



# Operating System Basics for Linux

*Department of Engineering Science, University of Oxford*

*Izzi Mear & Eric Peasley, Version 1.0 July 21*

An operating system is a large, complex collection of software that manages the computer. All communication between the central processor and the keyboard, mouse, screen, computer disk and network is controlled by the operating system. It is the operating system that detects the click of a button or the movement of the mouse. Other programs cannot communicate with these peripheral devices directly. They have to ask the operating system to carry out these tasks.

The Linux operating system is based on the much older Unix operating system, first developed in 1969 at Bell Labs in New Jersey.

This document introduces you to the basics such as accessing free applications, managing your files and using the terminal. There are many Linux distributions, and this guide does not cover all case. The examples will usually be from Ubuntu.

## Contents

- Useful Applications (e.g. Free access to Office365) Page 2
- Managing Applications (e.g. Split Screen and Virtual Desktops) Page 5
- Managing your Files (e.g. Organising files/folders and backups) Page 7
- Using a Terminal (e.g. Text-based commands to create folders) Page 13

Comments, corrections and suggestions to [izzi.mear@eng.ox.ac.uk](mailto:izzi.mear@eng.ox.ac.uk)

Icons throughout sourced from [FlatIcon.com](http://FlatIcon.com), made by [Freepik](https://www.freepik.com), Eucalyp, smashicons, wanicon, icongeek26 and tHose icons.



## Useful Applications

Your operating system will come bundled with some applications. Departmental computers will already have all the software you need, however if you are working on a personal computer there may be some software you need to install.

### Office 365

Students and staff at the University can access Office 365, which contains many useful applications for University work. The University of Oxford Office 365 system includes a cloud based working environment (called Nexus365). Follow the instructions here to get started: <https://help.it.ox.ac.uk/nexus365/getting-started>

Here you will find :-



**Outlook:** The Microsoft email client to view your University emails.



**One Drive:** A cloud based filing system. You can use this to backup your files and share files with other people.



**Word:** The Microsoft word processor. You can save files as a Word document(.docx) to **One Drive** or your own computer.



**Excel:** The Microsoft spreadsheet. You can save files as an Excel spreadsheet(.xlsx) to **One Drive** or your own computer.



**One Note:** A digital notebook, you can use this for [taking notes](#).



**Powerpoint:** Used for creating presentations.

Each of these can be used online, in your browser. This is the most reliable option as there is no official installation from Microsoft for Linux systems. Some browsers work better than others, so having multiple installed (e.g. [Chrome](#) & [Firefox](#)) can be useful.

You can only get full functionality by installing the software directly onto your machine. There are options for getting a full installation working on Linux using Virtual Machines. See for example the instructions on [makeuseof.com](http://makeuseof.com).



## Microsoft Teams App

Another part of Nexus365 is the Application **Teams**. It is an online chat/conferencing tool. Teams lets you message, talk to and see other members of the University. It will be used for the online laboratories. There is a web based version.

### Logging into the Teams App

Log in using your SSO and password:

Sign in using **abc123@OX.AC.UK**

Replacing **abc123** with your own Single Sign On (SSO) username.

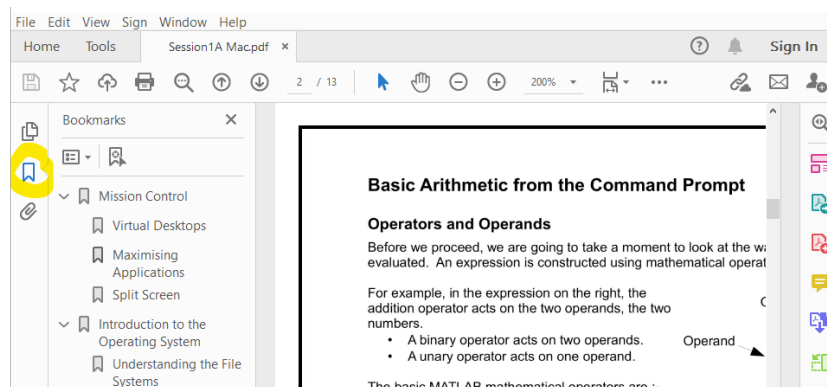
Sometimes it capitalizes the first letter of the SSO, if so change it back to lower case.



## PDF Reader

You will need to use PDFs. Whichever [PDF Reader](#) you use, make sure you can:

- **Make notes and highlight**
- **Search within PDFs:** You will often work with many documents, searching for key terms using the Search bar is much faster than scrolling.
- **Viewing Headings for Navigation:** You can save a lot of time by enabling the navigation pane to click through Headings, shown here circled in yellow:





## Creating Text Files

You will not always want files where the text is formatted. In some cases you will want something very simple: just words as a .txt file. This could be because of:

- personal preference (some people prefer the minimalist environment)
- the wider range of compatibility (all OS will be able to read a .txt even if Word is not installed).

Most Linux distributions should have a **Text Editor** installed. For some distributions, you can create a **New Document** by right clicking on the background in the **File Browser**. Then double click on the Document and it will open in a Text Editor.

There are two different types of files it can save:

- **Plain Text** format just contains ASCII character codes. You can't change the font. These save to **.txt** files.
- **Rich Text** format contains extra formatting codes for changing the font and size of the text and other formatting commands. These save to **.rtf** files.

### Try the following :-

Select **New Document**

In the **Format** menu select **Make Plain Text**

Enter the following text into the document :-

**This is an example text file.**

Save as **ReadMe.txt**. You can move this file later.



## Security

Keeping your computer, files and identity safe is hugely important. Sophos Anti-Virus is available for free from the Central IT pages, using the above link. You should also consider [creating strong passwords](#) and using a password manager. Much can be found online, for example from [Reporters without Borders](#).

## Managing Applications

Often you will need to work with multiple applications (apps) open. For example, if you are coding you may need a PDF of instructions open as well as MATLAB. Or if you are writing a report you may need an internet browser and a Word document. This section introduces you to different ways to work with multiple apps.

### Minimising, Closing and Quitting

To get the best from your computer, it is important to only have the applications open that you need for your current task. Having too many applications open at once will cause your machine to be slow. You might think you are quitting applications by minimising and closing:

- **Minimise** an app by pressing the - button in the top-right corner of the window
- **Close** windows of an app by pressing the X button in the top-right corner

However, **minimising and even closing windows does not always quit the app.** The app can still be using memory, which will slow down your computer.



Quit rather than close/minimise any applications you are not using. This will free up RAM (a type of computer memory).

### Using Systems Monitor

To check which applications are using memory, use the **Systems Monitor**.

- Open **Applications** then scroll to **Utilities**
- Select **Systems Monitor**

Under the Processes tab you will see a list of all programs running. To stop an unresponsive program, simply select it and right-click. You can then select:

- **Stop** to pause the process
- **End** which will fully close an application safely.
- **Kill** which forces the app to quit, only use if End Process does not work.

For more information, see this [article](#) on how Linux uses memory.



## Split Screen View

To work with two applications at once, a simple option is to split the screen between two applications. To do this:

- Open two applications on a Desktop.
- Move your cursor to the edge of a window, and when an arrow appears resize the window by holding and dragging.

You can drag the divide between the two applications to the left and right.



## Maximising Applications

The approach of splitting the screen is not always the best option – particularly if you have a small monitor. Sometimes we need to work with an application which has been maximised to fill the screen. Maximise an application by pressing the square in the top-right corner of an app window.

# Managing your Files

## Files and Folders



A **file** is just a lot of ones and zeros stored in a permanent location on the computer. Each file is used to store a particular type of object.

It could contain a picture, a document, a program or many other types of object.

Each file has a name, which may also include a suffix (ending • something) which helps to show what the file contains. For example:

File name with suffix	Type of File
Lecturenotes.pdf	A document
ReadMe.txt	Plain Text
photo.jpg	A Picture
MyScript.m	A MATLAB Program

There are thousands of files on a typical computer. A **file system** is used to keep track of all the files. The file system is divided into folders.



A **folder**, also known as a **directory**, is a location where all the files related to a particular topic are stored. Folders help keep related files together, so they do not get mixed up with other files on the system.

Files and folders may seem basic, but keeping them organised can be a difficult habit to develop. It is always worth taking a few moments to correctly store your files.



When you download a file, it is usually stored automatically in your **Downloads** folder. If you want to store it long term, **move this file** so it is **stored in a sensible folder elsewhere** in your file system.



As well as storing files on your computer **locally**, it is sensible to **store a copy of your files online too**. e.g. [using OneDrive](#)



## A File System

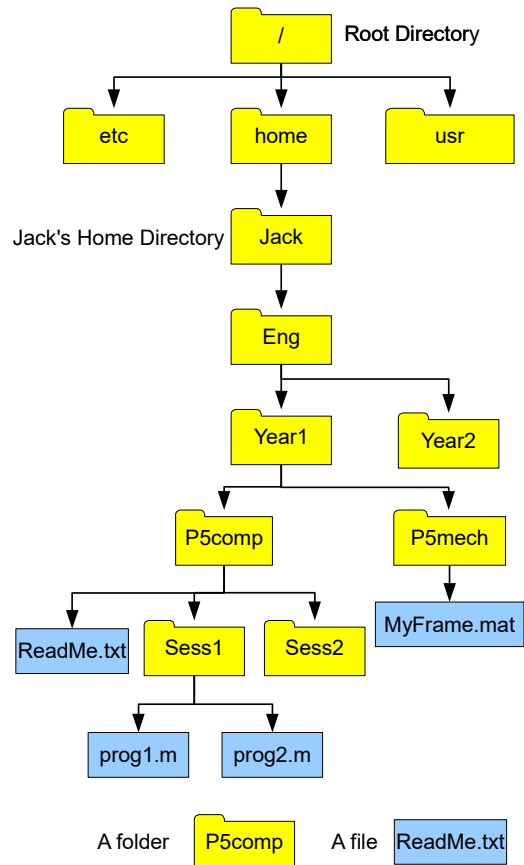
The whole file system is organised in a hierarchy of folders within folders. The diagram on the right represents a very small portion of a Linux file system. Another name for a folder is a **directory**.

At the top of the file system is the folder known as the **root directory**. It is specified with a forward slash “/”. The root directory contains many other directories than shown here.

The diagram looks a bit like a family tree. The words **parent** and **child** are used in computing to describe the relationship between directories.

In the diagram:

- The **parent** of **Sess1** is **P5comp**.
- **Sess1** and **Sess2** are the **children** of **P5comp**.



## Home Directory

When using a computer, you don't want to mix up your files with those used by the operating system and other users. Therefore, each user on a system is allocated a **Home Directory**, where they can store their personal files. On most Linux systems, these are all kept in the **home** folder. An example of **creating folders** is given on the next page. Follow these to create the structure shown in the diagram above.



Make sure that your **home** is always organised.  
This will make it easier to find files, and lead to efficient working.

For a more in-depth guide to the file system on Ubuntu, see [this video](#).

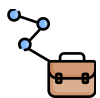


## Example: Creating Folders in your Home Folder

Some Linux systems have a choice of different GUI. There are also many ways of configuring a GUI. You will need to use a **File Browser**. The following instructions show to how build the example file structure on the previous page:

- Open a **File Browser**, go to your **home directory**
- Create a **New Folder** (right-clicking on the background in the File Browser).
- Rename the folder **Eng** (right-clicking the folder should give you this option).
- Go to the **Eng** folder (single or double clicking on the folder)
- Inside the new folder, create another new folder called **Year1**
- Inside **Year1**, create another folder called **P5comp**.

The idea is that as you do more work on your computer, you slowly build a hierarchy structure of folders, so that you can easily find the files that you need. e.g. If you create files for the mechanics laboratory, put them into a **P5mech** folder in **Year1**.



## File Paths

The **file path** is a way to describe the location of a file or folder, based on a fixed starting point. Every file/folder will have a unique **absolute path**.



The **Absolute Path of a file/folder** (e.g. fileA) is the path you would have to take from the **top of the system** (root directory) to reach **that file/folder**.

## Example: Absolute Path of ReadMe.txt

Consider the file **ReadMe.txt** on the previous page. To reach this file starting at the root directory, you would go into **home**, then the specific users home directory (Jack), then **Eng**, **Year1**, **P5comp** until you arrive at **ReadMe.txt**. The Absolute path is written **/home/Jack/Eng/Year1/P5comp/ReadMe.txt**

The absolute path of a file can be very long. A quicker way to specify the path from the **Current Working Directory** to the file or folder you are interested in. This is called the **relative path** to a file. This describes a files location **relative** to the **Current Working Directory**.



The **Relative Path of a file/folder** is the path you would have to take from the **current Working Directory** to the **original file/folder**.

### Example: Relative Path moving Down

Considering our example file structure, if the **Current Working Directory** is **/home/Jack/Eng/Year1/P5comp**

- The Relative Path to **ReadMe.txt** is just **ReadMe.txt**
- The Relative Path to **Prog1.m** is **Sess1/prog1.m**

A relative path does NOT start with a forward slash.

### Example: Relative Path moving Up

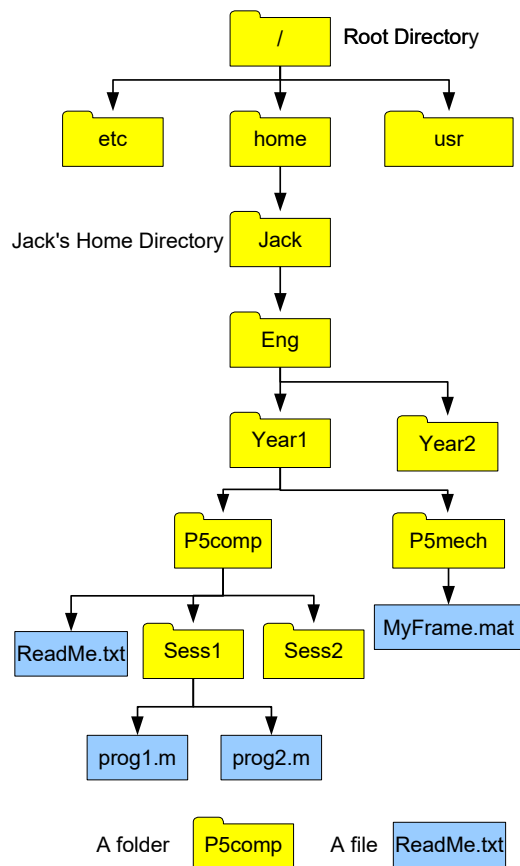
You can also have a relative path that goes up the tree as well as down. Two full stops in a path indicates the parent directory from wherever you are in the path.

If the Current Working Directory is folder

#### **Sess2**

- The Relative Path to **prog1.m** is **../Sess1/prog1.m**
- The Relative Path to **MyFrame.mat** is **../../P5mech/ MyFrame.mat**

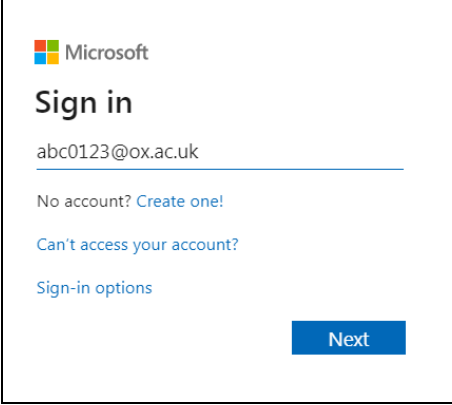
Relative paths take some getting used to, but can be a powerful tool when managing many files.



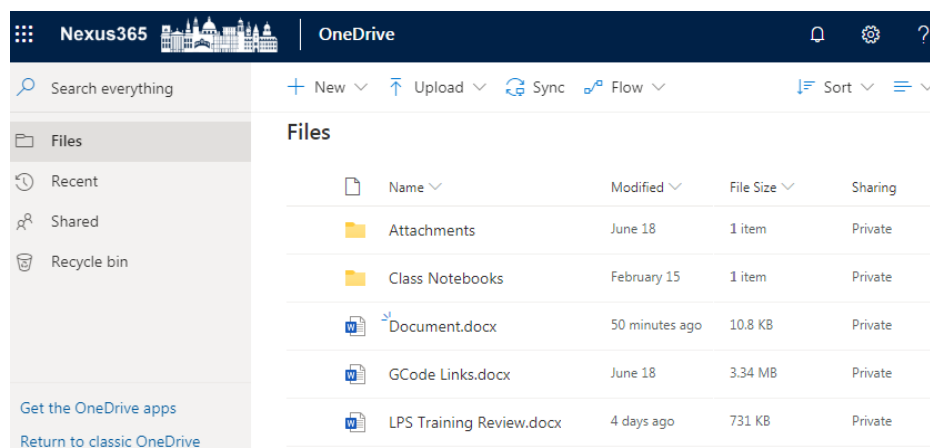
## Backup using OneDrive

It is always a good idea to take a copy of your files, just in case anything happens to your computer. As part of your University Single Sign On (SSO) account, you have access to **One Drive**. This is a file storage area linked to the Microsoft Office Suite available online.

- Visit [Office.com](https://office.com) and click **Sign In**.
- Use your SSO username with ending **@ox.ac.uk**
- At the **Shibboleth SSO Sign In** page, use your SSO username and password.



Once you logged in, select **OneDrive** on your **dashboard** to open your file storage:



- Use the **Upload** button to add files or folders to your storage area. You can also drag and drop. Try uploading a Folder.
- In OneDrive, Right Click any file and select **Download** to download a file.



If you want to avoid manually **uploading a back up to OneDrive**, you can set up a folder on your computer to [sync using OneDrive](#).



## Using Zip Folders

An easy way to transfer **multiple files** or folders between computers is to use a **zip file**. This stores all the data into a single file. It also uses compression software to reduce the size of the data.

### Zipping Items into a File

- Open the file or folder in File Browser
- Control-click it or tap it using two fingers
- Select Compress from the menu.

You will now see a .zip file is created.

### Extracting a Zip File

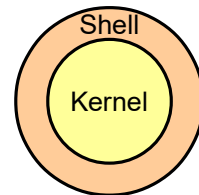
Double clicking on a zip file will extract the contents.

## Using a Terminal

So far we have shown the the easiest way of interacting with the operating system: using a **Graphical User Interface** or **GUI**. This is where you interact with the computer using a mouse pointer and click to interact with applications.

Operating systems (OS) are designed like a nut:

- The main part of the OS is called the **Kernel**. This contains all the **programs to manage the computer** and hardware.
- Surrounding the kernel is the **Shell**. The shell contains the **software** that allows you to interact with the kernel.



The GUI that you use to interact with your computer is a part of the Shell. However, this is not the only way of interacting with the Kernel. There are also text based commands that allow you to control the OS without a mouse. This is known as using the **Command Line**.



To use the command line, we use a **Terminal** environment.

It is a simple window where we can type text.

### Why use the Terminal?

Imagine Errol the Engineer has just returned from a six-month survey of materials in different off-shore structures. The 1430 samples have been run through a testing machine, resulting in 1430 data files. The research group has worked with files from this machine before, and know before they can graph the results the files need to be put through a program called **reformatData.sh**. In the past, they only had 20 files, so this script was run 20 times by hand. If Errol has to run **reformatData.sh** by hand using a GUI, they will have to **select and open a file 1430 times!** If it takes even 30 seconds to run each file, the whole process will take more than 12 hours. With the shell, Errol can instead assign the computer this mundane task!

It is important to **know the shell exists and what it can do**. The next few pages give a brief introduction. For a full introduction to shell, see this [online course](#).

## Navigating Files using the Terminal

You cannot do anything in a terminal without knowing some commands. Different terminals will expect different commands. Most Linux system, when you open the Terminal it expects **bash** (for Bourne Again Shell) commands. This section will introduce some key commands for looking at your files in the Terminal, like you would using **File Browser**. The basic tasks covered will be: Launching the Terminal, Checking the starting location, moving to a subfolder, listing the folders contents and then creating a folder.



### 1. Launch Terminal

This can be found in Applications.



### 2. Check Starting Location

When using the terminal, you have a **Current Working Directory**. That is your current location within the file system. This is like when you open a File Browser and have a particular Folder open. You can also think of it as looking at one folder.

In the terminal, enter the command :

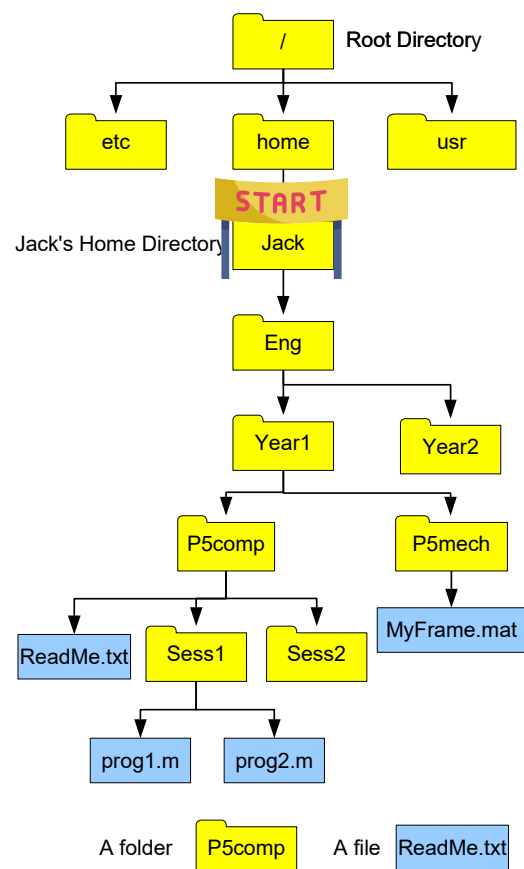
```
pwd
```

This **P**rints your **C**urrent **W**orking **D**irectory, your current location in the file system. To start with, the Current Working Directory is usually set to your Home directory.

Notice that when you press Enter, you will see a path returned.

In our example it would be:

**/home/Jack**





### 3. List items in Current Folder

To list the content of the Current Working Directory enter

```
ls This is a lower case L followed by a lower case s
```

You should see a **list** of all the files and folders in your Home directory.

In our example, it would simply list one folder **Eng**.



### 4. Move into a Subfolder

To **move around** to look at other files/folders, use the command **cd**. It changes the current working **directory**. Here we move to **subfolder** called **Eng** then list contents.

```
cd Eng
```

```
pwd
```

```
ls
```

In our example, we would see the current path printed (due to the **pwd** command):

```
/home/Jack/Eng
```

Then a list of folders Year1 and Year2.



### 5. Create a new folder

To **create a new folder**, use the command **mkdir**. This stands for **make directory**.

```
mkdir Year3
```

```
ls
```

You should then see the new folder listed after the list command.

### Further Examples using List

We can use the **ls** command to list the contents of any folder on the file system. We can specify the folder we want by using the folder's absolute path :

```
ls /
```

```
ls /usr
```

```
ls /home
```

or the relative path of the folder:

```
ls Year1
ls Year1/P5computing
```

You can use the tilde ~ to indicate your Home directory:

```
ls ~
ls ~/Downloads
```

Two full stops .. indicates the parent folder. The folder one level above :

```
ls ..
ls ../..
```

### Further Examples using cd

You can also use .. with the **cd** command:

```
cd Year1/P5computing
pwd
cd ..
pwd
cd /usr
pwd
```

The **cd** command on its own take you back to your **home**:

```
cd
pwd
```

Now make another new directory inside **P5Computing** using the command **mkdir**.

```
cd ~/Year1/P5computing
mkdir NewFolder
```

There are many more commands that you can use.

For a list of OS commands, see <http://ss64.com/bash>

For a comprehensive introduction to shell, see this [self-paced online course](#).