A LEVEL
# COMPUTER SCIENCE
## Recipe Card

# OPERATING SYSTEMS

### Introduction

In this activity you will learn about the key functions and features of operating systems that can be used with the Raspberry Pi computer.

The operating system or systems software, is a vital component that performs key functions:

- managing system resources, eg memory and storage
- CPU scheduling
- generation of the user interface (GUI or shell)
- system protection
- supporting network communication

The Raspberry Pi may have one of several operating systems installed but they will all be based on the Linux kernel which is an **open source** platform. In this activity, we will take a closer look at Linux and its operation. For illustrative purposes, the distribution Raspbian is used.

This page and video from the **Raspberry Pi Foundation's Carrie-Anne Philbin** explains how to set up the operating system:

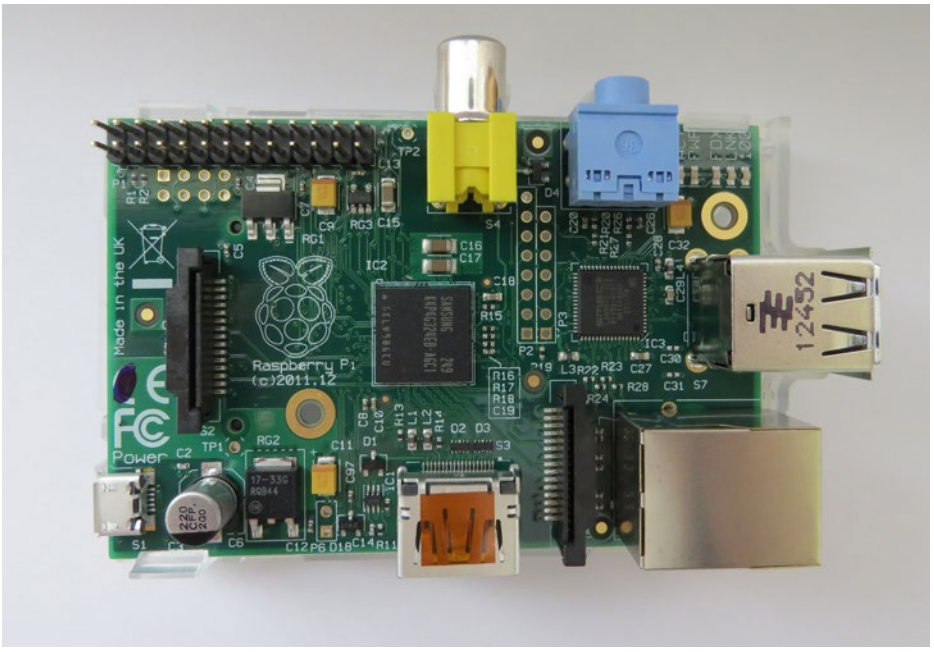
http://www.raspberrypi.org/introducing-the-new-out-of-box-software-noobs/

**Logging in**

The default root login is:

Username: **pi**

Password: **raspberry**




## Interacting with the shell

By logging in using the root username and password, you are the assumed system **administrator or super user**. This will give you total power including the ability to create accounts for new users and groups, and to delete users, groups and any files and folders that you choose. To accomplish certain actions by issuing commands in the terminal, they need to be preceded by **sudo** which stands for "**superuser do**". If you create a standard **user** account, their activities will be limited as only a subset of permissions are applied.

## Useful sudo commands

| Command | Purpose |
|---|---|
| **sudo apt-get update** | Checks for package updates (assumes network/internet connection functional) |
| **sudo apt-get install *package name*** | Installs software package |
| **sudo poweroff** | Gracefully shuts down the Raspberry Pi |
| **sudo reboot** | Gracefully restarts the Pi |
| **sudo addgroup *groupname*** | Creates a new user group using default user permissions |
| **sudo adduser *username*** | Creates a new user using default user permissions. You can also add this user to your newly created group, eg<br>Sudo addgroup *finance*<br>Sudo adduser *fred_smith finance* |

## Other useful shell commands

| Command | Purpose |
|---|---|
| **whoami** | Displays username of currently active user |
| **pwd** | Displays current working directory |
| **ls** | Lists files and sub directories in the current directory |
| **mkdir *myfolder*** | Creates a new sub directory called *myfolder* in the current directory |
| **cd myfolder** | Changes current working directory to *myfolder* |
| **chmod *permissions filename*** | Change default permissions for a specified file<br>See http://linuxcommand.org/lts0070.php for more details |
| **man *commandname*** | Provides help on a particular command *commandname* |
| **cp *myfile yourfile*** | Copies the existing file called myfile to a new file called *yourfile* in the same directory |
| **rm *myfile*** | Deletes the file myfile |
| **rmdir *myfolder*** | Deletes the directory *myfolder* |
| **sudo nano */myfolder/ mysubfolder/myfile*** | Open the nano text editor to edit the file *myfile*. Note: file path must be given |
| **kill *process_id*** | Stops a particular process |
| **python myprogram.py &** | Runs myprogram.py as a background process |
| **fg python myprogram.py** | Runs myprogram.py as a foreground process |

## Activities

1. Whilst logged in as the root user, create a **group** called ***students*** and a new user account for yourself with a suitable **username and password**. Add yourself as a member of the ***students*** group.

```
raspberrypi login: pi
Password: _
```

*Fig 1. Root login*

```
pi@raspberrypi ~$ sudo addgroup students
Adding group 'students' (GID 1002)...
Done.
pi@raspberrypi ~ $ sudo adduser jbloggs students
adduser: The user 'jbloggs'does not exist.
pi@raspberrypi ~ $ sudo adduser jbloggs
Adding user 'jbloggs' ...
Adding new group 'jbloggs' (1003) ...
Adding new user 'jbloggs' (100) with group 'jbloggs'...
Creating home directory '/home/jbloggs' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changin gthe user information for jbloggs
Enter the new value, or press ENTER for the defaults
    Full Name []: Joe Bloggs
    Room Number []: 1
    Work Phone []: 12345 678901
    Home Phone []: 98765 432109
    Other []:
Is the information correct? [Y/n] y
```

```
pi@raspberrypi ~ $ sudo fsmith
Adding user 'fsmith' ...
Adding new group 'fsmith' (1003) ...
Adding new user 'fsmith' (100) with group 'fsmith'...
Creating home directory '/home/fsmith' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for fsmith
Enter the new value, or press ENTER for the defaults
    Full Name []: Fred Smith
    Room Number []: 1
    Work Phone []: 12345 678901
    Home Phone []: 98765 432109
    Other []:
Is the information correct? [Y/n] y
pi@raspberrypi ~$ sudo adduser fsmith students
Adding user 'fsmith' to group 'students' ...
Adding user 'fsmith' to group 'students' ...
Done.
pi@raspberrypi: ~ $ _
```

*Fig 2. Creating a group and users. Adding users as members of the group.*

2. Create new **directories** for each subject that you are currently studying. Create a **text file** within one folder. You can do this within the GUI desktop environment by typing in **startx** at the terminal prompt.

```
jbloggs@raspberrypi ~ $ mkdir biology
jbloggs@raspberrypi ~ $ cd biology
jbloggs@raspberrypi ~ /biology $
```

*Fig 3. Creating new sub directories*

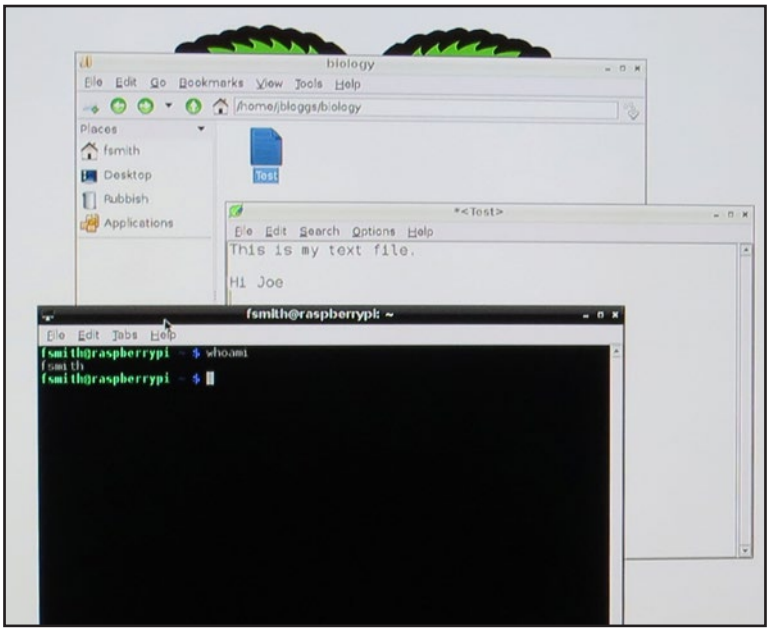



*Fig 4. Creating a text file and using "whoami" to show who is currently logged in.*

3. Log out of the root account and login to your new user account.

4. Access the text file that you created in step 2.

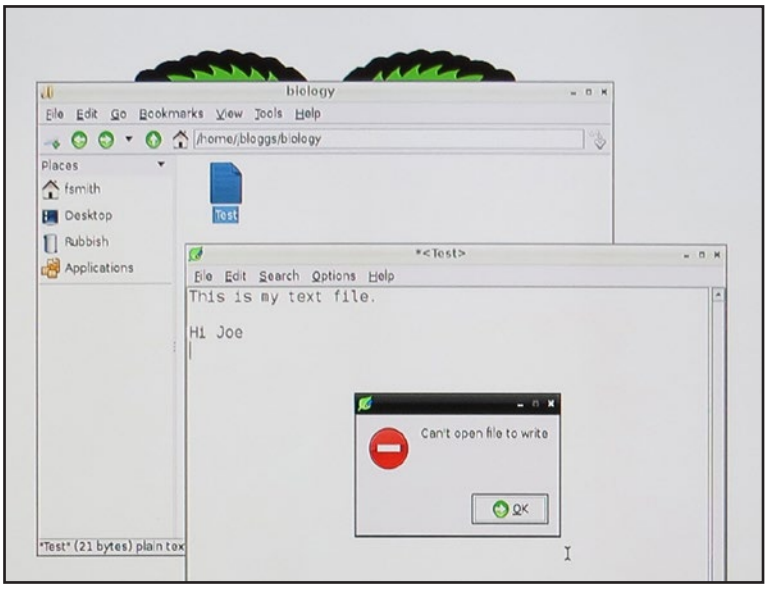



*Fig 5. When logged in as a different non-root user, the file cannot be edited as read only permissions apply by default.*

5. Make a change to the file and try to save the changes (do not select **save as**).

6. You should find that you will be unable to save the changes as you have **read only** permissions. Files that you create, you have ownership of and will be able to modify. However, files that you did not create, you will have read only permission by default. It is possible to modify file permissions using the **chmod** command. The **permissions** parameter is calculated by adding together the **octal values.**

The syntax requires us to give three numbers in sequence representing:

- file owner
- group to which they belong
- followed by the permission for everybody else

Owner permission: rwx =4+2+1 = 7

Group permission: rx = 4+1 = 5

Others: x = 1

**chmod 751 myfile**

| File permission | | Octal value |
|---|---|---|
| Read | r | 4 |
| Write | w | 2 |
| Execute | x | 1 |

```
jbloggs@raspberrypi ~ $ cd biology
jbloggs@raspberrypi ~ /biology $ chmod 777 Test
jbloggs@raspberrypi ~/biology $ _
```

*Fig 6. Creator of file uses chmod to give all users full access.*

7. Logout and log back in as the root user. Change the effective permissions of the file that you created in step 2 so that everybody (this includes "others") has full access. Remember that read precedes write so you must be able to read a file as well as being able to write to it. Log back in as your own account and repeat step 5 again. You should now be able to make changes to the text file and save them using the same file name.

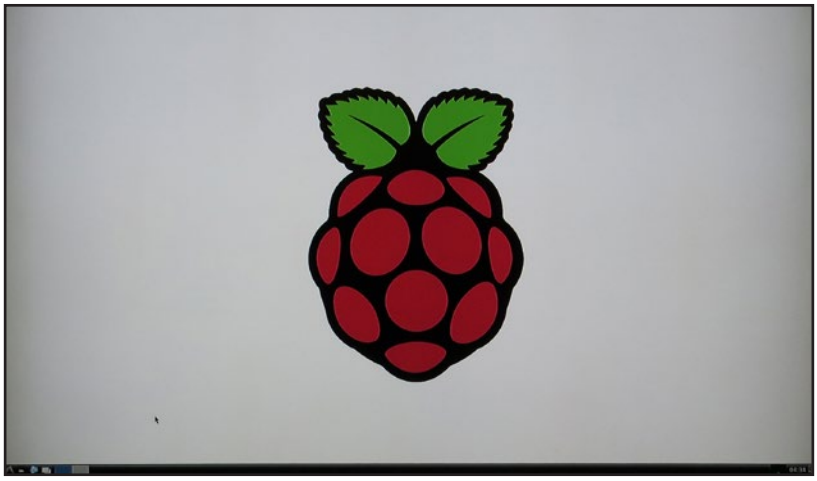


*Fig 7. Desktop GUI logged in as a non-root user.*

```
jbloggs@raspberrypi ~ $ man ls
```

```
LS(1)                    User Commands                    LS(1)

NAME
       ls  - list directory contents

SYNOPSIS
       ls [OPTION] ... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       - a,  - - all
              do not ignore entries starting with  .
       - A,  - - almost - all
              do not list implied  .  and . .

       - - author
              with - l, print the author of each file

       - b, - - escape
              print C - style escapes for nongraphic characters

       - -  block - size = SIZE
              scale sizes by SIZE before printing them. E.g., `- - block -
              size=M' prints sizes in units of 1,048,575 bytes. See size format
              below.

       -B, - - ignore-backups
              do not list implied entries ending with ~

       - c  with -lt: sort by, and show, ctime (time of last modification of
                   file status information)with -l: show ctime and sort by name
                   otherwise: sort by ctime, newest first.

       - C  list entries by columns

       - - color [=WHEN]
              colorize the output. WHEN defaults to `always' or can be `never' or
              `auto'. More info                    below

       - d,  - - directory
              list directory entries instead of contents, and do not dereference
              symbolic links

       - D, - - dired
              generate output designed for EMacs' dired mode

       - f  do not sort, enable -aU, disable -ls --color
```

*Fig 8. Using the command "man" to find help.*

## Further activities

8. Access the GUI by issuing the command **startx** in the console. Investigate the **system utilities** provided and **application software**. Investigate other software available for the Raspberry Pi.

9. The Raspbian OS supports **multitasking, multithreading and multiuser** access. Investigate features to support this.
10. Raspbian uses the **Completely Fair Scheduler** for CPU scheduling. This divides processor power between running processes in an attempt to give each the same amount. Investigate this algorithm and how it compares to more simple ones eg round robin scheduling.

## Useful resources

The Raspberry Pi.org website - http://www.raspberrypi.org/

The Raspbian project website - http://www.raspbian.org/FrontPage

Terminal commands - http://www.penguintutor.com/raspberrypi/useful-command-reference

Permissions reference - http://linuxcommand.org/lts0070.php

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest