

# Worksheet 1: Intro to Unix and Number Systems

Updated: 27<sup>th</sup> February, 2021

The objectives of this practical are to:

- provide an introduction to Unix and Vim;
- practice issuing terminal commands;
- practice using the Vim editor;
- understand and execute number system conversions;
- understand and execute signed and real number conversions.

## 1. Logging on to Linux

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

Now, open a terminal window. Use the menus within the Linux environment to locate where it is. You can add it as a shortcut by dragging the terminal from the menu to the task bar. You will be using the terminal frequently in this unit, for both file navigation and manipulation, so making a shortcut to it will save you time throughout the semester. Alternatively, you can use the keyboard shortcut Ctrl + Alt + T.

## 2. Accessing resources

**Note:** The Java API documentation is an important resource, which will prove very useful later in the semester. It is worthwhile getting acquainted with it now.

The University lab machines have the Firefox web browser 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. They are:

- **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 be made on Blackboard.**
- **Java API Documentation** - <https://docs.oracle.com/javase/7/docs/api/>  
This is the documentation for Java, which 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 libraries as these will be utilised during the semester. At this point, do not worry about having a perfect understanding, but it is worth having a browse through.
- **StackOverflow** - <http://stackoverflow.com/>  
This is a community-driven website that provides answers for programming-related questions. If you have a question, it is likely that someone else has already asked it (and had it answered) on the site. **Please double check with a tutor the correctness of the answers**, as they 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 that it is a community-run resource. A good idea is to use Wikipedia as a starting point, finding a relevant page to your topic and then having a look at the references and following them. If there are no references, **don't trust it!**

### 3. Introduction to Linux and basic directory structure

Using **only** the terminal, it is now time to construct a directory structure that you can use to keep your work for this unit organised. A **sample** of the unix commands available to you have been provided below; there are many more commands and you may use them through out the unit.

The **<** and **>** braces are just a convention to show something that you fill in with the relevant value for you, while **↪** indicates the line continues below. Also, note that in Unix, folders are referred to as **directories**.

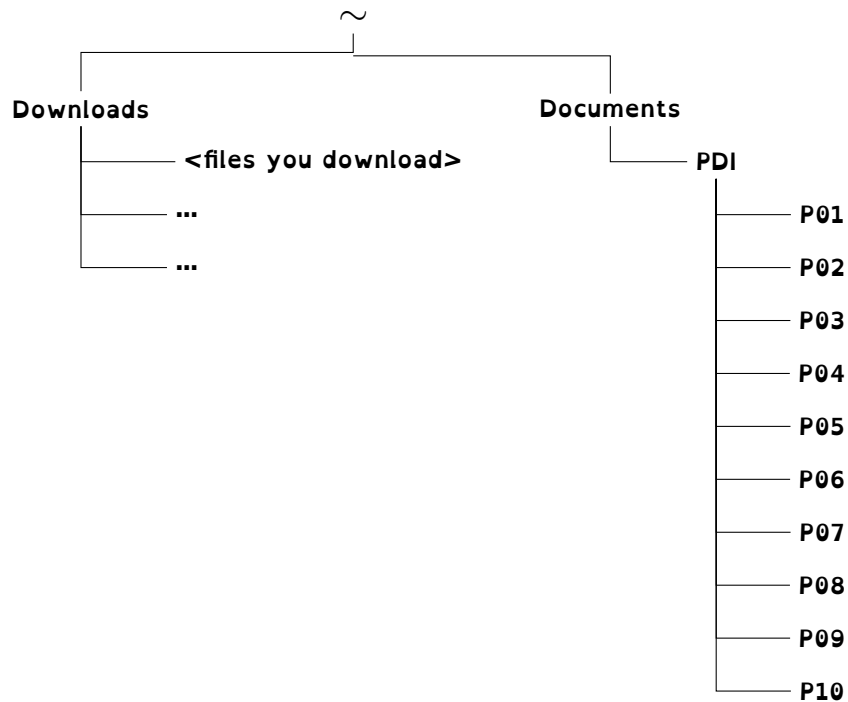
**Note:** The current directory is referenced by a single full stop (**.**) and the parent directory is referenced by two fullstops (**..**). All pathways are relative to the current directory. Your home directory can be referenced using the tilde (**~**).

Command	What it does	Example
<code>ls</code>	Lists all files in the current directory, if an argument is provided lists all files in the specified directory.	<code>ls</code> or <code>ls P01</code>
<code>cp &lt;source&gt; &lt;dest&gt;</code>	Copies the file from <b>source</b> to <b>dest</b> .	<code>cp Downloads/Test.java ↵ Documents/</code>
<code>mv &lt;source&gt; &lt;dest&gt;</code>	Moves the file from <b>source</b> to <b>dest</b> . If the <b>dest</b> ends in a file name it will rename the file.	<code>mv ../Downloads/TFA.avi ↵ Videos/Awesome</code> or <code>mv Test.java SimpleTest.java</code>
<code>pwd</code>	Lists the directory you are currently in ( <b>Print Working Directory</b> ).	<code>pwd</code>
<code>cd &lt;directory&gt;</code>	Moves to <b>directory</b> (this can be a path specification). <b>Change Directory</b>	<code>cd PDI</code> or <code>cd Documents/PDI/P01</code>
<code>mkdir &lt;directoryName&gt;</code>	Creates a new directory named <b>directoryName</b> in the current directory.	<code>mkdir PDI</code>
<code>rm &lt;fileName&gt;</code>	Removes a file.	<code>rm SimpleTest.java</code>
<code>rm -rf &lt;folderName&gt;</code>	Removes a <b>directory</b> . Be <b>very</b> careful with this command as you are forcibly removing something and you can not get it back.	<code>rm -rf PDI</code>

The best way to learn these commands is through practice – that is, regularly using the terminal. Your tutor will go through these commands with you in-depth before you continue on with the next step.

Using the above commands, create the following directory structure. The symbol ~ refers to your personal user area. An 'Introduction to Linux' has been uploaded to Blackboard; it will be helpful if you get stuck.

**Note:** You will be using the terminal for a lot of your work in this unit, so it is good practice to start now. There are situations where you won't be able to use a graphical interface, so while you have the opportunity, it is strongly suggested to practice using the terminal! You may also be assessed on the commands, so it is a good idea to practice them to cement your knowledge.



#### 4. Setting up Vim

For this unit, at this time, you are required to use **vim** as your (terminal) text editor, as per the Unit Outline. Vim is a highly customisable, terminal-based text editor.

Vim has two modes, **command** (a.k.a 'normal') and **edit**. When in **command** mode, you can use a variety of editor commands to move around and manipulate the text. You can enter this mode by pressing the Esc key, and it is the default mode that the program opens in. In order to add or edit text, you will need to enter **edit** mode by pressing the A or I keys.

Here are some of the commands you will need to use **vim** in this (and other) practicals.

Command	Meaning
<code>Esc</code>	Enter <b>command</b> mode.
<code>A</code> or <code>I</code>	Enter <b>edit</b> mode.
<code>G</code>	Moves the cursor to the last line of the document.
<code>G</code> <code>G</code>	Moves the cursor to the first line of the document.
<code>:&lt;num&gt;</code>	Moves the cursor to the line number <b>num</b> . Press <code>Enter</code> to execute.
<code>Y</code> <code>Y</code>	Copies the current line.
<code>D</code> <code>D</code>	Cuts the current line.
<code>P</code>	Pastes the line.
<code>U</code>	Undo.
<code>Ctrl</code> + <code>R</code>	Redo.
<code>/&lt;search&gt;</code>	Highlights everything matching the <b>search</b> term. Use <code>N</code> to move to the next item and <code>Ctrl</code> + <code>N</code> to move to the previous item. Press <code>Enter</code> to execute.
<code>:w</code>	Saves the document. Press <code>Enter</code> to execute.
<code>:q</code>	Quits. Use <code>:wq</code> to save and quit at the same time or <code>:q!</code> to force quit (when you don't want to save). Press <code>Enter</code> to execute.

**Note:** There is a difference between the two **edit** modes: **insert** mode `I` and **append** mode `A`. `I` will allow you to type before the cursor (as the cursor itself takes up 1 character), whereas `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 improve 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 will need to open the `.vimrc` file using Vim by issuing the command:

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

Enter **edit** mode by pressing `I`, then add the following text to the file:

```
syntax on
set tabstop=4
set softtabstop=4
set expandtab
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 just yet!**

The last step is to **source** the file in order to ensure that there are no errors. To do this, press `Esc` to enter **command** mode and then type `:source .vimrc` and press `Enter`.

If no errors occur, you will see line numbers appear in your terminal. Then, you can issue the command `:q` to exit Vim and return to the terminal.

**Warning:** If any error messages appear, ask your tutor for help as something has gone wrong! Do not be worried if this occurs, but do not proceed until the errors have been fixed.

You may also have multiple terminals open - for example, 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.

Do not try and edit the same file in multiple terminals – otherwise it is very likely you will overwrite one version of the file with another!

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

## 5. Number system conversions

The following questions should be attempted either on paper or within a word processor-style document (or even a text document):

- (a) Convert  $42_{10}$  from base 10 to binary;
- (b) Convert  $10101_2$  from binary to base 10;
- (c) Convert  $42_{10}$  from base 10 to octal;
- (d) Convert  $2B_{16}$  from hexadecimal to base 10;

## 6. Signed and real number conversions

The following questions should be attempted either on paper or within a word processor-style document (or even a text document):

- (a) Convert  $42_{10}$  from base 10 to negative eight-bit binary two's complement, i.e. the value of  $-42_{10}$ ;
- (b) Convert  $42.75_{10}$  from base 10 to a IEEE 754 binary32 number.

You may wish to confirm your answers for some of these questions by undertaking the 'reverse' conversion.

That's it! You've now finished your first PDI practical. Make sure you spend some time reading over the lecture slides to ensure you have a grasp of the concepts covered.

End of Worksheet