equivalent, why are there more lines here we didn't write earlier? What things are similar between the two?

Now that you've written your pseudo and Java code, it's time to test your program!

Make sure you save your file and exit vim. Whenever you test a program, you must always compile your code. Even if you make a small change, you must compile your program again for that change to be reflected.

> **Note:** You do not need to compile your pseudo as it is only a template for your Java program.

In your terminal type:

```
[user@pc]$ javac Calculator.java
```

The **javac** command followed by the name of the Java file will compile that Java program. This will create a new file in your directory called `Calculator.class`. This is the file that the Java Virtual Machine will use to run your program. It is also why you must always compile your program before running it as it will update your `.class` file with your changes.

To run your compiled Java program, type:

```
[user@pc]$ java Calculator
```

The **java** command followed by the name of the **.class**/**.java** file tells the terminal to run that class file. When you run the program, follow along with what the output says and press enter when responding to the program's outputs.

(a) What sort of outputs did you get from the program?

(b) What did it ask you to enter?

(c) What did it do with the input you gave it?

(d) What was the purpose of this program?

(e) Can you follow along with what the code says, to match your assertions?

Don't stress if you can't answer all of these straight away. These are just some things you can start thinking about before next week.

7. Time to Submit

Once you have completed upto this point in the practical, you are required to submit your work to Blackboard.

We also need a copy of your **.bash_history** (all the commands you used in your practical). So before you tarball and submit, first close all terminals down, using <ctrl>-d. Then open a new terminal, navigate to your `P01` directory and run this command in your terminal

```
[user@pc]$ history > BashHistoryP01.txt
```

Now there will be a copy of your **.bash_history** stored within a file called **BashHistoryP01.txt** in your `P01` directory. Now its time to `zip` and upload! To create a gzipped tarball use the following command from your `PDI` folder:

```
[user@pc]$ tar -cvzf <studentID>_P01.tar.gz P01
```

You will have noticed all the files jump into an archive called `<studentID>_P01.tar.gz`

Now log into blackboard and in your **Programming Design and Implementation** unit, click on under "**Worksheet Submissions**" then "**Submission 1a: Intro to Linux, Vim and Java**". Browse your files till you get to `Documents/PDI/` and select the file we just archived (**.tar.gz**)

Click submit and you have successfully uploaded your in class portion of your worksheet!

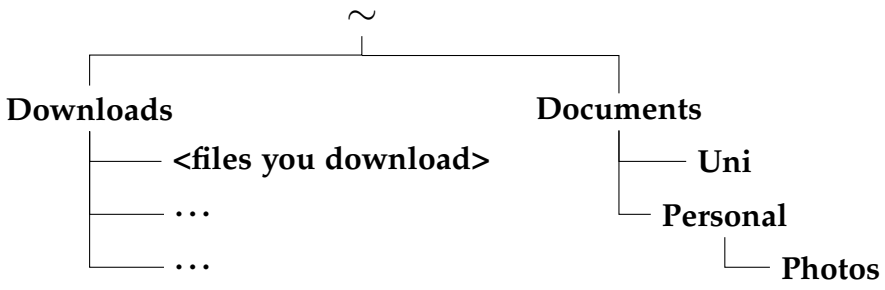
8. Assignment Questions

Hopefully after completing this prac you'll have a grasp on how to navigate a terminal and how to use vim. Your last task for this prac is to complete the following questions **at home and in your own time**.

> **Warning:** This question goes towards your portfolio/assignment mark and thus any collusion will be dealt with as per university policy.

Please create a file, with vim called "**`P01AssignmentQuestions.txt`**" inside your P01 directory and answer the following questions in it:

a) Using the directory structure provided below, if you were currently in the `Downloads` folder, what terminal command would you use to successfully navigate to the `Photos` folder?



(2 marks)

b) Using the same directory structure provided, if you were in the `Personal` folder, what command would you use to successfully navigate to the `Uni` folder?

(2 marks)

c) What terminal command would you use to create a `text` file in vim called `Shapes`?

(2 marks)

d) What terminal command would you use to create a `Java` file in vim called `Numbers`?

(2 marks)

e) What terminal command would you use to compile a Java file?

(2 marks)

f) What terminal command would you use to delete the directory called `Test`?

(2 marks)

Now submit this **`.txt`** file to Blackboard, under "**`Worksheet Submissions`**" then "**`Submission 1b: Assignment Questions`**"

And thats it! Your first PDI practical has finished. Make sure you spend your remaining weekly hours of the unit reading over the lecture slides to ensure you have a grasp of the concepts covered.