# Raspberry Pi Linux

Written by Mark Webster and ??

## Introduction

A Raspberry Pi single board computer can run many different operating systems such as Raspbian, Raspberry Pi Media Center, FreeBSD, Window 10 IoT core, RetroPi, RISC OS, RTAndroid, OSMC, Tiny Core, etc.

The default and most widely supported operating system is called Raspbian, which is a variant of Linux, which in turn is a variant of Unix. The operating system Unix is one of the oldest systems still in use and has evolved tremendous capabilities in the decades since it was created in 1971. Half the servers in the world run Linux or some variant of Unix.

Windows users may have heard of the original operating system from Microsoft called MS-DOS. That was also a command line or terminal based operating system.

This workshop is based on the default Raspbian operating system.

## Structure of Linux

Linux is designed to be open and completely customizable. Each layer of the operating system can be inspected and replaced. This means there have been many different versions of linux created that use the same linux kernel but look very different because of various versions of the Desktop layer. Linux users become almost religious about their favorite desktop (Gnome, KDE, XFCE, etc)

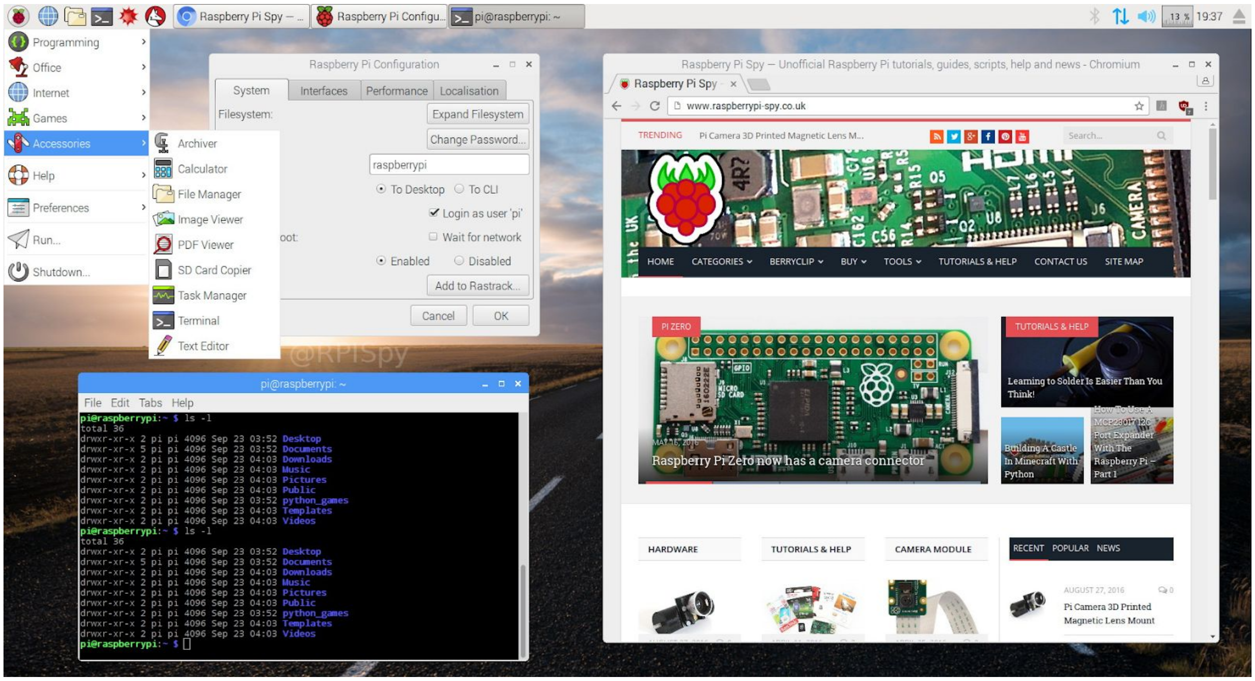
| ***General Linux Layer*** | ***Raspbian*** |
|---|---|
| Boot loader | On chip ROM. bootcode.bin |
| Kernel | Kernel (current release 4.14.79) |
| Daemons | Managed by systemd. Daemons are device drivers and processes that connect to the network, USB, etc |
| Shell | Bash (scripting and commands) |
| X Window Server | X-window server, xorg (allows graphical GUI) |

|  |  |
|---|---|
| Window Manager | Openbox |
| Desktop | PIXEL (modified LDXE desktop) |

Linux command line software guiding principles:

- Everything is a file. ( Including hardware, such as serial ports, USB drives, etc)
- Small, single-purpose programs.
- Ability to chain programs together to perform complex tasks.
- Avoid captive user interfaces. (programs don’t require back and forth interaction)
- Configuration data stored in text.

# PIXEL Desktop



Filesystem: Expand Filesystem

Change Password...

raspberrypi

● To Desktop ○ To CLI

☑ Login as user 'pi'

□ Wait for network

● Enabled ○ Disabled

Add to Rastrack...

```
pi@raspberrypi:~ $ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 Sep 23 03:52 Desktop
drwxr-xr-x 5 pi pi 4096 Sep 23 03:52 Documents
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Downloads
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Music
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Pictures
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Public
drwxr-xr-x 2 pi pi 4096 Sep 23 03:52 python_games
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Templates
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Videos
pi@raspberrypi:~ $ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 Sep 23 03:52 Desktop
drwxr-xr-x 5 pi pi 4096 Sep 23 03:52 Documents
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Downloads
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Music
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Pictures
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Public
drwxr-xr-x 2 pi pi 4096 Sep 23 03:52 python_games
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Templates
drwxr-xr-x 2 pi pi 4096 Sep 23 04:03 Videos
pi@raspberrypi:~ $
```

Application shortcuts are across the top, as are open window tabs.

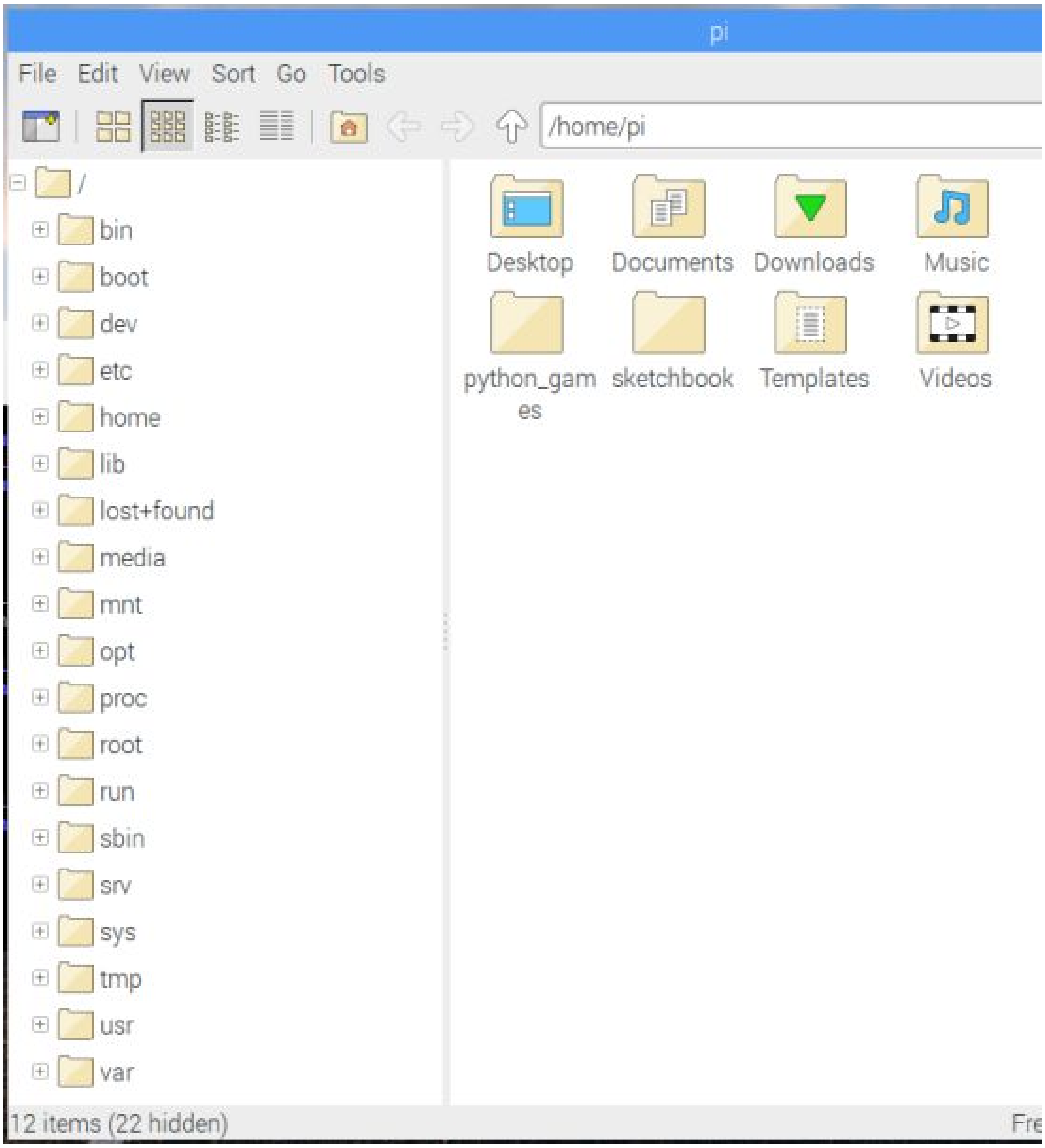
From left to right on the top menu bar:

- The raspberry icon on the left is the drop down application launch menu.
- The globe icon is the default browser is Chromium, the open source version of Chrome.
- The icon like stacked folder is the File Browser.

- The dark box icon with a right arrow is the terminal window where all command line shell programs are run.
- The other two red icons are free versions of Mathematica. These programs cost hundreds of dollars per year to license on a desktop computer, so Raspberry Pi is a great deal.

## File Manager

The graphical way to navigate through the linux file structure is with the File Manager application. As with all things linux, if you don’t like this program you can install a different file browser, even a command line one like mc (Midnight Commander)



Linux has a standard file system structure and the Raspberry Pi follows that template.

http://www.simplyembedded.org/archives/filesystems-with-the-raspberry-pi/

The website above lists the general use of the standard Linux directories.

**/bin:** directory where system executables (such as commands) go and are available to all users.

**/boot:** contains all the files required by bootloader to boot the kernel.

**/dev:** this directory is not actually part of the rootfs, it is created each time on boot and contains files that represent each of the devices which can be accessed in user space (look at the output from df to see that it is a special type of filesystem called devtmpfs).

**/etc:** where system and application configuration files are stored.

**/home:** contains another 'private' directory for each user. All of the user's personal files are typically stored in here. My documents, desktop, download etc.. all make up the a user's /home directory.

**/lib:** the directory where shared libraries and kernel modules are stored. When you compile and need to link in libraries, this is one directory where you would typically point the linker to.

**/media:** the default mount point for removable devices – for example, if you plug in a USB key

**/mnt:** Is the default mount point temporary filesystems (i.e. network shares)

**/opt:** Used to be where some third party applications are installed, supposed to be for add-on applications.

**/proc:** another virtual directory which is not really part of the root filesystem. In it are files which represent each of the processes in the system.

**/root:** the home directory for root

**/run:** a directory for applications to store data during runtime. This is not actually part of the rootfs, it is a temporary directory created at runtime.

**/sbin:** contains executables like the /bin directory, but these are typically for use only by the system and administrator.

**/sys:** another virtual filesystem which exposes some of the hardware interfaces in the kernel. It provides a way to see the kernel's configuration of the hardware. Though not recommended, you can access some hardware through this interface (i.e. read/write GPIOs, LEDs, etc)…

**/tmp:** a temporary filesystem, not actually part of the filesystem. Essentially a RAM disk which can be used by applications and the system to store temporary files

**/usr:** a directory where user application binaries, libraries, header files and documentation are stored.Typically files in here are all read-only. In some distributions this directory is on a separate partition from the root filesystem.

**/var:** another directory where application runtime information can be stored so that it doesn't end up in the /usr directory. Normally used for logs, locks. Some functions replaced by the /run directory but kept for compatibility.

## Command Line

On versions of Raspbian that have a GUI desktop, one must open a terminal window to execute command line programs. If the Raspberry Pi is running a non-GUI version of Raspbian, it will always boot into the command line terminal.

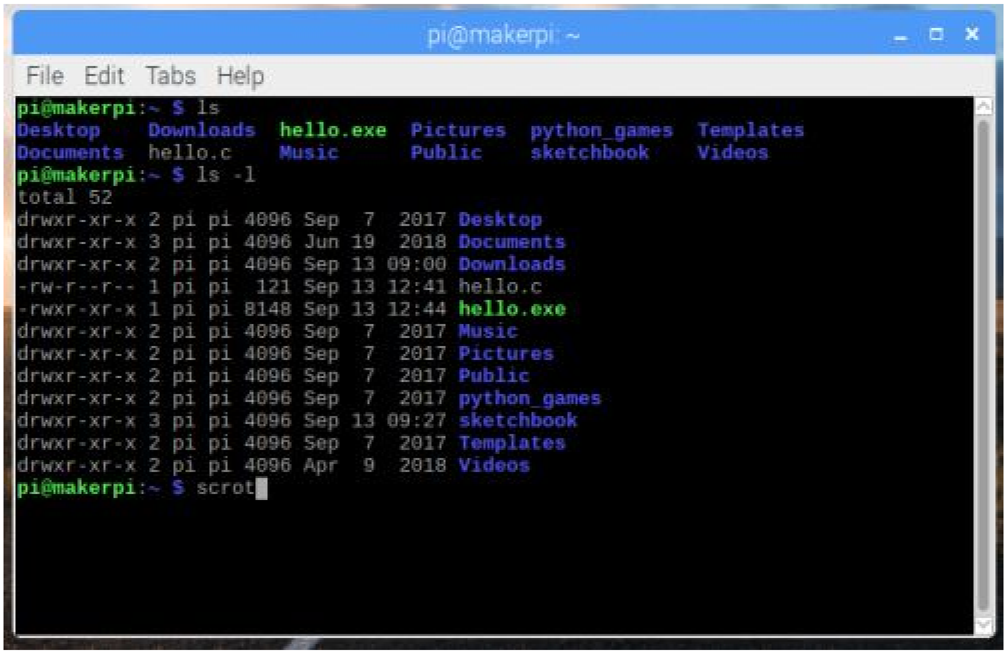
Ctrl-alt-t to open terminal window, or click the dark window icon. In the terminal window you can execute shell commands. The default shell on the Raspberry Pi is Bash, although others like Cshell can be installed. There can be many terminal windows open at the same time.
Linux comes with hundreds of default command line programs so that is often the fastest and easiest way to get things done.

Command line programs usually take parameters which are indicated with one or two dashes.

Unlike MS Windows, linux is case sensitive so "ls" is different than "LS"

The output from one command can be piped into another command or into a file.

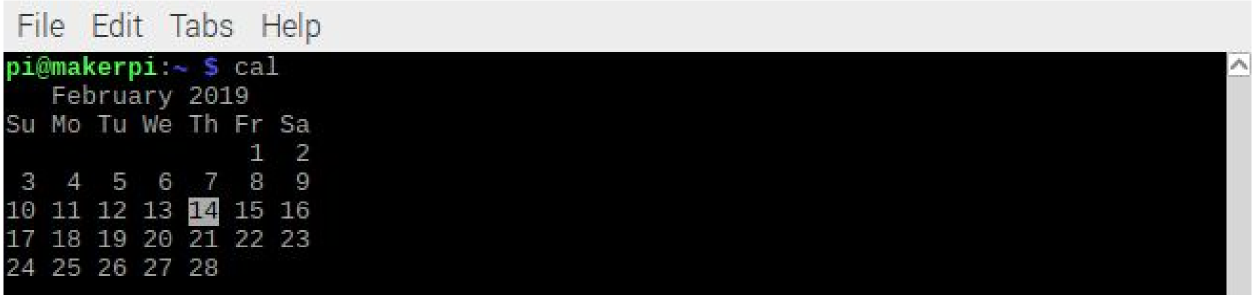
The command "ls" lists files. "ls -l" lists file details.



```
pi@makerpi:~ $ ls
Desktop    Downloads  hello.exe  Pictures  python_games  Templates
Documents  hello.c    Music      Public    sketchbook    Videos
pi@makerpi:~ $ ls -l
total 52
drwxr-xr-x 2 pi pi 4096 Sep  7  2017 Desktop
drwxr-xr-x 3 pi pi 4096 Jun 19  2018 Documents
drwxr-xr-x 2 pi pi 4096 Sep 13 09:00 Downloads
-rw-r--r-- 1 pi pi  121 Sep 13 12:41 hello.c
-rwxr-xr-x 1 pi pi 8148 Sep 13 12:44 hello.exe
drwxr-xr-x 2 pi pi 4096 Sep  7  2017 Music
drwxr-xr-x 2 pi pi 4096 Sep  7  2017 Pictures
drwxr-xr-x 2 pi pi 4096 Sep  7  2017 Public
drwxr-xr-x 2 pi pi 4096 Sep  7  2017 python_games
drwxr-xr-x 3 pi pi 4096 Sep 13 09:27 sketchbook
drwxr-xr-x 2 pi pi 4096 Sep  7  2017 Templates
drwxr-xr-x 2 pi pi 4096 Apr  9  2018 Videos
pi@makerpi:~ $ scrot
```

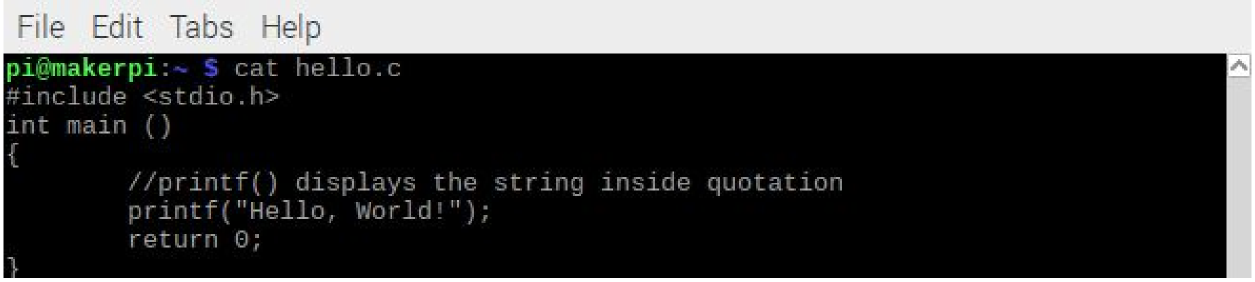
Note: the command “scrot” captures a picture of the screen to a file. There are many options such as “scrot -u” to capture the current window, “scrot -d 5” to delay five seconds, “scrot filename.png” to save to a particular filename.

The command “clear” clears the terminal window. “cal” lists a calendar. As you might guess, there are many options to “cal”. The default is display the current month.



```
pi@makerpi:~ $ cal
   February 2019
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28
```

The command “cat” copies a text file to the terminal screen.

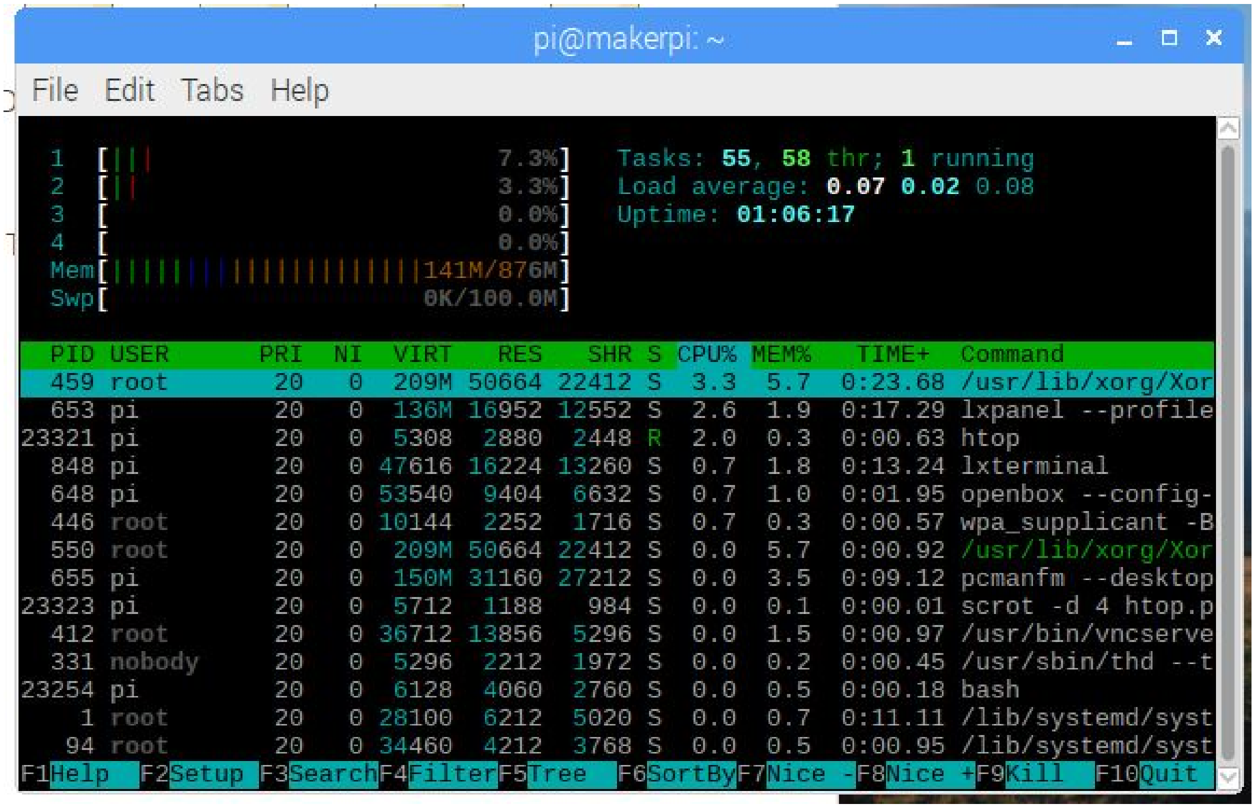


```
pi@makerpi:~ $ cat hello.c
#include <stdio.h>
int main ()
{
        //printf() displays the string inside quotation
        printf("Hello, World!");
        return 0;
}
```

The command “rm” removes (deletes) a file.
The command “cd” changes to another directory.
The command “mkdir” makes a directory directory.

Every linux file has an owner and permissions. The command “chmod” changes the permissions, “chown” changes the owner.

You can see all the executing processes with any of several commands. The most visual is “top” or “htop”

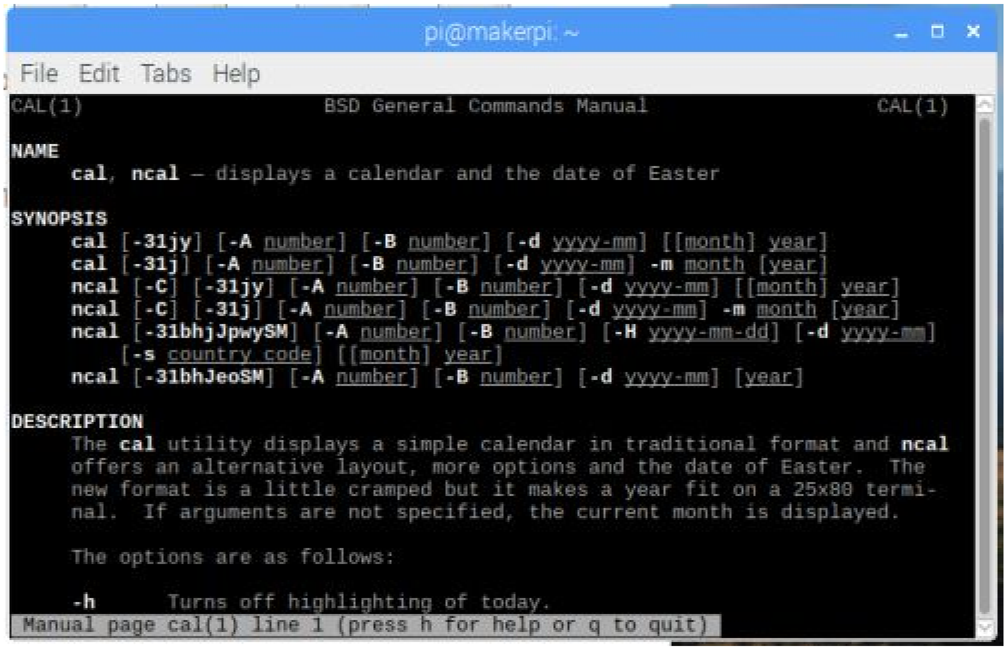


```
  1  [|||                       7.3%]   Tasks: 55, 58 thr; 1 running
  2  [||                        3.3%]   Load average: 0.07 0.02 0.08
  3  [                          0.0%]   Uptime: 01:06:17
  4  [                          0.0%]
  Mem[||||||||||||||||||||||141M/876M]
  Swp[                     0K/100.0M]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  459 root       20   0  209M 50664 22412 S  3.3  5.7  0:23.68 /usr/lib/xorg/Xor
  653 pi         20   0  136M 16952 12552 S  2.6  1.9  0:17.29 lxpanel --profile
23321 pi         20   0  5308  2880  2448 R  2.0  0.3  0:00.63 htop
  848 pi         20   0 47616 16224 13260 S  0.7  1.8  0:13.24 lxterminal
  648 pi         20   0 53540  9404  6632 S  0.7  1.0  0:01.95 openbox --config-
  446 root       20   0 10144  2252  1716 S  0.7  0.3  0:00.57 wpa_supplicant -B
  550 root       20   0  209M 50664 22412 S  0.0  5.7  0:00.92 /usr/lib/xorg/Xor
  655 pi         20   0  150M 31160 27212 S  0.0  3.5  0:09.12 pcmanfm --desktop
23323 pi         20   0  5712  1188   984 S  0.0  0.1  0:00.01 scrot -d 4 htop.p
  412 root       20   0 36712 13856  5296 S  0.0  1.5  0:00.97 /usr/bin/vncserve
  331 nobody     20   0  5296  2212  1972 S  0.0  0.2  0:00.45 /usr/sbin/thd --t
23254 pi         20   0  6128  4060  2760 S  0.0  0.5  0:00.18 bash
    1 root       20   0 28100  6212  5020 S  0.0  0.7  0:11.11 /lib/systemd/syst
   94 root       20   0 34460  4212  3768 S  0.0  0.5  0:00.95 /lib/systemd/syst
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

To list the current IP address, if connected to the Internet, type “hostname -I”. This is useful when using scp or ssh commands.

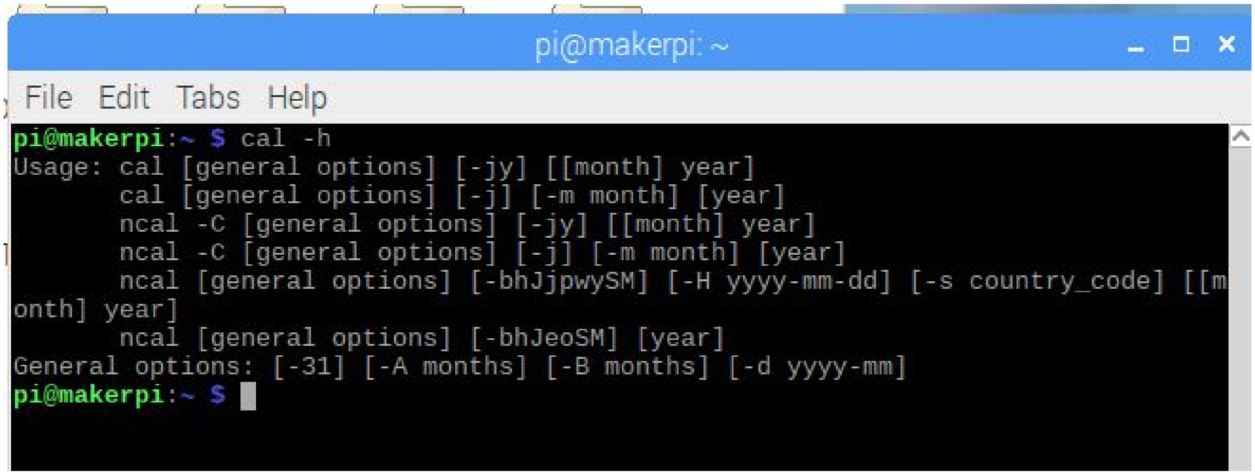
To run a program in the background add the “&” character at the end of the command.

All linux programs have a manual accessed with the “man” command. For example, “man cal” will bring up more information than you ever wanted.



```
CAL(1)                    BSD General Commands Manual                   CAL(1)

NAME
     cal, ncal — displays a calendar and the date of Easter

SYNOPSIS
     cal [-31jy] [-A number] [-B number] [-d yyyy-mm] [[month] year]
     cal [-31j] [-A number] [-B number] [-d yyyy-mm] -m month [year]
     ncal [-C] [-31jy] [-A number] [-B number] [-d yyyy-mm] [[month] year]
     ncal [-C] [-31j] [-A number] [-B number] [-d yyyy-mm] -m month [year]
     ncal [-31bhjJpwySM] [-A number] [-B number] [-H yyyy-mm-dd] [-d yyyy-mm]
          [-s country_code] [[month] year]
     ncal [-31bhJeoSM] [-A number] [-B number] [-d yyyy-mm] [year]

DESCRIPTION
     The cal utility displays a simple calendar in traditional format and ncal
     offers an alternative layout, more options and the date of Easter.  The
     new format is a little cramped but it makes a year fit on a 25x80 termi-
     nal.  If arguments are not specified, the current month is displayed.

     The options are as follows:

     -h      Turns off highlighting of today.
 Manual page cal(1) line 1 (press h for help or q to quit)
```

Many programs also have help. For example, “cal -h” gives a short summary of features.



```
pi@makerpi:~ $ cal -h
Usage: cal [general options] [-jy] [[month] year]
       cal [general options] [-j] [-m month] [year]
       ncal -C [general options] [-jy] [[month] year]
       ncal -C [general options] [-j] [-m month] [year]
       ncal [general options] [-bhJjpwySM] [-H yyyy-mm-dd] [-s country_code] [[m
onth] year]
       ncal [general options] [-bhJeoSM] [year]
General options: [-31] [-A months] [-B months] [-d yyyy-mm]
pi@makerpi:~ $ 
```

To copy a file from a remote computer to the current directory use
“scp user:(ipaddress):remote_file_name local_filename”
For example, to copy the file calman.png from the remote host pi at IP 192.168.3.12 to a local files name calendar_man_page.png use:

“scp pi:192.168.3.12:calman.png calendar_man_page.png”

Linux has wildcards like “rm *.txt” removes all files with the extension “txt”.   A single period “.” means the current directory, two periods means the higher directory. For example “cd ..” moves up one directory.

grep

## Installing and Updating Software

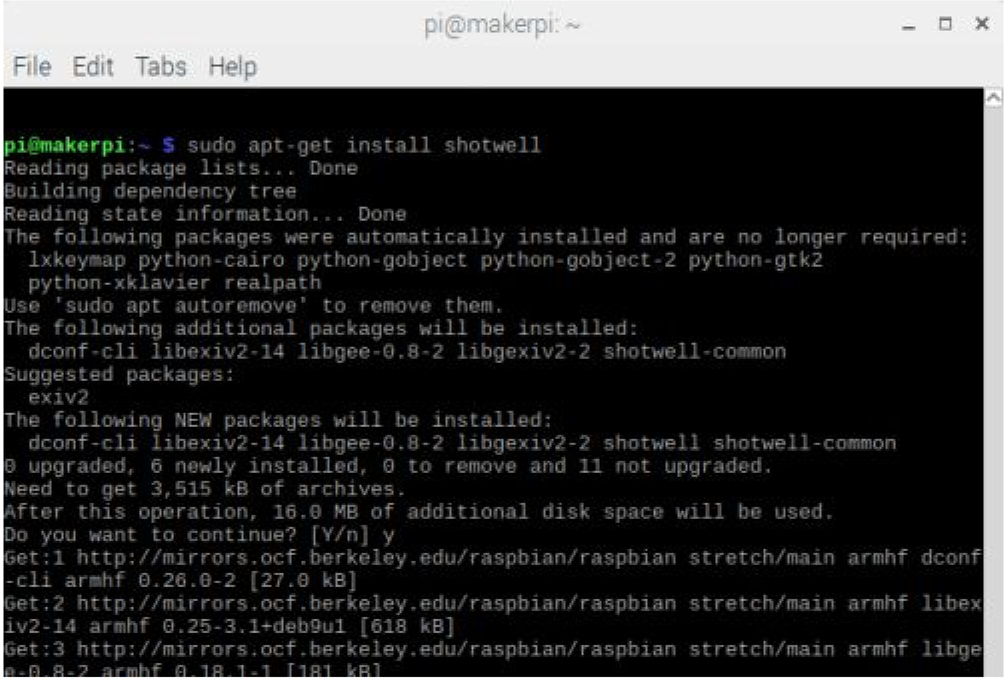
One of the greatest things about linux is how easy it is to update and install software. Due to the open source philosophy of the linux world, there are thousands of free programs to download, and even the source files for the programs.

Generally, to install new software requires an Internet connection. If no connection is available, then a *.deb file (package) can be downloaded on some other computer and transferred to the Raspberry Pi. It would be installed with the command “sudo dpkg -i debfilename”
The remaining description for installation is for Internet connected Raspberry Pis.

Each linux distribution has repositories of software. If something isn’t in the standard repositories, a user can download and install themselves, and even compile and create an executable version of the software themselves.

Sudo apt-get update
Sudo apt-get upgrade
Sudo apt-get install …

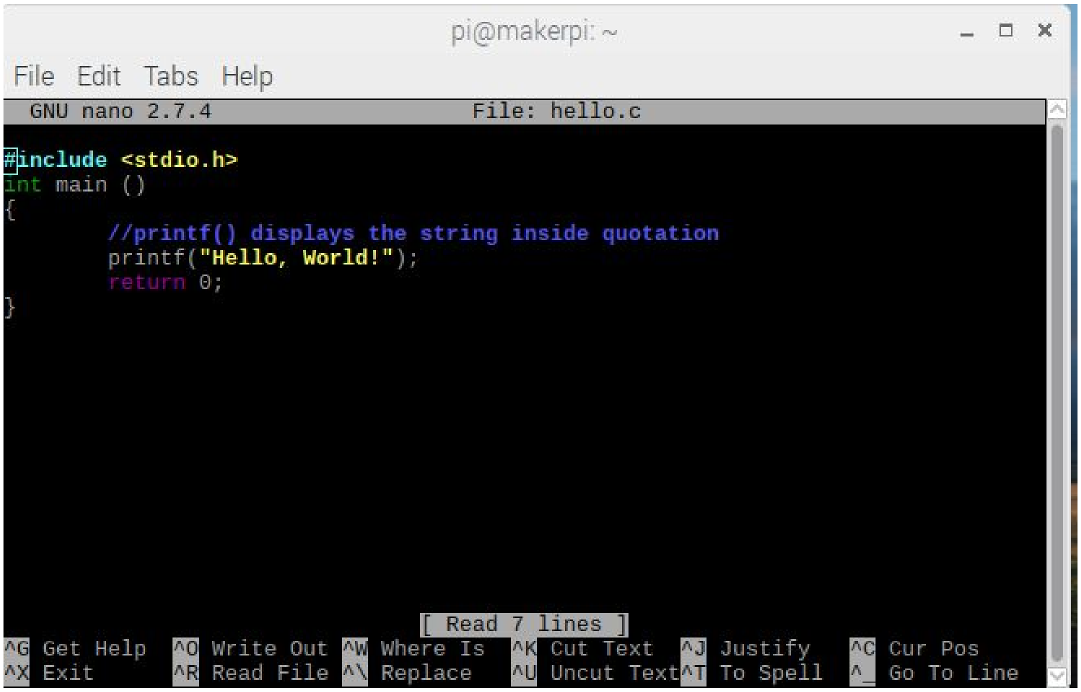
Here is an example of installing the image viewing software “shotwell” with the command “Sudo apt-get install shotwell”. It tells the user how much space is needed and confirms that is okay. Then it fetches and installs the program.



```
pi@makerpi:~ $ sudo apt-get install shotwell
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2
  python-xklavier realpath
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dconf-cli libexiv2-14 libgee-0.8-2 libgexiv2-2 shotwell-common
Suggested packages:
  exiv2
The following NEW packages will be installed:
  dconf-cli libexiv2-14 libgee-0.8-2 libgexiv2-2 shotwell shotwell-common
0 upgraded, 6 newly installed, 0 to remove and 11 not upgraded.
Need to get 3,515 kB of archives.
After this operation, 16.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.ocf.berkeley.edu/raspbian/raspbian stretch/main armhf dconf
-cli armhf 0.26.0-2 [27.0 kB]
Get:2 http://mirrors.ocf.berkeley.edu/raspbian/raspbian stretch/main armhf libex
iv2-14 armhf 0.25-3.1+deb9u1 [618 kB]
Get:3 http://mirrors.ocf.berkeley.edu/raspbian/raspbian stretch/main armhf libge
e-0.8-2 armhf 0.18.1-1 [181 kB]
```

# Editors and Programming

The default command line editor is nano. Type “nano filename to start the editor

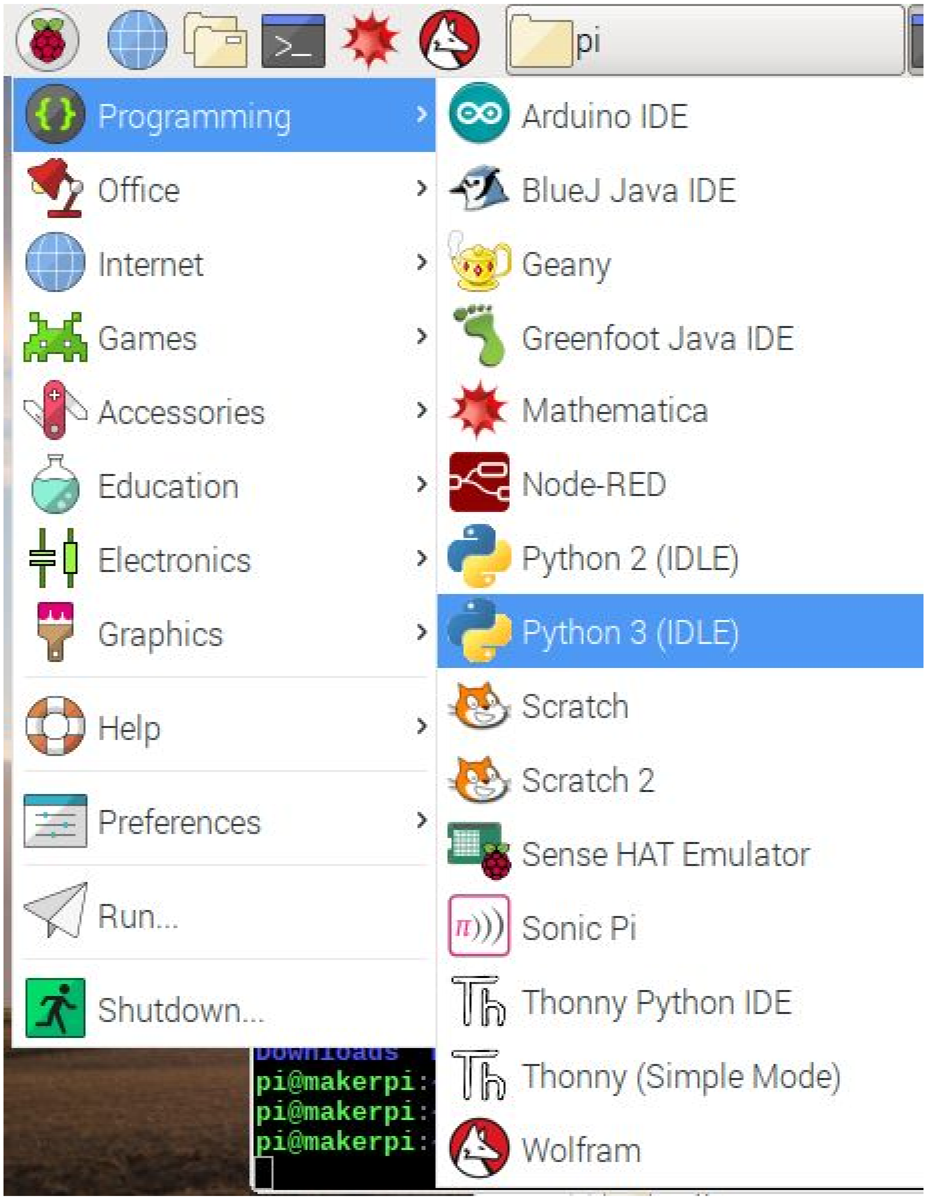


```
  GNU nano 2.7.4                    File: hello.c

#include <stdio.h>
int main ()
{
        //printf() displays the string inside quotation
        printf("Hello, World!");
        return 0;
}
```

```
                                [ Read 7 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

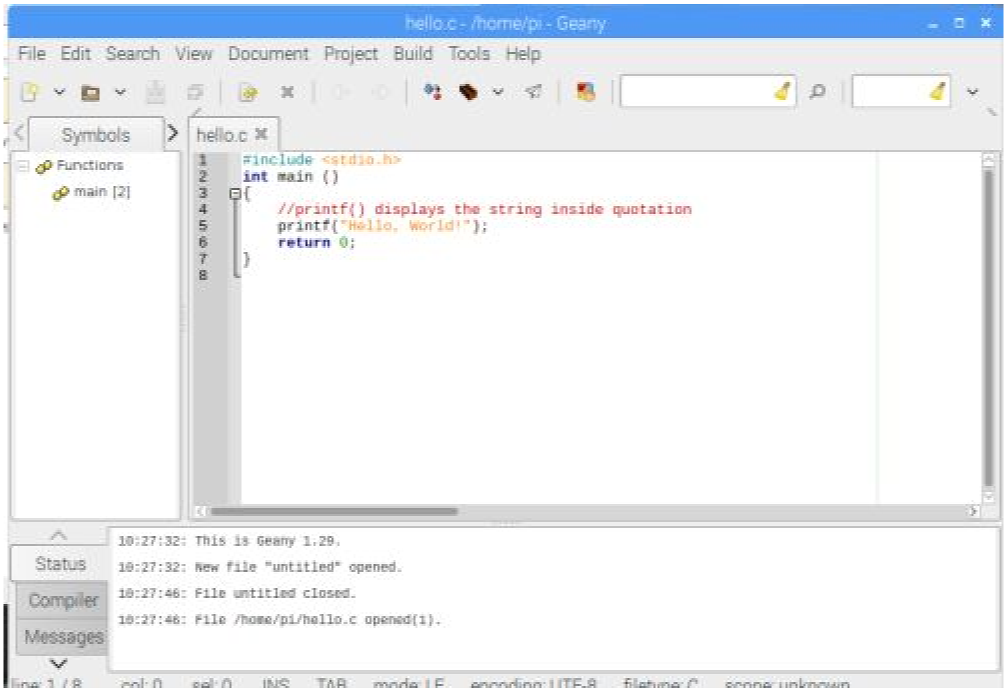
To compile this program called hello.c type “gcc hello.c -o hello.exe”

Some graphical text editing programs are pluma, Geany, Atom, or BlueJ. These are not default and have to be installed.



```
Downloads
pi@makerpi:
pi@makerpi:
pi@makerpi:
```

A popular lightweight editor is “Geany” which is installed with “sudo apt-get install geany”



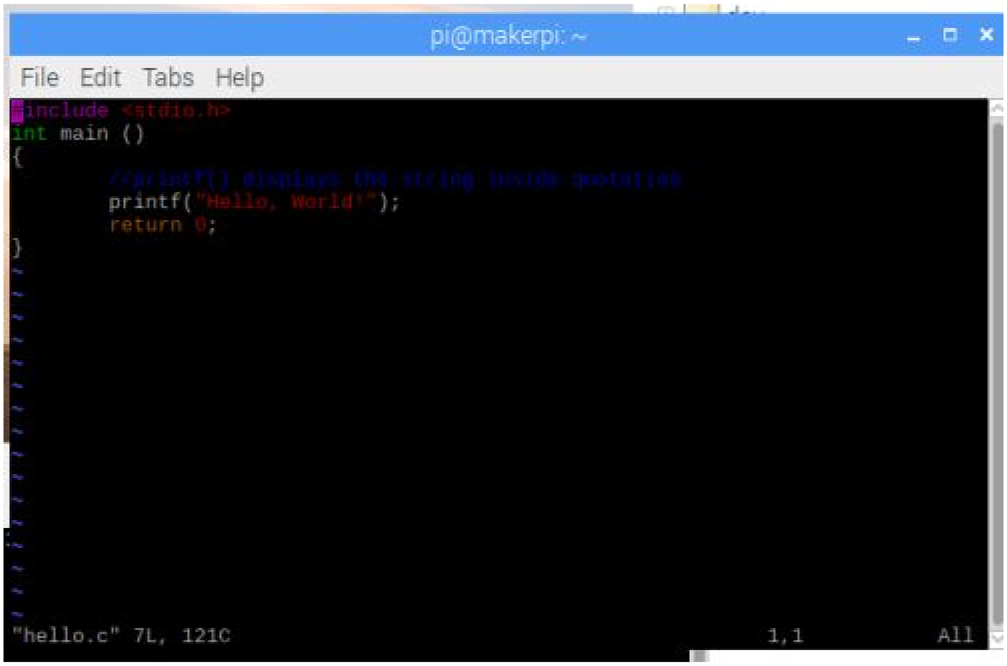
hello.c - /home/pi - Geany
File Edit Search View Document Project Build Tools Help
Symbols
Functions
main [2]
hello.c
Status
Compiler
Messages

```c
#include <stdio.h>
int main ()
{
    //printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

```
10:27:32: This is Geany 1.29.
10:27:32: New file "untitled" opened.
10:27:46: File untitled closed.
10:27:46: File /home/pi/hello.c opened(1).
```

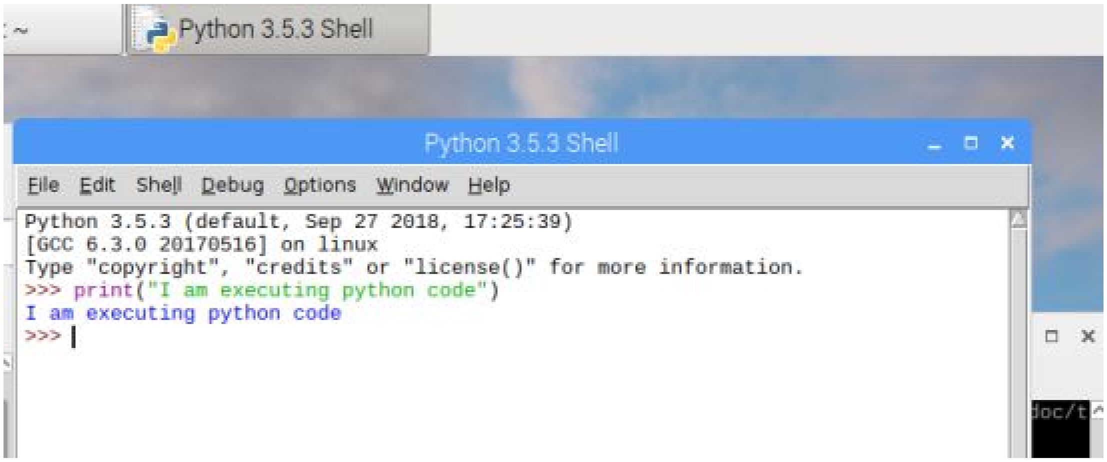
```
line: 1 / 8    col: 0    sel: 0    INS    TAB    mode: LF    encoding: UTF-8    filetype: C    scope: unknown
```

The two most popular command line editors are: vim and EMACS
Here is “vim hello.c”



```
#include <stdio.h>
int main ()
{
        //printf() displays the string inside quotation
        printf("Hello, World!");
        return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"hello.c" 7L, 121C                                            1,1           All
```

To execute the python interactive development environment, choose “Python3 IDLE” from the menu. Some older programs use Python 2 so that is available too.



```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>> print("I am executing python code")
I am executing python code
>>> 
```

The command line shell allows the creation of scripts that run multiple commands. This scripting ability is often used to automate installation or compilation of programs.

# Exercises

## Exercise 1: Raspi-config

***Hardware***: Raspberry pi 3 with Raspbian installed. Either a remote VNC connection or a screen, keyboard, and mouse.
***Software***: Raspbian
Execute Raspberry Pi configuration utility from the desktop (raspberry pi menu, preferences) and the command line with “sudo raspi-config”
Change SSH settings. Then change it back. Don’t restart the computer.

## Exercise 2: Bash Script

Create a bash file with nano text editor. In this example  it is called hello.sh. Where “sh” is the standard extension for shell scripts.
Here is the file:

```
#!/bin/bash
echo "Hello World"
ls -l
```

Make it executable:

```
chmod a+x hello.sh
```

Verify the chmod worked by doing ls -l

Execute a program with the “./” prefix. The . means the current directory and / means to execute.

```
./hello-world.sh
```

## Exercise 3: Copy a remote file

Find ip address with hostname -I

Using another computer on the same network, scp a file from that current computer.
For example:
scp pi@192.168.0.112:somefile .

You will be asked for the remote computer password. If the software Filezilla is installed, you can also copy the file using the desktop GUI.

Delete the file

## Exercise 4: Process control

start a long background process such as “find *.pdf &”. Or “nano hello.c”
View with htop or top, or ps.

> Note: htop uses mouse commands. Can also use the command line instructions; “pidof find” or “pidof nano”.

The processID will be a number such as 1296
Kill the process with “kill -9 processID”