Linux:

- Linux is a family of free and open-source software operating systems built around the Linux kernel.
- Typically, Linux is packaged in a form known as a Linux distribution (or distro for short) for both desktop and server use (such as: Ubuntu, Red hat, CentOS, Fedora, OpenSuse and many more).
- The defining component of a Linux distribution is the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds.
- Many Linux distributions use the word "Linux" in their name.
- The Free Software Foundation uses the name GNU/Linux to refer to the operating system family, as well as specific distributions, to emphasize that most Linux distributions are not just the Linux kernel, and that they have in common not only the kernel, but also numerous utilities and libraries, a large proportion of which are from the GNU project.

Virtual Box

- In order to use Ubuntu OS on our computer we have 2 options:
- Installing it as an OS alongside our Windows OS (Dual boot).
- Using virtual machine (Oracle Virtual box/ VMware).
- To simplify our work, we are going to use the second option using Oracle VirtualBox which can be found online https://www.virtualbox.org/wiki/Downloads
- VirtualBox is powerful *Cross-platform Virtualization Software*.
- "Cross-platform" means that it installs on Windows, Linux, Mac OS X and Solaris x86 computers.
- And "Virtualization Software" means that you can create and run multiple Virtual Machines, running different operating systems, on the same computer at the same time.
- For example, you can run Windows 7 and Linux Ubuntu on your Windows 10 machine.
- Each virtual machine has a all the abilities of a regular computer includs:
    - Motherboard Options
    - CPU options
    - RAM options
    - Drives (CD/Floppy, Hard disks)
    - Audio
    - Network
    - And more..

Shell & Terminal

- **Shell –**
  - Shell is a program that takes commands from the keyboard and gives them to the operating system to perform.
  - In the old days, it was the only user interface available on a Unix-like system such as Linux.
  - Nowadays, we have graphical user interfaces (GUIs) in addition to command line interfaces (CLIs) such as the shell.
  - On most Linux systems a program called bash.
- **Terminal-** It's a program called a terminal emulator.
  - This is a program that opens a window and lets you interact with the shell.
  - There are a bunch of different terminal emulators you can use.
- **Sudo (super user do)-**
  - Sudo is a program for Unix-like computer operating systems that allows users to run programs with all security privileges, to perform any kind of operation on the machine.
  - Sudo can be looked at like the "**Run as administrator**" on Windows machine

Commands –

- Basically, a shell is a program that receives commands from the user and gives it to the OS to process, and then shows the output.
- To open the terminal, press Ctrl+Alt+T in Ubuntu
- Here are few commands:

pwd -

- When you first open the terminal, you are in the home directory of your user.
- To know which directory you are in, you can use the **“pwd”** command.
- It gives us the absolute path, which means the path that starts from the root.
- The root is the base of the Linux file system.
- It is denoted by a forward slash(/).
- The user directory is usually something like "/home/username".

ls -

- Use the **"ls"** command to know what files are in the directory you are in.
- You can see all the hidden files by using the command **“ls -a”**.

cd -

- Use the **"cd"** command to go to a directory.
- For example, if you are in the home folder, and you want to go to the /john/downloads folder, then you can type in **“cd** /john/downloads**”**.
- Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is.

mkdir & rmdir & rm —

- Use the **mkdir** command to create a folder or a directory.
- For example, if you want to make a directory called “a”, then you can type **“mkdir a**”.
- Use **rmdir** to delete a directory (will work only on empty directories).
- To delete a directory containing files, use **rm**.

touch -

- The **touch** command is used to create a file.
- It can be anything, from an empty txt file to an empty zip file. For example, “**touch users.txt**”.

cp & mv-

- Use the **cp** command to copy files through the command line.
- It takes two arguments: The first is the location of the file to be copied, the second is where to copy.
- For example: **cp users.txt /users/users/music**
- mv will move the file instead of only copying it – **mv users.txt /users/users/music**

echo-

- The "**echo**" command can be used to write text into a file.
- For example, if you want to create a new text file or add to an already made text file, you just need to type in, "**echo hello world >> new.txt**".

cat -

- The **cat** command is used to display the contents of a file, and its usually used to easily view programs.
- For example **cat users.txt** will print the text file content to terminal.

tar -

- Tar is used to work with tarballs (or files compressed in a tarball archive) in the Linux command line.
- It has a long list of uses. It can be used to compress and uncompress different types of tar archives like .tar, .tar.gz, .tar.bz2,etc.

zip & unzip -

- zip is used to compress files into a zip archive, and unzip to extract files from a zip archive.
- Foe example: zip 123.zip

Chmod

- chmod is used to change the permissions of files or directories.
- On Linux and other Unix-like operating systems, there is a set of rules for each file which defines who can access that file, and how they can access it.
- These rules are called file permissions or file modes.
- The command name chmod stands for "change mode", and it is used to define the way a file can be accessed.
- permissions defines the permissions for the:
    - Owner of the file (the "user")
    - Members of the group who owns the file (the "group")
    - Anyone else ("others").
- There are two ways to represent these permissions: with symbols (alphanumeric characters), or with octal numbers (the digits 0 through 7).
- Let's say you are the owner of a file named myfile, and you want to set its permissions so that:
    - The **user** (owner) can read, write, and execute it.
    - **Members** of your group can read and execute it.
    - And **others** may only read it.
- To achieve it we will run the command **chmod 754 myfile**
- Here the digits 7, 5, and 4 each individually represent the permissions for the **user**, **group**, and **others** in that order.
- Each digit is a combination of the numbers 4, 2, 1, and 0:

    - 4 stands for "read"
    - 2 stands for "write"
    - 1 stands for "execute"
    - 0 stands for "no permission."
- 7 is the combination of permissions 4+2+1 (read, write, and execute),
- 5 is 4+0+1 (read, no write, and execute).
- 4 is 4+0+0 (read, no write, and no execute).

Package installer

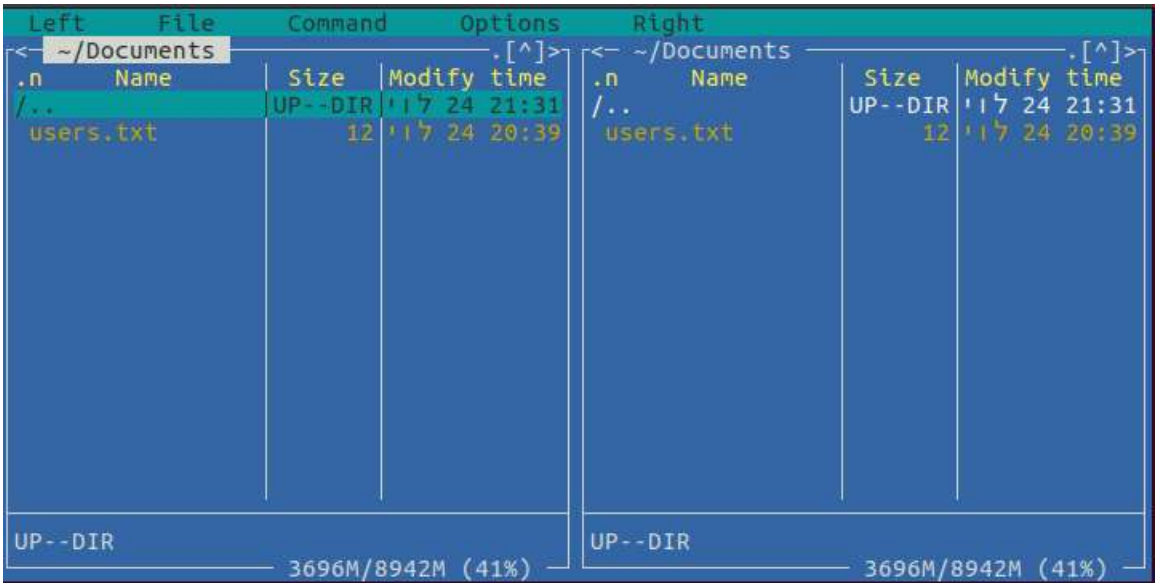
- Apt is a command line frontend for the dpkg packaging system and is the preferred way of managing software from the command line for many distributions.
- Apt-get operates on a database of known, available software.
- It performs installations, package searches, and many other operations.
- Due to this fact, before beginning any packaging operations with apt-get, we need to ensure that our local copy of the database is up-to-date.
- Update the database with the following command. Apt-get requires administrative privileges for most operations:
- Here are a few operations:
    - Install a package: **sudo apt-get install package1**
    - Upgrade a package: **sudo apt-get remove package1**

Midnight commander

- GNU Midnight Commander is a visual file manager, licensed under GNU General Public License and therefore qualifies as Free Software.
- It's a feature rich full-screen text mode application that allows you to copy, move and delete files and whole directory trees, search for files and run commands in the subshell. Internal viewer and editor are included.
- To install MC, open a terminal and type:

  sudo apt-get install mc

- To run just type mc or sudo mc (for su privileges).



VI & VIM & Nano

- Linux has a few text editors.
- They can all achieve a similar functionality, but they work differently.
- **VI** – (for visual editor)
    - Vi is the first one and its initial release date was 1976.
- **VIM –** (" vi improved")
    - As the name suggest it adds lot of functions to the original vi interface.
- **Nano –** (" vi improved")
    - nano has always been known to be user-friendly when it comes to terminal text editors.
    - To edit a text file, simply, call the desired editor and the file name **sudo <vi / vim / nano> users.txt**

VI

- Lets start with VI edit: vi <file_name>
- While working with the vi editor, we usually come across the following two modes −
    - **Command mode** − This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting and pasting the lines or words.
    - **Insert mode** − This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and placed in the file.
- vi always starts in the command mode.
- To enter text, you must be in the insert mode for which simply type **i**
- To get out of the insert mode, press the **Esc** key, which will take you back to command mode.
- To save press **:wq**
- Lets try the following instructions: open file (**vi <filename>**) → press **i** (to enter insert mode)→ type anything → press **esc** (to change back to command mode) → type **:wq** (to save and exit)
- In case vi is called on a file that does not exist it will create it once we save the file.

VIM

- vim has everything vi has.
- To open a file with Vim, use: vim <file_name>
- Vim adds onto those features.
- Here are a some of the extended vim features:
    - Vim includes support (syntax highlighting, code folding, etc) for several popular programming languages (C/C++, Python)
    - Vim includes multilevel undo/redo.
    - Vim allows the screen to be split for editing multiple files.
    - Vim includes a built in diff for comparing files (vimdiff).
- Vim doesn't come by default in Ubuntu, to get it we will need to install it: **sudo apt-get install vim**

Nano

- Nano is the easiest most user-friendly editor
- One of the cool things about nano, is the fact that when you want to use "control" mode, you will use the combination of the Ctrl key with any key, without pressing the control key, it will be using the "insert" mode.
- To use Nano, type: **nano <FILE-NAME> , e.g: nano 1.txt**
- In order to go to a line use arrows
- To save and exit press the file press **Ctrl+X** and then **y**
- Nano also gives us the ability to search text inside the file using **Ctrl+W**

SSH

- Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.
- The standard TCP port for SSH is 22. The best known example application is for remote login to computer systems by users.
- SSH provides a secure channel over an unsecured network in a client-server architecture, connecting an SSH client application with an SSH server.
- Common applications include remote command-line login and remote command execution, but any network service can be secured with SSH.
- SSH uses public-key cryptography to authenticate the remote computer and allow it to authenticate the user, if necessary.
- There are several ways to use SSH;
    - one is to use automatically generated public-private key pairs to simply encrypt a network connection, and then use password authentication to log on.
    - Another is to use a manually generated public-private key pair to perform the authentication, allowing users or programs to log in without having to specify a password.

- To SSH, we will need both client and server.
- Both Linux and Mac OS X has a built-in SSH.
- On Windows 10, Microsoft added SSH option as well.

- Server setup:
    - Open terminal (in your Ubuntu).
    - Install OpenSSH server **sudo apt-get install openssh-server**
    - Find out server IP address by using **ifconfig** (normally it will be enp0s?)
- Client setup:
- If you Mac OS X / Linux / Windows 10, simply open CMD/Terminal and type: **ssh <user name>@<server IP address>**
- In case you have something other, you can download a software named **Putty**, enter the server address and press **Open**

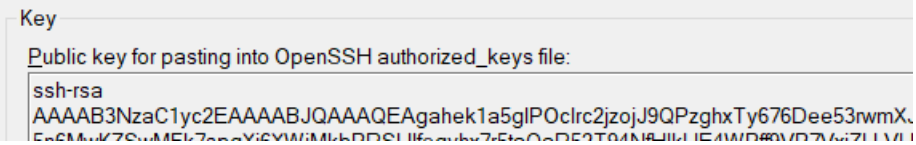
Cyberduck

- **Cyberduck** is a free open-source, cross platform tool which offers a secure file transfer between a local and a remote computer.
- Beyond this, it offers a basic file manager and file synchronization functionality.
- For secure transfers, it uses Secure Shell (SSH).
- Download can be found at https://cyberduck.io/
- Once installed you can create a new connection to your server by preesing **Open connection** changing to **SFTP** and typing **server address**, **Username** and **Password**
- Once the connection established you can explore the files on the server and simply drag them into your computer or from your computer into the server.

SSH keys

- SSH keys provide a more secure way of logging into a server with SSH than using a password alone.
- While a password can eventually be cracked with a brute force attack, SSH keys are nearly impossible to decipher by brute force alone.
- The first step is to create a key pair on the client machine (your computer): **ssh-keygen**
- You will be asked about the creation location (you can simply press enter).
- Next you will be asked to add passphrase which is highly recommended.
- A passphrase adds an additional layer of security to prevent unauthorized users from logging in case of a stolen key.
- You now have a public and private key that you can use to authenticate.
- The next step is to place the public key on your server so that you can use SSH-key-based authentication to log in.

- Windows users-
    - Download and use PuttyGen to create a public & private keys using passphrase
    - Copy key text and save it as authorized_keys file (no .txt)



    - Mac OS X users-
    - In the terminal **ssh-keygen -t rsa** and add a passphrase
- Create a folder in server under the user name called .ssh and paste file using Cyberduck
- Open /etc/ssh/sshd_config as with sudo nano and change PasswordAuthentication to no
- Restart ssh server – **sudo service ssh restart**
- Disconnect and connect using Cyberduck with the private key you generated

- ** Please note: When you paste the key, it needs to be all on one line, with no extra spaces.