

Internet of Things (IoT)

UNIX

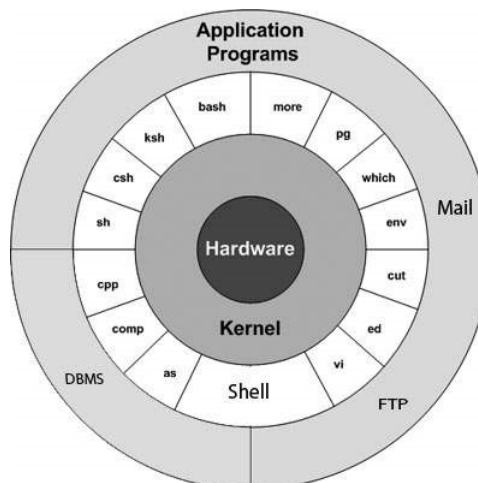
This unit is about one of the major operating systems, UNIX. We will be using UNIX commands in the following exercises of the workshop. If you are familiar with UNIX commands, you can skip this exercise. If not, please read the descriptions carefully and finish the exercise.

Introduction

UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. There are many different versions of UNIX, and the most popular varieties of UNIX are **Sun Solaris**, **Linux**, and **MacOS X**!

The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the **kernel**.

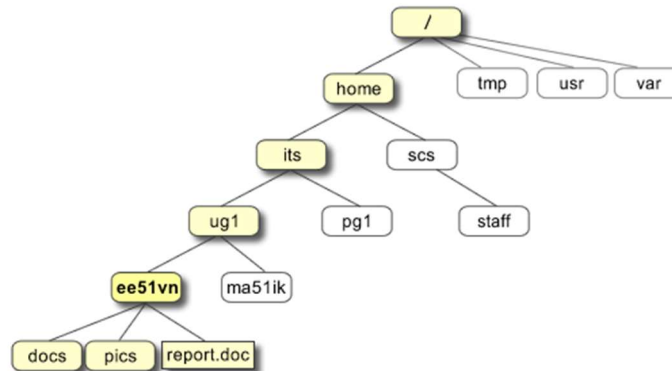


Users communicate with the kernel through a program known as the **shell**. The shell is a command line interpreter: it translates commands entered by the user and converts them into a language that is understood by the kernel.

Directory Structure

We keep a lot of files in a computer. In MS Windows, we would put files in different folders. A folder can have sub-folders and files. Each folder/file has a name. The way we organize files is called directory structure. It can be represented as an inverted tree with the **root** at the top, where branches (folders) expanded downwards. In MS Windows, the root is **C:**.

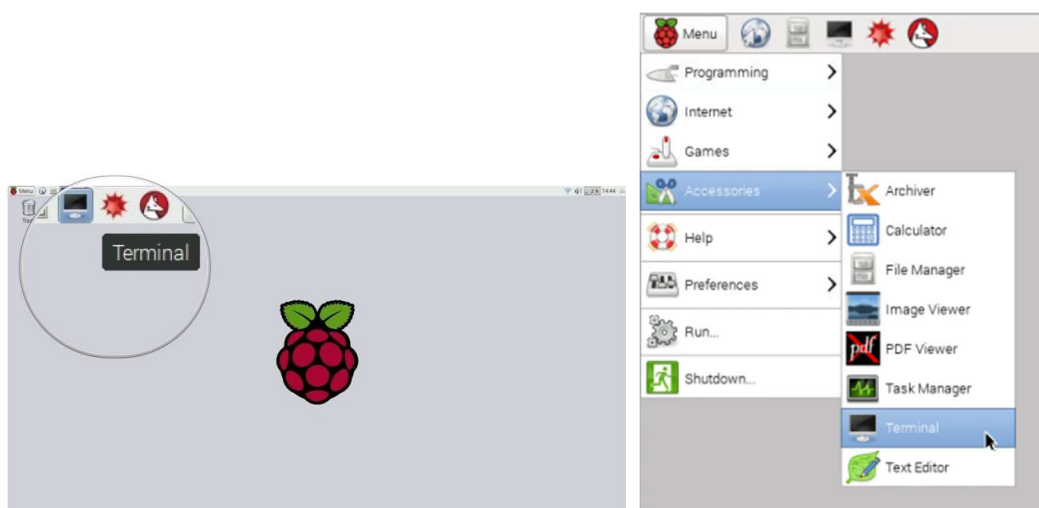
Similarly, in UNIX, the file-system is arranged in a hierarchical structure. The **root** is written as **/**.



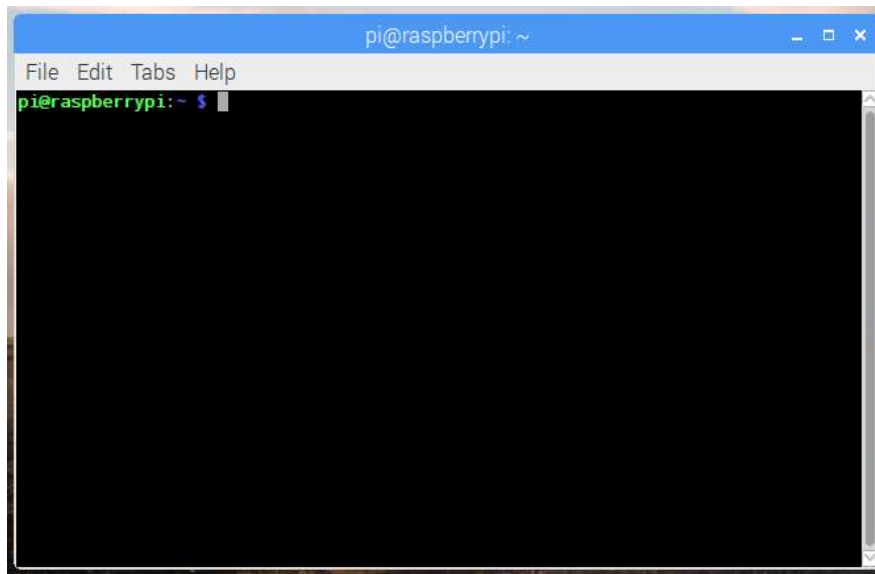
In the diagram above, we see that the home directory of the undergraduate student **ee51vn** contains two sub-directories (**docs** and **pics**) and a file called **report.doc**. The full path to the file **report.doc**, telling how to locate the file from the root, is: **/home/its/ug1/ee51vn/report.doc**

Useful UNIX Commands

To open a UNIX terminal window on your raspberry pi, click on the "**Terminal**" icon, or through **Menu -> Accessories -> Terminal**. Terminal is the counterpart of command prompt in MS Windows. It accepts text-based commands to issue instructions to the computer.



A terminal is something like this, where you can enter your commands:



The command prompt, **pi@raspberrypi~\$**, tells you some very useful information, especially you are working with several machines at the same time. **pi** is the username, **raspberrypi** is the machine name, and **~** is your current home directory.

Note that as a machine would keep files for different users, meaning the machine is keeping the home directories of John and Mary under two different folders. The name of the home directory of a user is the same of the username.

We now introduce some frequently used commands. Note that all commands are case-sensitive. In case you find some characters on your keyboard do not show up as you want, make sure you are using the right keyboard setting (Menu→Preferences→Mouse and Keyboard Settings, use United States keyboard layout).

- **ls (list)**

To find out what is in your home directory, type **ls** (lowercase L and lowercase S). The **ls** command lists the contents of your current working directory.

- **mkdir (make directory)**

We will now make a subdirectory in your home directory to hold the files you will be creating and using in this exercise.

To make a subdirectory (think of it as creating a new folder) called **workshop**, type **mkdir workshop**. Now, try to type **ls** again to see the directory you have just created.

- **cd (change directory)**

The command **cd directory** means changing the current working directory

to another directory called **directory**. The current working directory is your current position in the file-system tree. To change to the directory you have just made, type **cd workshop**. Type **ls** again to see the contents inside **workshop** (which should be empty).

- **pwd (print working directory)**

To find out your current working directory, type **pwd** to list the full path.

- **nano**

It is a text editor like notepad in MS Windows. **nano file** will open **file** using the text editor.

- **cp (copy)**

cp file1 file2 makes a copy of **file1** and names it as **file2**

- **mv (move)**

mv file1 file2 moves (or renames) **file1** to **file2**

To move a file from one place to another, use the **mv** command. This has the effect of moving rather than copying the file, so you end up with only one file rather than two.

It can also be used to rename a file, by moving the file to the same directory, but giving it a different name.

- **rm (remove), rmdir (remove directory)**

rm test.txt delete (remove) a file named **test.txt**

rmdir abcd remove an empty directory named **abcd**
(UNIX will not let you remove a non-empty directory!)

- **python3**

python3 file.py runs python program inside file.py

Here is a quick summary for your reference! 😊

Feel free to try out all commands including those not covered in the workshop today.

Command	Meaning
ls	list files and directories
ls -a	list all (including hidden) files and directories
mkdir	make a directory
cd <i>directory</i>	change to named directory
cd ~	change to home-directory
cd ..	change to parent directory
pwd	display the path of the current directory
cp <i>file1 file2</i>	copy file1 and call it file2
mv <i>file1 file2</i>	move or rename file1 to file2
rm <i>file</i>	remove a file
rmdir <i>directory</i>	remove a directory
cat <i>file</i>	display a file
less <i>file</i>	display a file a page at a time
head <i>file</i>	display the first few lines of a file
tail <i>file</i>	display the last few lines of a file
grep '<i>keyword</i>' <i>file</i>	search a file for keywords
wc <i>file</i>	count number of lines/words/characters in file
sort	sort data
who	list users currently logged in
man <i>command</i>	read the online manual page for a command
chmod [<i>options</i>] <i>file</i>	change access rights for named file
python3 <i>file</i>	run python file

Exercise – Using UNIX commands

1. Create a directory called “***ILoveEEE***” under your ***home*** directory
 2. Under directory “***ILoveEEE***”, create a file named “***ghost***” using any text editors (e.g. nano) and add some contents into it (e.g. “*EEE is very interesting!*”). You can type “**nano ghost**” directly in the command prompt.
 3. Copy the file “***ghost***” to your home directory. Note that ~ represents the home directory. [You might need to change the keyboard type to display ~ properly.] Use “**cp ghost ~**” command.
 4. Go back to your home directory, **rename** the file “***ghost***” into “***secret***”
 5. Delete the directory “***ILoveEEE***” and all of its contents
-