Introduction to Linux

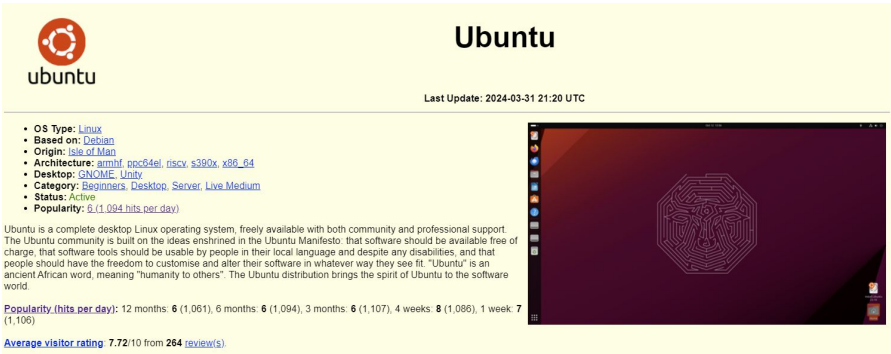
Nguyen Le Duc Minh, MD

Benefits of Linux

- Cost
  - Many free options
  - Paid options include support
    - Red Hat Enterprise Linux desktop: 49$/seat
    - SUSE Linux Enterprise desktop: 120$
- Open source
- Reduced attack surface
  - Windows: ~Swiss Army Knife (does a lot of different things)
  - Linux: ~Hammer
- Can use older computer systems:
  - Utilize old, outdated hardware
  - Backup server, file server, a router, etc...

Distributions

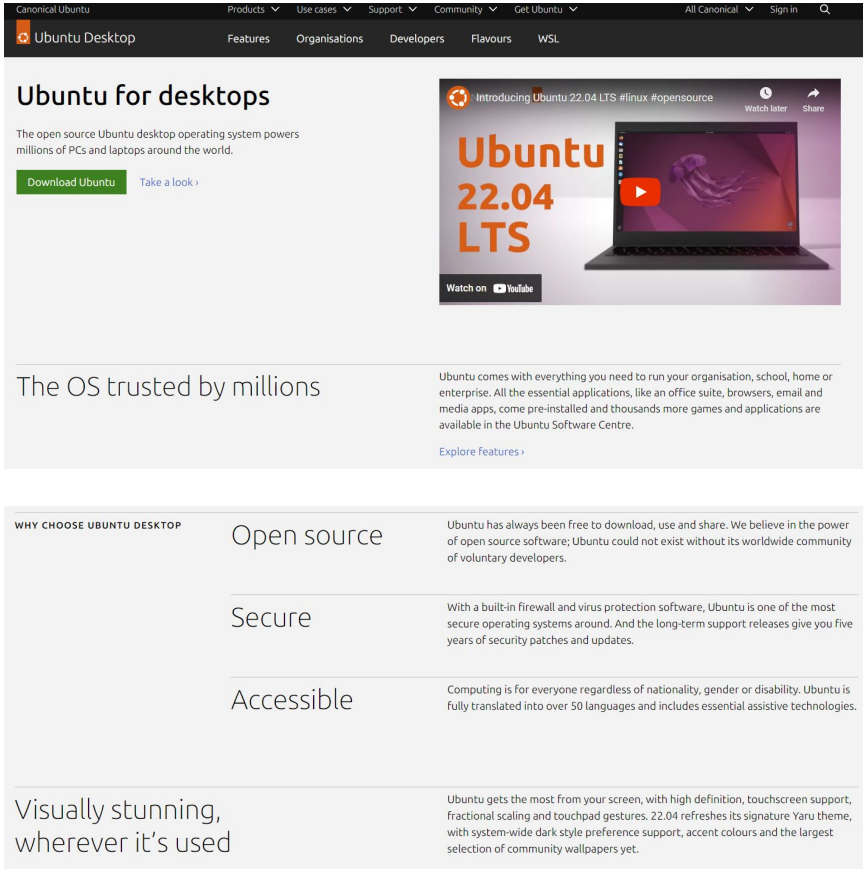
- Lots of Linux distributions
- Different distros for different purposes
- Linux is a Unix-like operating environment
- Regardless of the distro, there is always:
  - Linux Kernel (Core, talks to underlying hardware: memory, CPU, etc.)
  - Default GNU software (Command utilities)
  - General utilities (text editors, etc.)
- This course will focus only with **Ubuntu**



https://distrowatch.com/dwres.php?resource=popularity

Ubuntu distribution

- Extremely popular
- 3 Editions:
  - Desktop: GUI desktop environment
  - Server: Command line only server
  - Core: For IoT, Raspberry PI, etc.

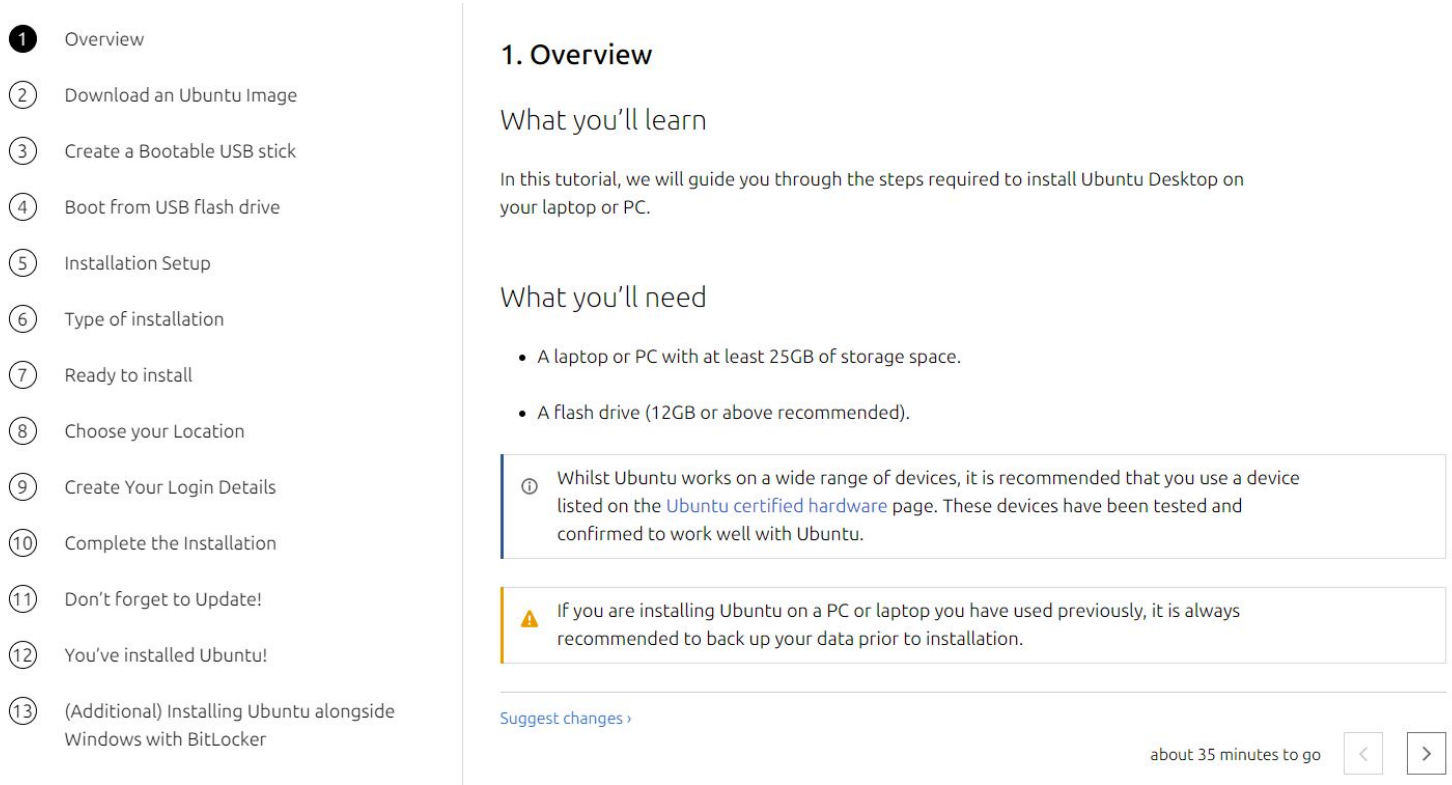


https://ubuntu.com/

Install Ubuntu Desktop on a computer

https://ubuntu.com/download/desktop

Requirements



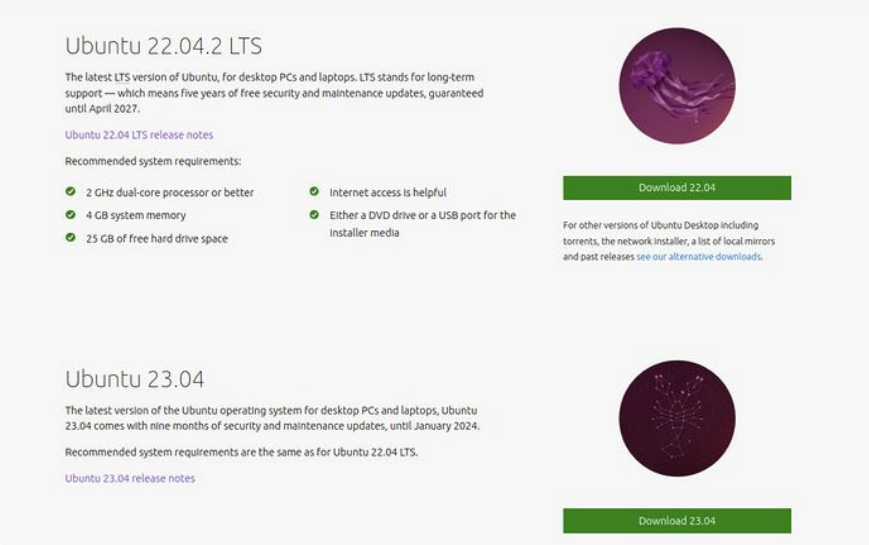
https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview

Ubuntu Image

2. Download an Ubuntu Image

You can download an Ubuntu image here. Make sure to save it to a memorable location on your PC! For this tutorial, we will use the Ubuntu 23.04 release which uses the new Ubuntu Desktop installer that will be included in all future Ubuntu releases.

If you are installing an older version of Ubuntu, such as Ubuntu 22.04 LTS, you will find that the visual presentation of the installer is different, but the overall flow should remain the same.

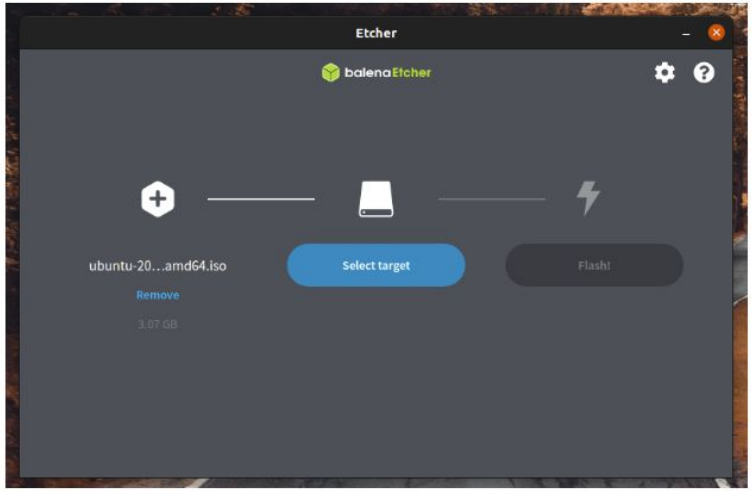


3. Create a Bootable USB stick

To install Ubuntu Desktop, you need to write your downloaded ISO to a USB stick to create the installation media. This is not the same as copying the ISO, and requires some bespoke software.

For this tutorial, we'll use balenaEtcher, as it runs on Linux, Windows and Mac OS. Choose the version that corresponds to your current operating system, download and install the tool.

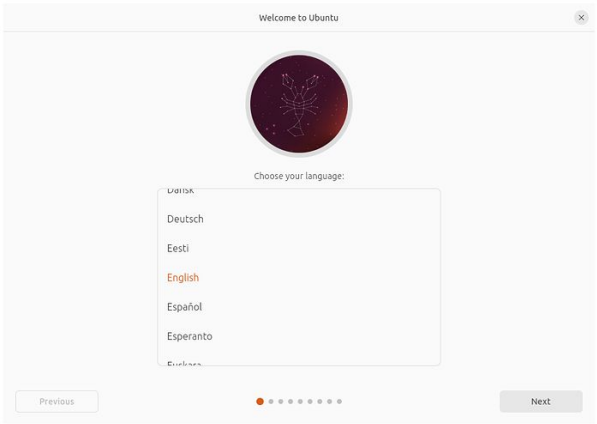
Select your downloaded ISO, choose your USB flash drive, and then click **Flash!** to install your image.

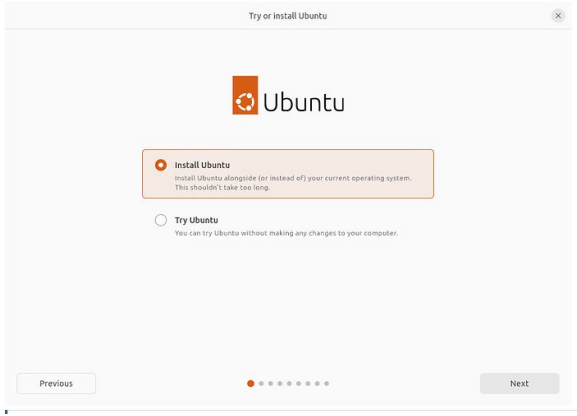


4. Boot from USB flash drive

Insert the USB flash drive into the laptop or PC you want to use to install Ubuntu and boot or restart the device. It should recognise the installation media automatically. If not, try holding F12 during startup and selecting the USB device from the system-specific boot menu.

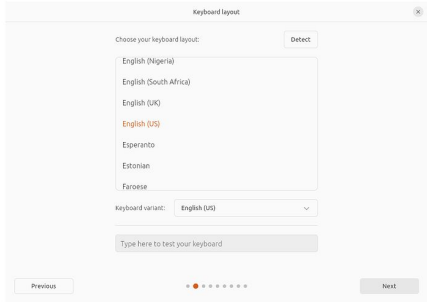
> F12 is the most common key for bringing up your system's boot menu, but Escape, F2 and F10 are common alternatives. If you're unsure, look for a brief message when your system starts – this will often inform you of which key to press to bring up the boot menu.





To proceed, click **Install Ubuntu**.

You will be asked to select your keyboard layout. Once you've chosen one, click **Continue**.

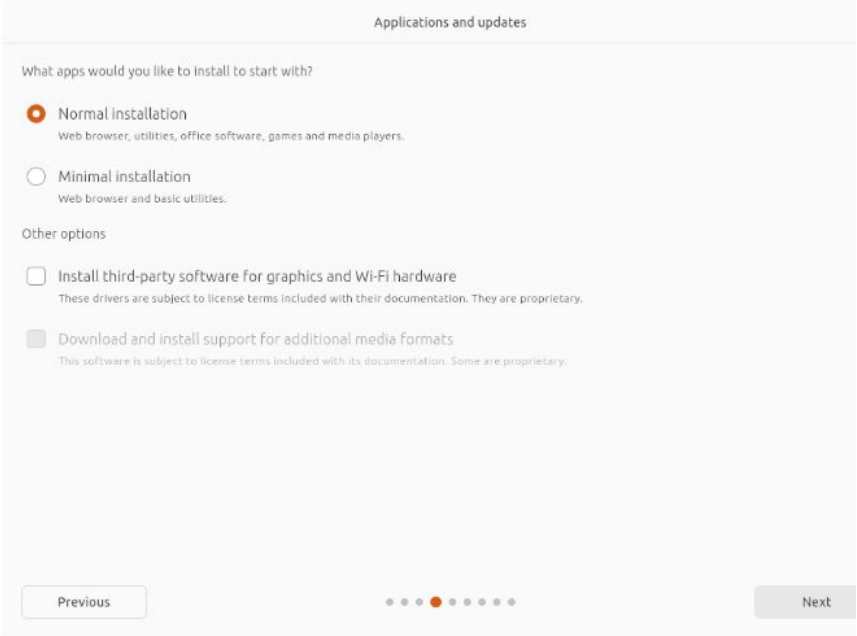


Next you will be asked to connect to wi-fi, this will allow Ubuntu to download updates and third party drivers (such as NVIDIA graphics drivers) during installation. Once you have connected to wi-fi (or chosen to proceed offline) then we can continue to the installation setup.

5. Installation Setup

You will be prompted to choose between the **Normal installation** and **Minimal installation** options. The minimal installation is useful for those with smaller hard drives or who don't require as many pre-installed applications.

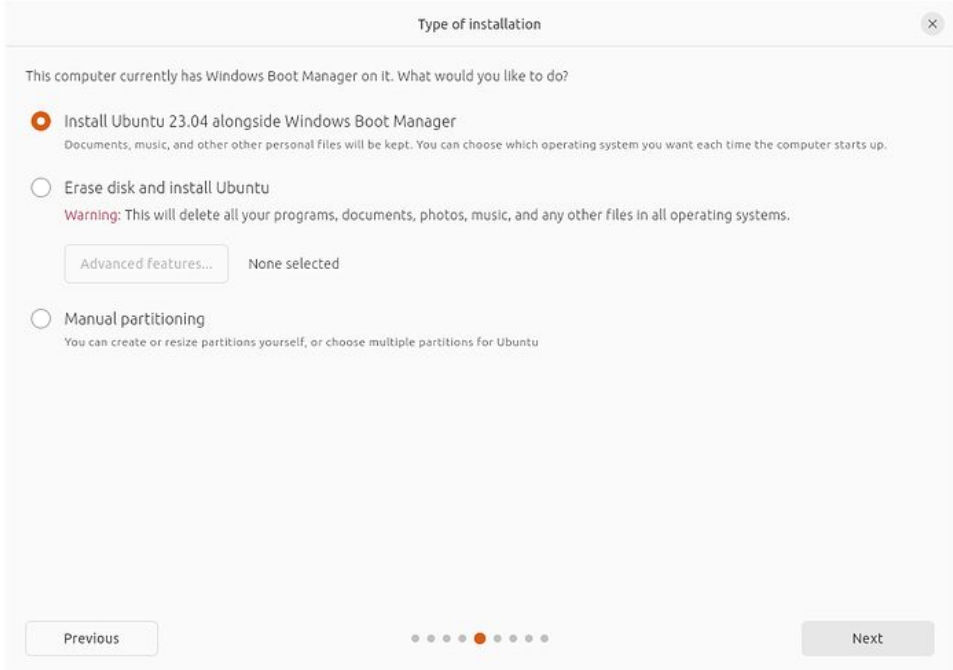
In **Other options**, you will be prompted to download updates as well as third-party software that may improve device support and performance (for example, Nvidia graphics drivers) during the installation. It is recommended to check both of these boxes.



6. Type of installation

This screen allows you to configure your installation. If you would like Ubuntu to be the only operating system on your hard drive, select **Erase disk and install Ubuntu**.

If your device currently has another operating system installed, you will receive additional options to install Ubuntu alongside that OS rather than replacing it.



Ubuntu Skype Screen Sharing Not Working

Solution?

https://www.tutsmake.com/ubuntu-skype-screen-sharing-not-working/

How to install vietnamese on ubuntu?

https://truongtc.com/cai-dat-bo-go-tieng-viet-ibus-bamboo-tren-ubuntu/

Command line interface (CLI) concept

Before diving into the world of linux command line, we need to learn two terms:

+ *The Shell*
+ *Terminal*

THE SHELL (CLI)

The shell is a program that takes keyboard commands and passes them to the operating system to carry out

TERMINAL

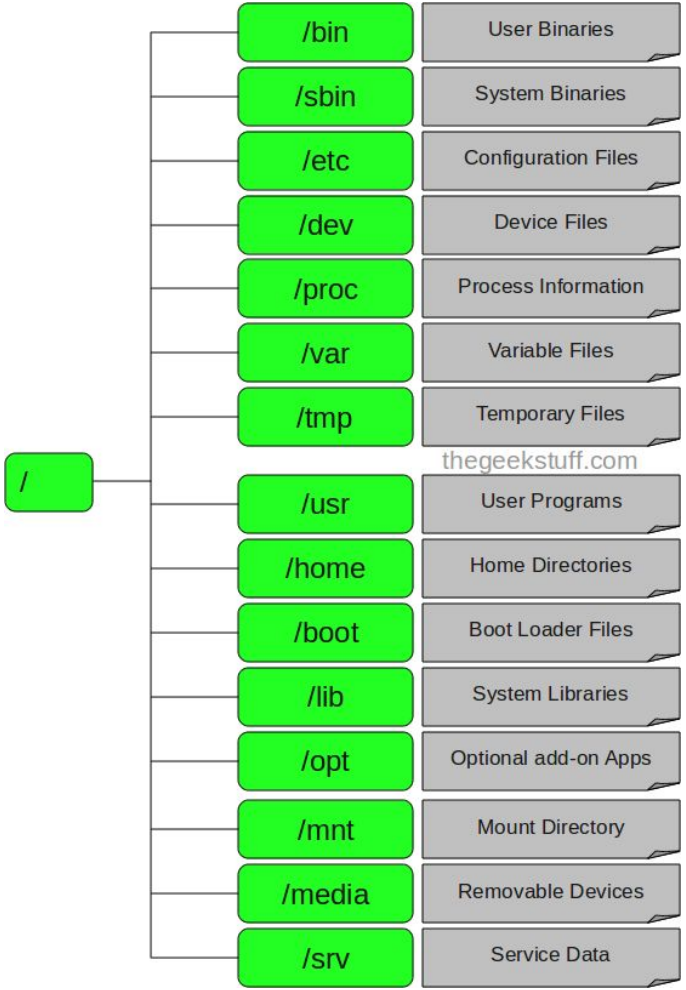
Terminal is a tool which you can use to pass your shell commands. This is a program that opens a window and lets you interact with the shell

Terminal

- Understanding the command line is essential
- Allows users to input text commands
- The Linux shell interprets your commands and pushes them into the kernel for execution
- Remember the terminal is Case-Sensitive!
- The terminal is available in server and desktop Linux distros
- “Tab” key to auto-complete commands

The Linux File System

- Logically organizes the files into directories (folders)



https://www.thegeekstuff.com/2010/09/linux-file-system-structure/

man **command**

- “Manual” for linux commands

Ex: man ls: manual page for ls command

- *To navigate and search:*
  - ➢ Ctrl-f: page down
  - ➢ Ctrl-b: page up
  - ➢ g: beginning of the manual
  - ➢ G: the end of the manual
  - ➢ /<string>: search manual
- *To display help:* h
- *To quit*: q

man -k

- man -k ifconfig

  Search all man files for ifconfig
- man -k “copy files”

  Search all man files for the string in quotes

Shortcuts

- Ctrl-L clears the terminal screen
- Up arrows: shows previous commands
- history: show the history of commands you have run
- Ctrl-A: moves the cursor to the beginning of the line
- Ctrl-E: moves the cursor the end of the line

Viewing command

- **less command**

The less command is a program to view text files

less command syntax

less filename
less /home/dminh/filename

How do I see the options?
man command
ex: man less

Command	Action
Page Up or b	Scroll back one page
Page Down or space	Scroll forward one page
Up Arrow	Scroll up one line
Down Arrow	Scroll down one line
G	Move to the end of the text file
1G or g	Move to the beginning of the text file
/*characters*	Search forward to the next occurrence of *characters*
n	Search for the next occurrence of the previous search
h	Display help screen
q	Quit less

Navigation command (1)

- **pwd command**

pwd (print working directory) command shows you in which directory you are currently at

Navigation command (2)

- **`cd` command syntax: cd [dir]**

cd command is going to land you in the current home directory

For example you want to go to somewhere:

cd */ change to the root directory*

cd .. *it means we want to go to the parent directory of the current working*

cd Documents *change to the Documents directory*

cd ***/home/dminh/Documents/*** *change to the Documents directory by absolute path*

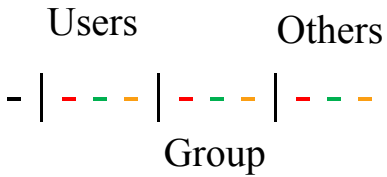
Navigation command (3)

- **ls command**

  To list the files and directories in the current working directory

  ls command syntax: ls [option] [fd]

  - ls -l *lists files in long form*



d - directory
r - read
w - write
x - execute

  - ls --all / ls –a *show all files including hidden files*
  - ls -lS *will sort directory by their size*
  - ls -la /var/ *combine both arguments*

Manipulating Files And Directories (1)

- **mkdir command**

We can use mkdir command to create directories in Linux

Syntax with mkdir command:

mkdir image *Create a directory called image*
mkdir image/picture *Create a subdirectory inside this image directory*
mkdir -p names/minh
mkdir --parents names/minh
Create a subdirectory for a directory that does not exist
mkdir --parents alphabets/{A,B,C} *Create multiple subdirectory*

Manipulating Files And Directories (2)

- **`rmdir` and `rm` command**

We can use rmdir and rm command to remove the directory or directory structures

Syntax with mkdir command: rmdir -options directory/rm -options directory

rmdir <dir> *Remove a directory thanh*

rmdir -p names/minh/duy/phu *Remove a directory structures*

rmdir -pv names/minh/duy/phu *Remove a directory structures and tell us extended information*

rm -r minh *Remove a directory minh and subdirectory inside*

Manipulating Files And Directories (3)

- **cp command**

We can use cp command to copies files or directories.

Syntax with cp command: cp options source destination

cp file1.txt file2.txt *Copy a file1 into a new file called file2 and copy the content*

cp file1.txt directory1 *Copy a file1 to a directory1*

cp file1.txt file2.txt directory2 *Copy a multiple file to a directory2*

cp -r : copy directory to directory

How to transfer these two files to directory1 one but we already know that file1.txt is already existing inside the directory1?

cp -i file1.txt file2.txt directory1

Manipulating Files And Directories (4)

- **mv command**

  The mv command performs both file moving and file renaming, depending on how it is used. In either case, the original filename no longer exists after the operation. mv is used in much the same way as cp

Syntax with mv command: mv options source destination

```
mv item1 item2
```

to move or rename file or directory “item1” to “item2” or:

```
mv item... directory
```

to move one or more items from one directory to another.

Manipulating Files And Directories (5)

- **df -h**
  - Report free disk space
  - -h makes the output human readable

Working with files/Redirection command (1)

- **cat command**

  cat is one of the most frequently used command on linux it has three related function with regard to text files.
  - *Displaying the text*
  - *Combining copies of text files*
  - *Creating new text*

Syntax: cat [option] [filename1 filename2 ....]

```
With no FILE, or when FILE is -, read standard input.

-A, --show-all
       equivalent to -vET

-b, --number-nonblank
       number nonempty output lines, overrides -n

-e     equivalent to -vE

-E, --show-ends
       display $ at end of each line

-n, --number
       number all output lines

-s, --squeeze-blank
       suppress repeated empty output lines

-t     equivalent to -vT

-T, --show-tabs
       display TAB characters as ^I

-u     (ignored)

-v, --show-nonprinting
       use ^ and M- notation, except for LFD and TAB

--help display this help and exit

--version
       output version information and exit
```

Working with files/Redirection command (2)

- **cat command**

Redirection in Linux: redirection simply means capturing output from a file command or program and sending it as an input to another file command or program

Syntax: [output] > [file]
Syntax with cat command:
cat > [file] *create a file and then transferring our you know stream to this file, now whatever you write here then this content will be written to the file*

cat >> [file] appending to the existing content of a file

Working with files/Redirection command (3)

- **touch command**

touch command is the easiest way to **create new empty files** in Linux. It is also used to **change the timestamps** on existing files or directories.

touch command syntax

touch filename

- **nano command**

nano is a small and friendly text editor and beside the text editing nano offers many extra features like an interactive, search or replace,...

Root

- There is always a root user on any Linux system
- There is only **one** root user
- The root user has absolute power over the system
- Don’t use root for day to day tasks because you have the ability to cause serious harm to the system

Root commands (1)

- sudo su
  - Allows a user to run commands as root
  - The user must have permissions to do this
  - The password required is the password for this user
- id
  - Show the user you are logged in as and the groups you are part of
- Exit
  - Exit out of sudo

Root commands (2)

sudo command

super user command allows you to execute some command with the superuser or perform some tasks most probably it will ask you for this super user privileges

sudo command syntax

- sudo [option]
- example: sudo apt-get update
  - sudo mkdir minh (*ex: root directory)*
  - sudo -s
  - sudo su

Command for Searching & Files Permissions (1)

chmod command (chmod – Change File Mode)

To change the mode (permissions) of a file or directory

chmod supports two distinct ways of specifying mode changes: **octal number representation, or symbolic representation.**

chmod command syntax

chmod *option* filename1

Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Command for Searching & Files Permissions (2)

chmod command (chmod – Change File Mode)

Symbol	Meaning
u	Short for “user” but means the file or directory owner.
g	Group owner.
o	Short for “others,” but means world.
a	Short for “all.” The combination of “u”, “g”, and “o”.

Notation	Meaning
u+x	Add execute permission for the owner.
u-x	Remove execute permission from the owner.
+x	Add execute permission for the owner, group, and world. Equivalent to a+x.
o-rw	Remove the read and write permission from anyone besides the owner and group owner.
go=rw	Set the group owner and anyone besides the owner to have read and write permission. If either the group owner or world previously had execute permissions, they are removed.
u+x,go=rx	Add execute permission for the owner and set the permissions for the group and others to read and execute. Multiple specifications may be separated by commas.

Processes Command

top command

top command provides a dynamic real-time view of your running system

The time which is the current

Stands for process ID

```
thanhthanh@Pavilion:~$ top

top - 22:24:12 up 1 day, 21:27,  1 user,  load average: 1,38, 1,32, 1,07
Tasks: 309 total,   2 running, 307 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5,3 us,  1,5 sy,  0,0 ni, 93,2 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  15888,1 total,   4961,1 free,   3715,6 used,   7211,4 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.  10608,9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2209 thanhth+  20   0 5899452 346388 115556 R  28,2   2,1  55:41.61 gnome-shell
   5042 thanhth+  20   0 1131,0g 472048 148196 S  16,5   2,9  81:37.53 chrome
   1905 thanhth+  20   0 1593812 141808  63900 S   4,9   0,9  33:49.54 Xorg
   4109 thanhth+  20   0   32,9g 207020 123700 S   2,9   1,3  50:03.54 chrome
  43575 thanhth+  20   0   24872   4680   3708 R   2,9   0,0   0:09.81 top
     13 root      20   0       0      0      0 S   1,0   0,0   1:57.35 ksoftirqd/0
     14 root      20   0       0      0      0 I   1,0   0,0   1:02.77 rcu_sched
   2854 thanhth+  20   0  561592  14740  12248 S   1,0   0,1   0:03.45 xdg-desktop-por
   5376 thanhth+  20   0   36,6g 147452 107340 S   1,0   0,9   2:37.53 rstudio
   5456 thanhth+  20   0  754512 193308  76472 S   1,0   1,2   4:11.39 rsession
  33351 thanhth+  20   0   36,8g 169864 113380 S   1,0   1,0   2:15.44 skypeforlinux
  34090 thanhth+  20   0 1153540 425812 206076 S   1,0   2,6   4:53.96 wpp
  34235 thanhth+  20   0  911924  61160  43244 S   1,0   0,4   0:29.64 gnome-terminal-
  42964 root       0 -20       0      0      0 D   1,0   0,0   0:02.18 kworker/u17:1+i915_flip
  43372 root      20   0       0      0      0 I   1,0   0,0   0:04.60 kworker/0:0-events
      1 root      20   0  168200  13384   8072 S   0,0   0,1   0:07.68 systemd
      2 root      20   0       0      0      0 S   0,0   0,0   0:00.07 kthreadd
      3 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 netns
      9 root       0 -20       0      0      0 I   0,0   0,0   0:03.81 kworker/0:1H-events_highpri
     10 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0,0   0,0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0,0   0,0   0:00.00 rcu_tasks_trace
     15 root      rt   0       0      0      0 S   0,0   0,0   0:00.34 migration/0
     16 root     -51   0       0      0      0 S   0,0   0,0   0:00.00 idle_inject/0
     18 root      20   0       0      0      0 S   0,0   0,0   0:00.00 cpuhp/0
     19 root      20   0       0      0      0 S   0,0   0,0   0:00.00 cpuhp/1
     20 root     -51   0       0      0      0 S   0,0   0,0   0:00.00 idle_inject/1
     21 root      rt   0       0      0      0 S   0,0   0,0   0:00.57 migration/1
     22 root      20   0       0      0      0 S   0,0   0,0   0:00.96 ksoftirqd/1
     24 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker/1:0H-events_highpri
     25 root      20   0       0      0      0 S   0,0   0,0   0:00.00 cpuhp/2
     26 root     -51   0       0      0      0 S   0,0   0,0   0:00.00 idle_inject/2
     27 root      rt   0       0      0      0 S   0,0   0,0   0:00.59 migration/2
```

The name of the program which is running currently

Key option in top

s Change time refreshing of the data

i Just filter out any idle processes

k Kill a process for

locate command

Locate a file, just like the search commands in Windows

- **apt install plocate**
  - installs the package for locate
- **locate cloud-init.log**
  - Displays directory containing cloud-init.log
- **locate -i CLOUD-init.log**
  - Locate regardless of case

find command

Similar to locate, but can be focused on a specific directory

- **find /var -name *.log**
  - Searches within /var and subdirectories

grep command (1)

- Search a text file or the output of a command
- Prints out lines that contain the pattern you searched for

grep user /etc/ssh/ssh_config

Display any lines that include user

grep -i “COMMAND LINE” /etc/ssh/ssh_config

-i option: Ignore upper/lower case

grep command (2)

- **grep -R <string> /etc/**

  Search all files in the etc directory

- **grep “user” /etc/ssh/ssh_config > sample.txt**

  Sends command output to a text file

- **ls | grep <string>**

  Search the output of a command for a string

head command

- View the first few lines of any text file
- Default: first 10 lines of the file
- -n: specify the number of lines
- Syntax: head -n <int> “/path/to/file”

tail command

- View the last few lines of any text file
- Default: last 10 lines of the file
- -n: specify the number of lines
- Syntax: tail -n <int> “/path/to/file”

echo command

- Print <string> to the terminal
- Add text to a file
- Syntax: echo <string>

diff command

- Compares the contents of two files
- Syntax: diff <file1.txt> <file2.txt>

ping command

- ping 8.8.8.8

Ctrl-C to stop

- ping **-c** 3 8.8.8.8

Ping a certain of times

ifconfig command

- Install the net-tools package

  sudo apt install net-tools

- Display network interface configurations

  ifconfig

netstat command

- Display the route table

  netstat -r

- Display open connections for a specific port

  Netstat -np | grep “80”

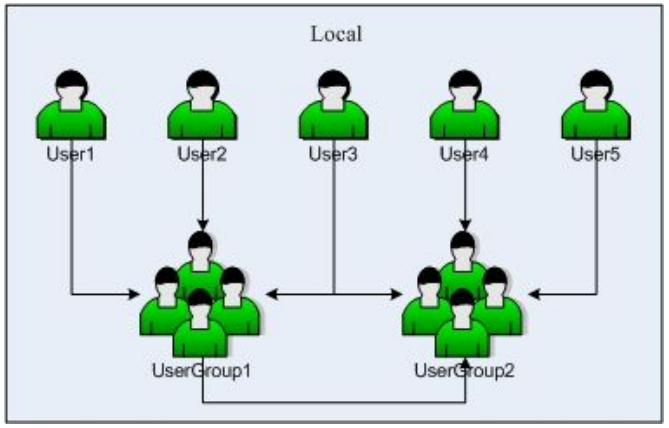
hostname command

- Name of the hosting network
- Display the IP address
  - hostname -i

1. Users and Groups

useradd command

useradd is used to add user accounts to your system.
Syntax : **useradd** [options] [name_of_the_user]




```
root@5aa33498a11b:/# ls /home/
root@5aa33498a11b:/# useradd -m dminh
root@5aa33498a11b:/# ls /home/
dminh
root@5aa33498a11b:/#
```

Some options for useradd

- To give a home directory path for new user: **-m** (--create-home)
- To create a user with a specific login group: **-g** (--gid)
- To create a user with a comment: **-c** (--comment)
- To set a password of the new account: **-p** (--password)
- See more with **useradd -h or useradd --help**

```
root@5aa33498a11b:/# useradd PBCM -m -g users -c "comment"
root@5aa33498a11b:/# ls /home/
PBCM  dminh
root@5aa33498a11b:/# ls /home/ -lah
total 16K
drwxr-xr-x 1 root  root  4.0K Jun 22 16:59 .
drwxr-xr-x 1 root  root  4.0K Jun 22 16:55 ..
drwxr-x--- 2 PBCM  users 4.0K Jun 22 16:59 PBCM
drwxr-x--- 2 dminh dminh 4.0K Jun 22 16:55 dminh
```

```
root@5aa33498a11b:/# passwd
New password:
Retype new password:
passwd: password updated successfully
```

userdel command

The **userdel** command removes the user account and deleting all the entries which refer to the username LOGIN.

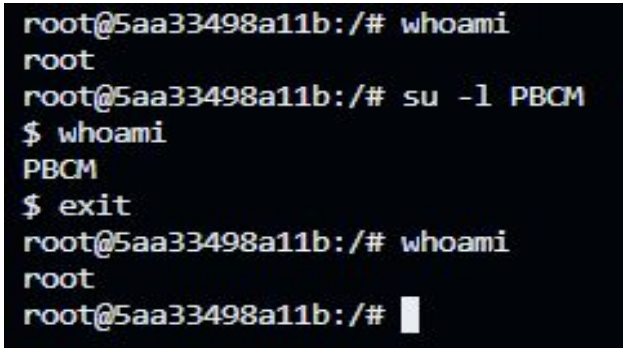
Syntax: **userdel** [options] LOGIN

```
root@5aa33498a11b:/# userdel -h
Usage: userdel [options] LOGIN

Options:
  -f, --force                   force removal of files,
                                even if not owned by user
  -h, --help                    display this help message and exit
  -r, --remove                  remove home directory and mail spool
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
      --extrausers              Use the extra users database
  -Z, --selinux-user            remove any SELinux user mapping for the user
```

```
root@5aa33498a11b:/# ls /home/
PBCM  dminh
root@5aa33498a11b:/# userdel -r dminh
userdel: dminh mail spool (/var/mail/dminh) not found
root@5aa33498a11b:/# ls /home/
PBCM
root@5aa33498a11b:/#
```

su command

- '**su**' (short for substitute or switch user). Using '**su**' is the simplest way to switch to the administrative account in the current login session.
- The **su** command lets you switch the current user to any other user. If you need to run a command as a different (non-root) user, use the **–l** [username] option to specify the user account.



Linux prompt show a $, instead of the login name and path how to fix it

https://askubuntu.com/questions/388440/why-is-there-no-name-showing-at-the-command-line

Basic Group Management (groups, groupadd, groupdel)

- A group is a collection of users. Groups make it easy to manage users with the same security and access privileges. A user can be part of different groups. Group information is held in the **/etc/group** file

```
root@5aa33498a11b:/# groups
root
root@5aa33498a11b:/# cat /etc/group
```

groupadd command is used to create a new user group

```
root@5aa33498a11b:/# groupadd PBCM
```

tail /etc/group

```
root@5aa33498a11b:/# tail /etc/group
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
PBCM:x:1000:
```

- The file shows group information in the following format
- group_name : password : group-id : list-of-members

Set password for group

```
root@5aa33498a11b:/# gpasswd PBCM
Changing the password for group PBCM
New Password:
Re-enter new password:
root@5aa33498a11b:/#
```

groupdel command is used to delete a existing group

```
root@5aa33498a11b:~# groupdel PBCM
root@5aa33498a11b:~# groups
```

How to add an existing User to a Group

```
root@5aa33498a11b:~# useradd -m test
root@5aa33498a11b:~# usermod -a -G PBCM test
root@5aa33498a11b:~# tail /etc/group
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
PBCM:x:1000:test
test:x:1002:
```

2. Giving the information

whatis command

- “whatis” command is used to offer a one-line overview of command
  - Syntax: whatis [keyword]

```
@ducminhnguyenle →/workspaces/microbiome-PBCM (main) $ whatis ls
ls (1)               - list directory contents
@ducminhnguyenle →/workspaces/microbiome-PBCM (main) $
@ducminhnguyenle →/workspaces/microbiome-PBCM (main) $ whatis grep
grep (1)             - print lines that match patterns
```

- If you want to know the details about multiple commands simultaneously, enter all the names as input

```
@ducminhnguyenle →/workspaces/microbiome-PBCM (main) $ whatis ls
ls (1)               - list directory contents
@ducminhnguyenle →/workspaces/microbiome-PBCM (main) $ whatis ls grep cat which
ls (1)               - list directory contents
grep (1)             - print lines that match patterns
cat (1)              - concatenate files and print on the standard output
which (1)            - locate a command
```

man command

- ‘**man**’ command in Linux is used to display the user manual of any command that we can run on the terminal
- Syntax : man [COMMAND NAME]...

```
@ducminhnguyenle →/workspaces/microbiome-PBCM (main) $ man whatis
```

```
WHATIS(1)                          Manual pager utils                          WHATIS(1)

NAME
       whatis - display one-line manual page descriptions

SYNOPSIS
       whatis [-dlv?V] [-r|-w] [-s list] [-m system[,...]] [-M path] [-L locale] [-C file] name ...

DESCRIPTION
       Each manual page has a short description available within it.  whatis searches the manual page names and displays
       the manual page descriptions of any name matched.

       name may contain wildcards (-w) or be a regular expression (-r).  Using these options, it  may  be  necessary  to
       quote the name or escape (\) the special characters to stop the shell from interpreting them.

       index  databases  are  used during the search, and are updated by the mandb program.  Depending on your installa-
       tion, this may be run by a periodic cron job, or may need to be run manually after new manual pages have been in-
       stalled.  To produce an old style text whatis database from the relative index database, issue the command:

       whatis -M manpath -w '*' | sort > manpath/whatis

       where manpath is a manual page hierarchy such as /usr/man.
```

- **man -f** [COMMAND NAME] = **whatis** [COMMAND NAME]

https://www.geeksforgeeks.org/man-command-in-linux-with-examples/

3. Text editor

Create a script in linux

Script used to make typescript or record all the terminal activities

1. Create file hello.sh

```
@ducminhnguyenle →/workspaces/codespaces-blank $ nano hello.sh
```

2. In order for the script to execute, we must grant it permission

First line put #! (shebang) and the location of bash

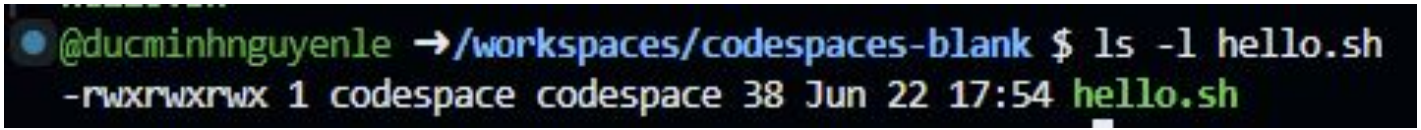
```
  GNU nano 6.2

#!/usr/bin/bash
echo "Hello everyone"
```

```
@ducminhnguyenle →/workspaces/codespaces-blank $ ls -lah hello.sh
-rw-rw-rw- 1 codespace codespace 38 Jun 22 17:54 hello.sh
@ducminhnguyenle →/workspaces/codespaces-blank $ chmod +x hello.sh
@ducminhnguyenle →/workspaces/codespaces-blank $ ls -lah
total 12K
drwxrwxrwx+ 2 codespace root      4.0K Jun 22 17:54 .
drwxr-xrwx+ 5 codespace root      4.0K Jun 22 15:30 ..
-rwxrwxrwx  1 codespace codespace   38 Jun 22 17:54 hello.sh
```

nano --> to save Ctrl+o (Enter); to exit Ctrl+x
or
exit existing files: Ctrl+x (Exit) → y (Yes to save) → Enter)

- In Linux, each file is associated with an owner and a group and assigned with permission access rights for three different classes of users:
  - The file owner.
  - The group members.
  - Others (everybody else).
- There are three file permissions types that apply to each class:
  - The read permission.
  - The write permission.
  - The execute permission.

```
Output
-rw-r--r-- 12 linuxize users 12.0K Apr  8 20:51 filename.txt
|[-][-][-]-   [------] [---]
| |  |  | |      |       |
| |  |  | |      |       +-----------> 7. Group
| |  |  | |      +-------------------> 6. Owner
| |  |  | +--------------------------> 5. Alternate Access Method
| |  |  +----------------------------> 4. Others Permissions
| |  +-------------------------------> 3. Group Permissions
| +----------------------------------> 2. Owner Permissions
+------------------------------------> 1. File Type
```

https://www.digitalocean.com/community/tutorials/an-introduction-to-linux-permissions

vim command

- ‘**vim’** is an editor to create or edit a text file using linux terminal.
- Operating modes in vim editor:
  - *Command mode*: by default, used to copy, paste, delete, or move text.
  - *Insert mode*: To write the contents in the file, we must go to insert mode. Press ‘I’ to go to insert mode. If we want to go back to command mode, press the [Esc] key.
  - *Syntax*: vim [filename]

- **Step 1**: Create a new file with ‘vim’ command
- **Step 2**: Go to Insert Mode
- **Step 3**: Write the content
- **Step 4**: Save the file and exit from the editor: To save the file and exit from it, press the [Esc] key and the ‘:wq’.
- **Step 5**: Check the data has been created successfully or not

4. Viewing Resources

“**df** ” command report the amount of "available disk space" being used by your filesystem

```
@ducminhnguyenle →/workspaces/codespaces-blank $ df
Filesystem     1K-blocks     Used Available Use% Mounted on
overlay         32847680 10835464  20318116  35% /
tmpfs              65536        0     65536   0% /dev
shm                65536        0     65536   0% /dev/shm
/dev/root       30298176 24390556   5891236  81% /vscode
/dev/sdb1       46127956      104  43752276   1% /tmp
/dev/loop3      32847680 10835464  20318116  35% /workspaces
```

So for making it "Human-readable" use "**-h**" flag with this command. Just write "**df -h**"

```
@ducminhnguyenle →/workspaces/codespaces-blank $ df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          32G   11G   20G  35% /
tmpfs            64M     0   64M   0% /dev
shm              64M     0   64M   0% /dev/shm
/dev/root        29G   24G  5.7G  81% /vscode
/dev/sdb1        44G  104K   42G   1% /tmp
/dev/loop3       32G   11G   20G  35% /workspaces
```

“**du**” command is used to estimate and display the disk space used by "files".

```
@ducminhnguyenle ➜~ $ du -h java/21.0.3-ms/
32K     java/21.0.3-ms/legal/jdk.crypto.ec
32K     java/21.0.3-ms/legal/jdk.charsets
32K     java/21.0.3-ms/legal/jdk.management
76K     java/21.0.3-ms/legal/java.desktop
32K     java/21.0.3-ms/legal/jdk.unsupported
32K     java/21.0.3-ms/legal/java.transaction.xa
32K     java/21.0.3-ms/legal/java.sql
32K     java/21.0.3-ms/legal/java.rmi
36K     java/21.0.3-ms/legal/jdk.dynalink
32K     java/21.0.3-ms/legal/jdk.jsobject
32K     java/21.0.3-ms/legal/jdk.management.agent
44K     java/21.0.3-ms/legal/java.xml.crypto
32K     java/21.0.3-ms/legal/jdk.httpserver
32K     java/21.0.3-ms/legal/java.management.rmi
32K     java/21.0.3-ms/legal/jdk.editpad
32K     java/21.0.3-ms/legal/java.logging
32K     java/21.0.3-ms/legal/jdk.sctp
32K     java/21.0.3-ms/legal/jdk.net
```

```
@ducminhnguyenle ➜~ $ du -h --max-depth=0
2.1G    .
```

```
@ducminhnguyenle ➜~ $ du -h --max-depth=1
15M     ./.oh-my-zsh
12K     ./.config
5.4M    ./.cache
1.6G    ./.local
480M    ./.vscode-remote
4.0K    ./.maven
4.0K    ./.python
4.0K    ./.minikube
4.0K    ./.nvs
4.0K    ./.hugo
8.0K    ./.jupyter
4.0K    ./.ruby
4.0K    ./.php
12K     ./.rbenv
8.0K    ./.conda
2.1G    .
```

“**free**” command displays the total amount of free and used physical and swap memory in the system as well as the buffer memory

```
@ducminhnguyenle →~ $ free
               total        used        free      shared  buff/cache   available
Mem:         8119872     1590616      247100       64656     6282156     6142188
Swap:              0           0           0
```

But this is also not human-readable so there are some flags. Which you can use with this **free** command and these flags are:

- "**-b**" for byte.
- "**-k**" for kilobyte.
- "**-m**" for megabyte.
- "**-g**" for gigabyte.
- "**-t**" for terabyte.

```
@ducminhnguyenle →~ $ free -m
               total        used        free      shared  buff/cache   available
Mem:            7929        1540         254          63        6135        6011
Swap:              0           0           0
@ducminhnguyenle →~ $ free -g
               total        used        free      shared  buff/cache   available
Mem:               7           1           0           0           5           5
Swap:              0           0           0
```

5. Basic file manipulation

Head and Tail Commands

- '**head**' command, print the top 10 number of data of the given input.
- Syntax: head [OPTION] [FILE]

```
-c, --bytes=[-]NUM       print the first NUM bytes of each file;
                           with the leading '-', print all but the last
                           NUM bytes of each file
-n, --lines=[-]NUM       print the first NUM lines instead of the first 10;
                           with the leading '-', print all but the last
                           NUM lines of each file
-q, --quiet, --silent    never print headers giving file names
-v, --verbose            always print headers giving file names
-z, --zero-terminated    line delimiter is NUL, not newline
    --help     display this help and exit
    --version  output version information and exit
```

'**-n**' option with 5 to print the first five lines

```
@ducminhnguyenle ➜~ $ head /etc/group -n 5
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
```

```
@ducminhnguyenle ➜~ $ head /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
```

'**-c**' option with 70 to print the first 70 characters

```
@ducminhnguyenle ➜~ $ head /etc/group -c 70
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
```

‘**tail**’ command reads the last 10 lines of the file.

- ‘**-n**’ option with 3 will display the last 3 lines

```
@ducminhnguyenle →~ $ tail /etc/group -n 3
oryx:x:991:codespace
python:x:990:codespace
pipx:x:989:codespace
```

- ‘**tail**’ command with 70 to print the last 70 characters

```
@ducminhnguyenle →~ $ tail /etc/group -c 70
pace
oryx:x:991:codespace
python:x:990:codespace
pipx:x:989:codespace
```

- ***Head*** *&* ***Tail*** *Command with Multiple Files*

```
@ducminhnguyenle →~ $ head -n 2 /etc/group /etc/fstab
==> /etc/group <==
root:x:0:
daemon:x:1:

==> /etc/fstab <==
# UNCONFIGURED FSTAB FOR BASE SYSTEM
```

wc command

‘wc’ used to find out number of lines, word count and characters count in each file

Syntax: wc [OPTION] [FILE]

```
Print newline, word, and byte counts for each FILE, and a total line if
more than one FILE is specified.  A word is a non-zero-length sequence of
characters delimited by white space.

With no FILE, or when FILE is -, read standard input.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.
  -c, --bytes            print the byte counts
  -m, --chars            print the character counts
  -l, --lines            print the newline counts
      --files0-from=F    read input from the files specified by
                           NUL-terminated names in file F;
                           If F is - then read names from standard input
  -L, --max-line-length  print the maximum display width
  -w, --words            print the word counts
      --help     display this help and exit
      --version  output version information and exit
```

```
@ducminhnguyenle →~ $ wc -l /etc/group
58 /etc/group
@ducminhnguyenle →~ $ wc -m /etc/group
852 /etc/group
@ducminhnguyenle →~ $ wc -w /etc/group
58 /etc/group
@ducminhnguyenle →~ $ wc -L /etc/group
23 /etc/group
```

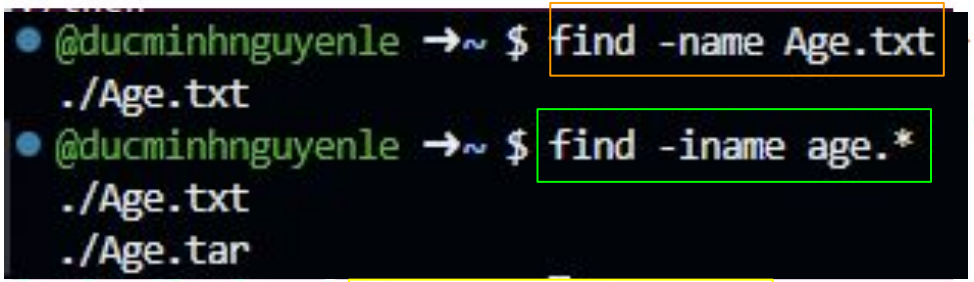
find command

‘find’ command is used to search for files or directories in a directory hierarchy

Syntax : find [path...] [options] [expression]

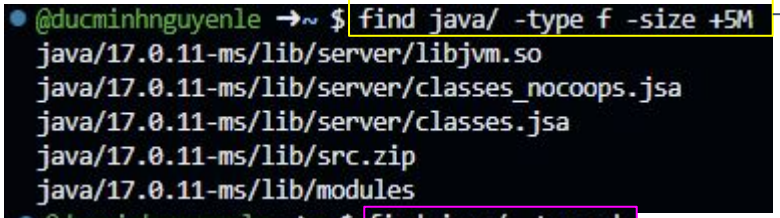
-name: Find a single file by name

-type: Find files based on their type

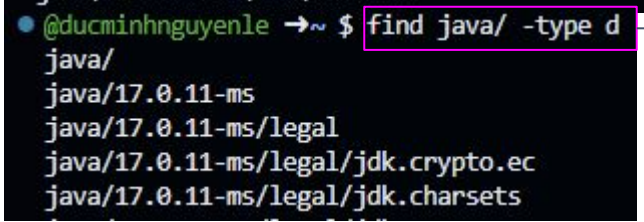


Find a single file by name

Find a single file by approximate name



Find a single file by type



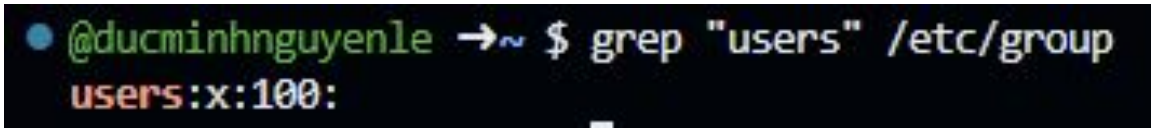
Find directory

grep command

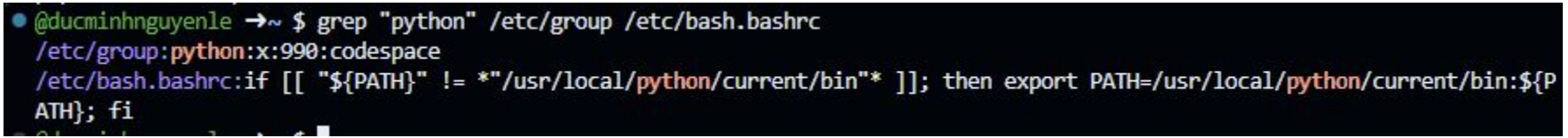
'grep' command used to search text and strings in a given file.

Syntax: grep [string] [filename(s)]

To print any line from a file



To search any line in multiple files



- grep commands are case sensitive, the terminal displays both uppercase and lowercase results.
grep -i [xxx] *
- grep can display count of lines where it finds a match for your word. grep -c [xxx] *

```
@ducminhnguyenle →~ $ grep --help
Usage: grep [OPTION]... PATTERNS [FILE]...
Search for PATTERNS in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNS can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
  -E, --extended-regexp     PATTERNS are extended regular expressions
  -F, --fixed-strings       PATTERNS are strings
  -G, --basic-regexp        PATTERNS are basic regular expressions
  -P, --perl-regexp         PATTERNS are Perl regular expressions
  -e, --regexp=PATTERNS     use PATTERNS for matching
  -f, --file=FILE           take PATTERNS from FILE
  -i, --ignore-case         ignore case distinctions in patterns and data
      --no-ignore-case      do not ignore case distinctions (default)
  -w, --word-regexp         match only whole words
  -x, --line-regexp         match only whole lines
  -z, --null-data           a data line ends in 0 byte, not newline
```

S

```
Miscellaneous:
  -s, --no-messages         suppress error messages
  -v, --invert-match        select non-matching lines
  -V, --version             display version information and exit
      --help                display this help text and exit

Output control:
  -m, --max-count=NUM       stop after NUM selected lines
  -b, --byte-offset         print the byte offset with output lines
  -n, --line-number         print line number with output lines
      --line-buffered       flush output on every line
  -H, --with-filename       print file name with output lines
  -h, --no-filename         suppress the file name prefix on output
      --label=LABEL         use LABEL as the standard input file name prefix
  -o, --only-matching       show only nonempty parts of lines that match
  -q, --quiet, --silent     suppress all normal output
      --binary-files=TYPE   assume that binary files are TYPE;
                            TYPE is 'binary', 'text', or 'without-match'
  -a, --text                equivalent to --binary-files=text
  -I                        equivalent to --binary-files=without-match
```

6. Compress and Extract Files

‘tar’ command

‘tar’ command is used to compress a group of files into an archive. The command is also used to extract, maintain, or modify tar archives.

Syntax: tar [flags] [destinationFileName] [sourceFileName]

Some flags to customize the command input:

-c Create a new archive.

-z zip, tells tar command that creates tar file using gzip

-v Display progress in the terminal while creating the archive, also known as “verbose” mode. .

-f Archive file name.

-x Extract from a compressed file.

To compress file

```
@ducminhnguyenle →~ $ tar -cvf Age.tar Age.txt
Age.txt
@ducminhnguyenle →~ $ ls
Age.tar  Age.txt  java  nvm
```

To extract file

```
@ducminhnguyenle →~ $ tar -xvf Age.tar
Age.txt
@ducminhnguyenle →~ $ ls
Age.tar  Age.txt  java  nvm
```

gzip command

‘gzip’ command compresses files. Each single file is compressed into a single file.

gzip [options] [file names]

To decompress a file: -d

```
@ducminhnguyenle ➜~ $ gzip Age.txt
@ducminhnguyenle ➜~ $ ls
Age.tar  Age.txt.gz  java  nvm
```

```
@ducminhnguyenle ➜~ $ gzip -d Age.txt.gz
@ducminhnguyenle ➜~ $ ls
Age.tar  Age.txt  java  nvm
```

One can unzip and open gz file using: gunzip [file.gz]

```
@ducminhnguyenle ➜~ $ ls
Age.tar  Age.txt.gz  java  nvm
@ducminhnguyenle ➜~ $ gunzip Age.txt.gz
@ducminhnguyenle ➜~ $ ls
Age.tar  Age.txt  java  nvm
```

bc command

‘bc’ command is used for command line calculator. It is similar to basic calculator by using which we can do basic mathematical calculations.

Syntax: bc [option] [expression]

The ‘bc’ command supports the following features:

- Arithmetic operators
- Increment or Decrement operators
- Assignment operators
- Comparison or Relational operators
- Logical or Boolean operators
- Math functions
- Conditional statements
- Iterative statements

```
@ducminhnguyenle →~ $ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2 + 3 *4
14
3 < 4
1
```

Ctrl + D or typing quit to exit

```
@ducminhnguyenle →~ $ echo "x=2; y=x*5; y" | bc
10
@ducminhnguyenle →~ $ bc <<< "x=2; y=x*5; y"
10
```

https://www.geeksforgeeks.org/bc-command-linux-examples/

7. Installing package

apt-get is a command-line tool which helps in handling packages in Linux. Its main task is to retrieve the information and packages from the authenticated sources for installation, upgrade and removal of packages along with their dependencies. Here APT stands for the *Advanced Packaging Tool*.

Some Used Commands with -apt-get

- **update**: to perform an update before you upgrade or dist-upgrade.

apt-get update

- **upgrade**: used to install the latest versions of the packages currently installed

apt-get upgrade

- **install**: used to install or upgrade packages.

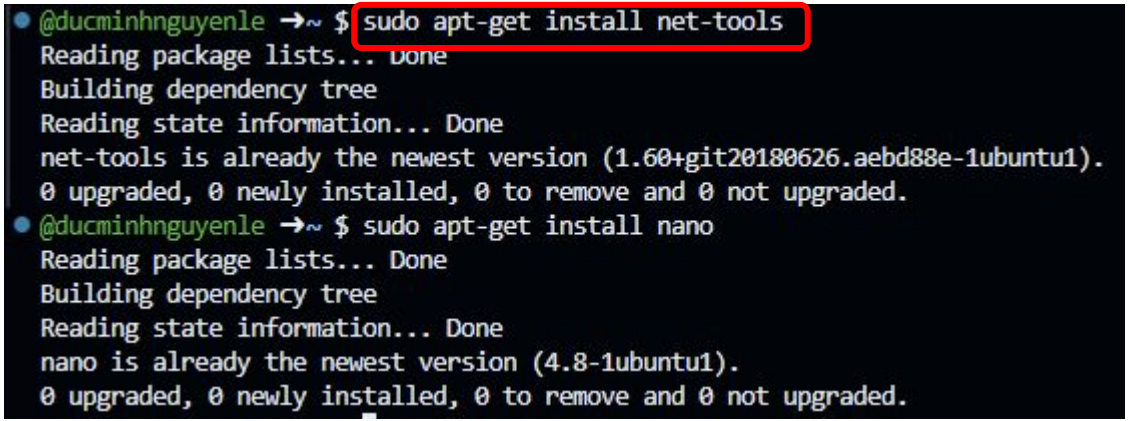
apt-get install [...PACKAGES]

- **remove:** used to remove package. It does not remove any configuration files created by the package.

apt-get remove [...PACKAGES]

- **purge**: removes the packages, and also removes any configuration files related to the packages. **apt-get purge [...PACKAGES]**
- **check**: This command is used to update the package cache and checks for broken dependencies **apt-get check**
- **download**: This command is used to download the given binary package in the current directory. **apt-get download [...PACKAGES]**

see more with: apt-get -h



To get files and documents

wget command

- ‘wget’ stands for web get. The wget is a free non-interactive file downloader command. Non-interactive means it can work in background when user is not logged in. This allows user to get disconnected with the system while wget finish its work.
- It supports HTTP, HTTPS, and FTP protocols

Syntax: wget [options] [url]

```
@ducminhnguyenle ➜~ $ wget "https://raw.githubusercontent.com/khuongduying/microbiome-PBCM/main/README.md?token=GHSAT0AAAAA
ACBBMBB3CI3VZZ6QZ3CIPBAUZTXEKOA" -O "README.md"
--2024-06-22 19:14:30--  https://raw.githubusercontent.com/khuongduying/microbiome-PBCM/main/README.md?token=GHSAT0AAAAAACBB
MBB3CI3VZZ6QZ3CIPBAUZTXEKOA
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 980 [text/plain]
Saving to: ‘README.md’

README.md                   100%[=====================================================>]     980  --.-KB/s    in 0s

2024-06-22 19:14:30 (56.8 MB/s) - ‘README.md’ saved [980/980]
```

some option use with 'wget'

- To save the downloaded file under a different name: **wget -O [fileName] [URL]**
- To resume a partially downloaded file: **wget -c [URL]**
- Download multiple files: **wget -i [fileName]** (file name is a .txt and contain URLs)
- Downloading and save a File to a Specific Directory : **wget -P [path]   [URL]**

curl command

- ‘curl’ (short for "Client URL") is a command line tool that enables data transfer over various network protocol
- Using any of the supported protocols (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP, or FILE).

Syntax: curl [options] [URL...]

```
@ducminhnguyenle ➜~ $ curl "https://raw.githubusercontent.com/khuongduying/microbiome-PBCM/main/README.md?token=GHSAT0AAAAA
ACBBMBB3CI3VZZ6QZ3CIPBAUZTXEKOA"
# PBCM

## Week 1

### Course overview - Introduction to Metagenomics

- Lecture: [Slides]()   [PDF](./01.Week_01/)

### Ubuntu and basic Linux commands

- Lecture: [Slides](https://docs.google.com/presentation/d/e/2PACX-1vTEmslVWH_NwaOs_VTtQy5eI4G4B_2lsv0hjrlo7sXyo0LihDtJCmctC
xSvuAgItgvqA2Na8K-aI88i/pub?start=false&loop=false&delayms=3000)  [PDF]()
- [Basic Linux commands overview](https://drive.google.com/file/d/1aJnb7SY3MZ6hHReuirwGVZfb3EPp9XXJ/view?usp=drive_link)
- Step-by-step installation: [Slides](https://docs.google.com/presentation/d/e/2PACX-1vT3trknqGoSubt2oxHYqV-jM429z6Jg1ILmIsd
eORU4u9mzwKAz5rwtEKP7RgdAaw/pub?start=true&loop=false&delayms=60000)  [PDF](./01.Week_01/)

### Introduction to bash scripting

- Lectures: [Slides](https://docs.google.com/presentation/d/e/2PACX-1vQYgv5jt-I4ztXF1osNQtH2lzTP0T21axACDloT8PcExN4j858SeUZu
OA0qv6tKsm04uQeqkzAquJ0d/pub?start=true&loop=false&delayms=60000)   [PDF](./01.Week_01/)
```

history command

'history' command is used to view the previously executed command

Basic syntax

```
@ducminhnguyenle ➜~ $ history
```

To show the limited number of commands that executed previously

```
@ducminhnguyenle ➜~ $ history 5
   98  rm README.md\?token\=GHSAT0AAAAAACBBMBB3CI3VZZ6QZ3CIPBAUZTXEKOA
   99  wget "https://raw.githubusercontent.com/khuongduying/microbiome-P
Z3CIPBAUZTXEKOA" -O "README.md"
  100  curl "https://raw.githubusercontent.com/khuongduying/microbiome-P
Z3CIPBAUZTXEKOA"
  101  history
  102  history 5
```

This command can also be used along with grep

```
@ducminhnguyenle ➜~ $ history | grep "ls"
    1  ls
    2  ls -l hello.sh
    9  ls
   11  ls
   17  ls
   18  ls nvm/
   22  ls
   23  ls java
```

Summary

Command line	Function
Users and Groups	
useradd	add user accounts to your system.
userdel	removes the user account
groupadd	create a new user group
groupdel	removes the user account
Giving the information	
whatis	offer a one-line overview of command
man	manual of any command

Summary

Command line	Function
Text editor	
nano	a small, free and friendly editor
vim	create or edit a text file
Viewing Resources	
df	report the amount of "available disk space" being used by filesystem
du	estimate and display the disk space used by "files"
free	displays the total amount of physical and swap memory as well as the buffer memory

Summary

Command line	Function
Basic file manipulation	
head	print the top 10 lines of data of the given input
tail	print the last 10 lines of the given input
wc	find out number of lines, word count and characters count in each file
find	used to search for files in a directory hierarchy
grep	used to search text and strings in a given file
Compress and Extract Files	
tar	used to compress a group of files into an archive
gzip	compresses files

Command line	Function
Installing package	
apt-get	install
To get files and documents	
wget, curl	download file from URL
history	view the previously executed command
bc	calculator

Thank you