



# Aerial and Underwater Robotics Society | JUIT

## Introduction to Linux Operating System

### Contributors:

Akhilesh Kumar

Aditya Tiwari

This tutorial is meant for internal distribution to the members of the AURS | JUIT team only. Any form of sharing or usage of content outside is strictly prohibited without the permission from the team heads

### Module 1

#### Introduction to Linux

---

Hello people! Let's get familiar with Linux Operating System.

- Linux is one popular Unix based Operating System.
- It is just like any other Operating System, trust me much powerful than others!
- Gave us the all GNU things we have right now! It's free to use strongly supports Open Source. You can find all the source codes online.

### Why Linux?

#### Package Managers

- Linux operating system enable download of third party softwares through package managers unlike windows.
- We don't need to face setup/dependencies hassle while installing software.
- Ubuntu uses the ***apt-get*** utility as the package manager.
- Typing ***sudo apt-get install*** is all you need to do! :-D

#### The Terminal \$

- Linux's **Terminal** is much more powerful when compared to Window's **Command Prompt**.
- The **Terminal** can practically do anything. Can delete everything from your computer too!

- Make it your best friend as soon as possible.

### The “Developers” Feel

- As a Linux User, one would need to spend a lot of time on the **Terminal**.
- Typing feels a lot more like a programmer when compared to clicking.
- Linux has several flavors, each having its strength and weakness. You can always have choosing between so many distros.

## Module 2

### Installing Ubuntu 16.04 LTS

---

- Download the Ubuntu ISO files or Copy from our folder on Server.
- Burn the ISO file on an installation Disk using software like UUI, Rufus. We prefer Rufus.
- Boot from the Disk and Go for a clean install. We won't count Dual Boot or using Virtual Machines, that doesn't make you learn anything.
- On the installation Disk, you will find an option of try Ubuntu. Go for it, see how it feels. See the build, file systems and everything else.
- Install Ubuntu on your machine.
- Set your partition sizes, if needed. Less partitions are recommended for better disk health.
- Personalize your installation.
- Once installed, reboot your PC and do come customizations.

**Congratulations!**

**Now You Have Ditched Windows for Good :-P**

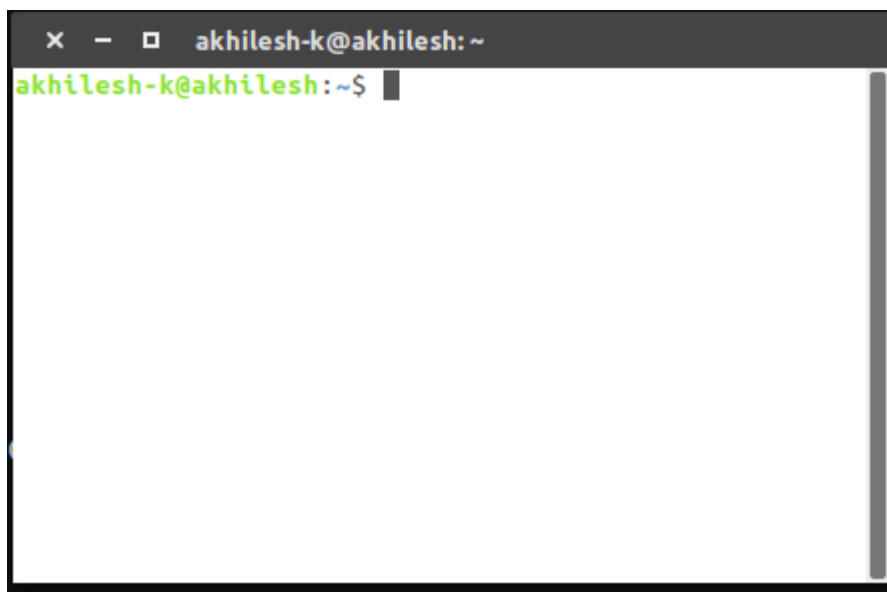
## Module 3

### Basic Commands

---

#### Your first time with Terminal! Let's get started

If you've don't have any experience with terminal, then it's the right time to get yourself introduced to the mighty terminal!



This is how a typical terminal looks like.

~ represents home directory

ls Lists all files in the home directory

cd Downloads Directory changed to Downloads

cd .. Go one directory back

/ Root folder

Press Tab for auto completion of commands or add \* after command.

#### Colors in Terminal:

Blue: Represents a folder

Green: An executable file

Pink: Graphical File

Red: Archive file

## Installing Softwares

`sudo apt-get update`

`sudo apt-get install firefox`

`sudo` Super Do (Grants root access)

`apt-get` The Package manager of Ubuntu

`update` Listing of changes in repository. It doesn't actually download the repository.

## Basic file operation

`mv` move file

`rm` remove file

`cp` copy file

## Downloading files

`wget` Simply downloads the file off the network.

## Programmer's Playground

All paths start from the root folder.

<code>usr/include/</code>	All header files are here
<code>usr/lib/</code>	The corresponding library files
<code>usr/bin/</code>	The home for all the executable files
<code>etc/NetworkManager</code>	configure network here

\*Using these folders require super user permission

\$ `sudo su`

## Random Commands

`Ctrl+C` or `Ctrl+Z` End programs

`Ctrl+Shift+C` Copies from terminal

**ifconfig** Details of network status

**find** Used to locate files on PC

## Useful Commands

**grep** Used to find the snippets of text from set of files

**top** Top displays running processes on the PC

```

x - □ akhilesh-k@akhilesh: ~/Downloads
top - 15:59:20 up 18:48, 1 user, load average: 0.08, 0.16, 0.17
top - 15:59:21 up 18:48, 1 user, load average: 0.15, 0.17, 0.17
top - 15:59:21 up 18:48, 1 user, load average: 0.15, 0.17, 0.17
top - 15:59:21 up 18:48, 1 user, load average: 0.15, 0.17, 0.17
top - 15:59:21 up 18:48, 1 user, load average: 0.15, 0.17, 0.17
Tasks: 216 total, 2 running, 214 sleeping, 0 stopped, 0 zombie
%Cpu(s): 50.0 us, 16.7 sy, 0.0 ni, 33.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2923700 total, 304948 free, 1755104 used, 863648 buff/cache
KiB Swap: 3057660 total, 3008284 free, 49376 used. 639948 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 2122 akhiles+  20   0 3494756 875256 107404 S 100.0 29.9   81:28.77 firefox
    999 root      20   0 483616   84976  69356 R  50.0   2.9   12:20.66 Xorg
 1904 akhiles+  20   0 1227728 101904  50324 S  50.0   3.5   13:40.36 compiz
10099 akhiles+  20   0 656000   39420  28920 S  50.0   1.3    0:10.95 gnome-terminal-
top - 15:59:21 up 18:48, 1 user, load average: 0.15, 0.17, 0.17
Tasks: 216 total, 3 running, 213 sleeping, 0 stopped, 0 zombie
%Cpu(s): 15.4 us, 3.1 sy, 0.0 ni, 81.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

```

**Killall** | kill or killall <processname>

**kill** |

**cat** used to view the contents of file

\*one we will use in finding the IP of Raspberry Pi to establish a secure shell connection.

## Module 4

### Essential Softwares

---

#### Your first time with Linux!

If you've never used Linux before, there are a bunch of things that you need to do.

- IDE/Editors

##### **Sublime Text:**

- `$ sudo add-apt-repository ppa:webupd8team/sublime-text-2`
- `$ sudo apt-get update`
- `$ sudo apt-get install sublime-text`

##### **Vim:**

- `$ sudo apt-get install vim`
- `$ vim filename.py`

##### **Android Studio:**

Use the link provided to download Android Studio

<https://developer.android.com/sdk/installing/index.html?pkg=studio>

You might have to make studio.sh an executable and then run it. Do following for the same.

- `$ sudo chmod 777 studio.sh`
- `$ ./studio.sh`

##### **Arduino IDE:**

- `$ sudo apt-get install arduino arduino-core`

##### **Gazebo**

Use the link provided to download Gazebo, it is available for Linux only.

<https://gazebo-sim.org/download>

Run the debian file to install.

### **Fritzing**

Use the link provided to download Fritzing

<https://fritzing.org/download>

Run with command `./Fritzing`

### **Python**

- `$ sudo apt-get install python3`

Check the installed version

- `$ python3 -V`

Use Pip3 as package manager for python.

- `$ sudo apt-get install python3-pip`

### **CodeBlocks**

- `$ sudo apt-get install codeblocks`

### **Octave**

- `$ sudo apt-get install octave`

(Don't type the \$; that just indicates that you're doing this at the command line.)

## Module 5

## Shell Scripting

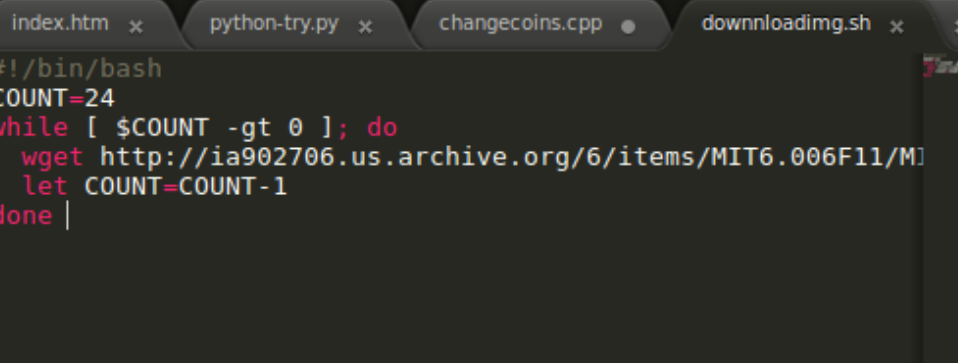
## Just a Bit about Shell Scripts

Shell scripting is fun, it is the same programming thing with if, for, switch and the 10B11C1111 stuffs and a terminal in hands!

Magic. We got the same things just in new avatar.

`./run.sh` You will need to make it executable and run like this.

Lets see what a script looks like

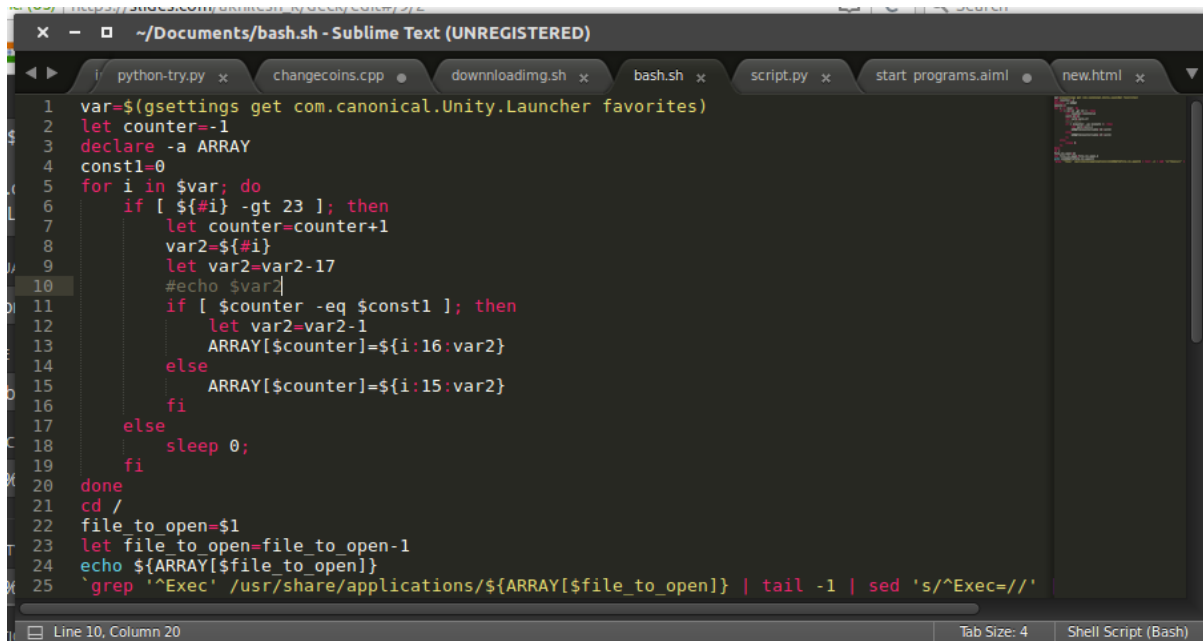


```
1 #!/bin/bash
2 COUNT=24
3 while [ $COUNT -gt 0 ]; do
4     wget http://ia902706.us.archive.org/6/items/MIT6.006F11/M
5     let COUNT=COUNT-1
6 done |
```

Line 6, Column 6      Tab Size: 4      Shell Script (Bash)



This script is programmed to download 24 gif images from MITOCW website. It pretty much automates life!

A screenshot of a Sublime Text editor window titled '~/.Documents/bash.sh - Sublime Text (UNREGISTERED)'. The editor shows a bash script with the following content:

```
1 var=$(gsettings get com.canonical.Unity.Launcher favorites)
2 let counter=-1
3 declare -a ARRAY
4 const1=0
5 for i in $var; do
6     if [ ${#i} -gt 23 ]; then
7         let counter=counter+1
8         var2=${#i}
9         let var2=var2-17
10        #echo $var2
11        if [ $counter -eq $const1 ]; then
12            let var2=var2-1
13            ARRAY[$counter]=${i:16:var2}
14        else
15            ARRAY[$counter]=${i:15:var2}
16        fi
17    else
18        sleep 0;
19    fi
20 done
21 cd /
22 file_to_open=$1
23 let file_to_open=file_to_open-1
24 echo ${ARRAY[$file_to_open]}
25 `grep '^Exec' /usr/share/applications/${ARRAY[$file_to_open]} | tail -1 | sed 's/^Exec=/'`
```

The status bar at the bottom indicates 'Line 10, Column 20', 'Tab Size: 4', and 'Shell Script (Bash)'. The script is designed to process a list of favorites from a Unity Launcher, splitting them into segments and opening them using a command like 'tail -1 | sed 's/^Exec=/''.

Here a pretty more advanced version. Here number inputs are taken via terminal arguments.

Here we have just introduced what is bash and some little things about shell scripts. This was just a motivational page.