

# SIRE511 : LINUX AND BIOINFORMATICS DATA SKILLS

## Fundamental Linux PART I

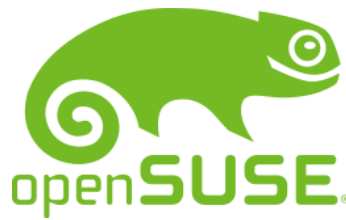
1st Week, 03/09/2024

Kwanrutai Mairiang, Ph.D

# Introduction to Linux

# What is Linux?

- Linux is an operating system based on Unix kernel, like macOS.
- Free and open-source software development and distribution.
- There are several distributions to serve different purposes.



# Why Linux is widely used in bioinformatics?

- Most open-source bioinformatics tools are command-line based and developed for Linux OS.
- Linux offers a variety of powerful and flexible commands that are useful for editing and managing biological files, especially sequencing results.
- Most web-applications are deployed on Linux.
- The container images (e.g., Docker image) are usually developed in Linux.
- HPC and cloud are usually run on Linux.



# Ubuntu

- Ubuntu is one of the most popular Linux distributions.
- Ubuntu can be installed on a computer, a virtual machine, or cloud computing.
- A British company called "Canonical" introduced Ubuntu in 2004. It was created based on "Debian," which was a popular distribution at that time.
- Ubuntu was created to be more user-friendly than Debian, which was difficult to install.
- Ubuntu works well for cloud computing, servers, desktops, and internet of things (IoT) devices.



# Ubuntu

## What makes Ubuntu so popular?

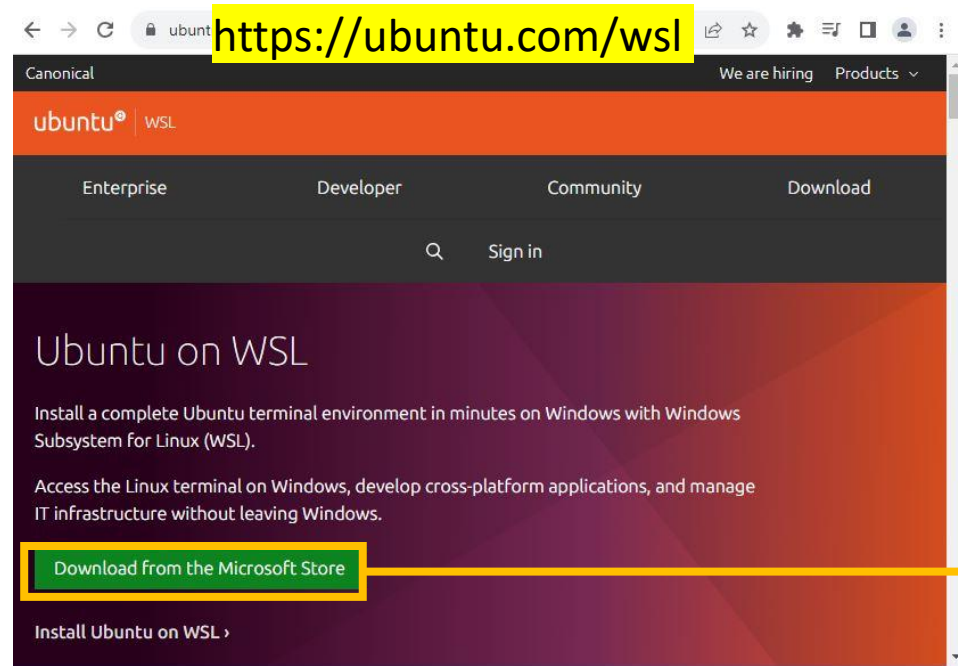
- User-friendly
- Strong security
- More software options
- Enhanced privacy
- Lightweight performance
- Free of charge

# Possible options to Use Ubuntu on Mac OS and Windows

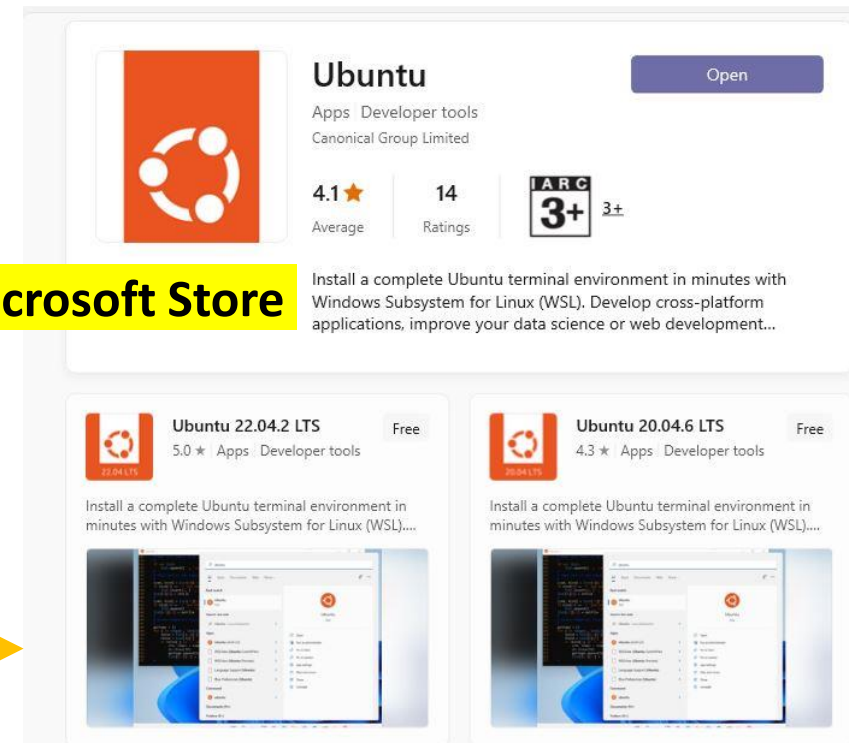
- Windows Subsystem for Linux (WSL)
  - Windows
- Virtual Machine
  - Windows, Mac OS
- Docker
  - Windows, Mac OS
- Create remote virtual server on Cloud, e.g., Amazon, Google Cloud, DigitalOcean.
  - Windows, Mac OS

# What is WSL?

- The Windows Subsystem for Linux enables developers to operate a Linux environment, including a wide range of command-line tools, utilities, and applications, directly within the Windows operating system.

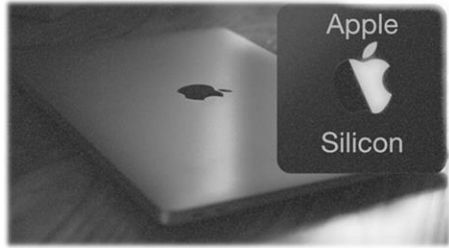


## Microsoft Store





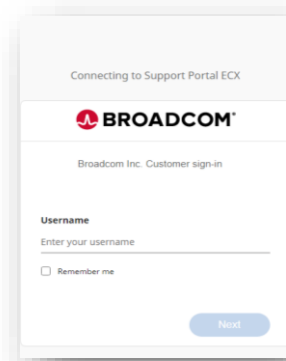
# Virtual Machine



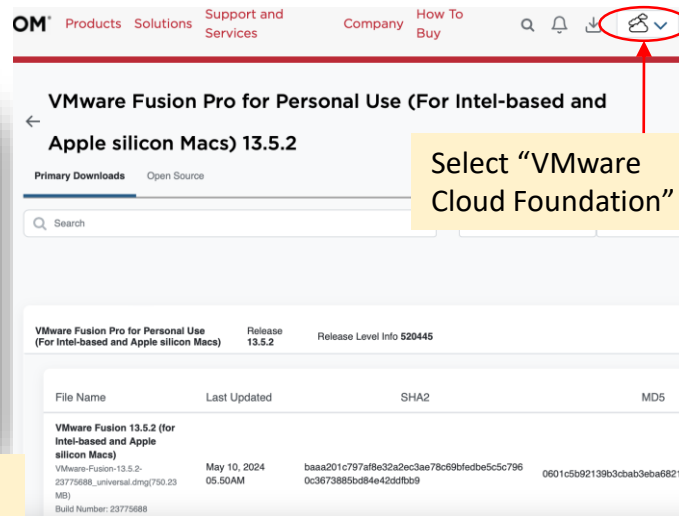
Apple Silicon M1 & M2

Download VMware Fusion Pro for Personal Use :  
<https://access.broadcom.com/default/ui/v1/signin/>

①



Register and login to BROADCAST



Select "VMware Cloud Foundation"

Download Ubuntu 22.04 LTS for 64-bit ARM:  
② <https://cdimage.ubuntu.com/jammy/daily-live/current/>

Ubuntu 22.04.3 LTS (Jammy Jellyfish) Daily Build

Select an image



Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure. Warning: This image is oversized (which is a bug) and will not fit onto a standard 703MiB CD. However, you may still test it using a DVD, a USB drive, or a virtual machine.

64-bit ARM (ARMv8/AArch64) desktop image

For 64-bit ARMv8 processors and above.

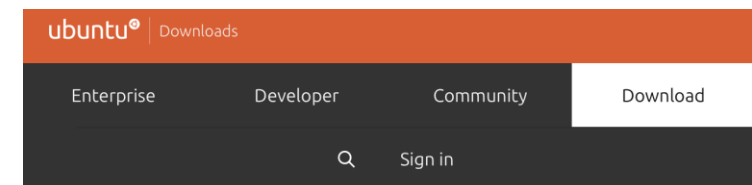


Microsoft Windows

① Download VirtualBox for Windows:  
<https://www.virtualbox.org/wiki/Downloads>



② Download Ubuntu 22.04 LTS for 64-bit AMD  
<https://ubuntu.com/download/desktop#download>



Ubuntu Desktop >

Download Ubuntu desktop and replace your current operating system whether it's Windows or Mac OS, or, run Ubuntu alongside it.

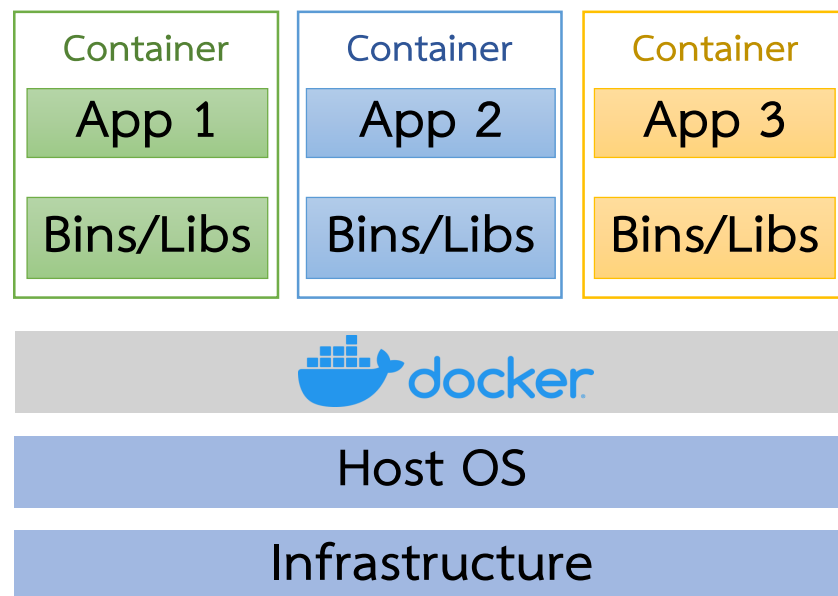
Get Ubuntu Desktop

Ubuntu Server >

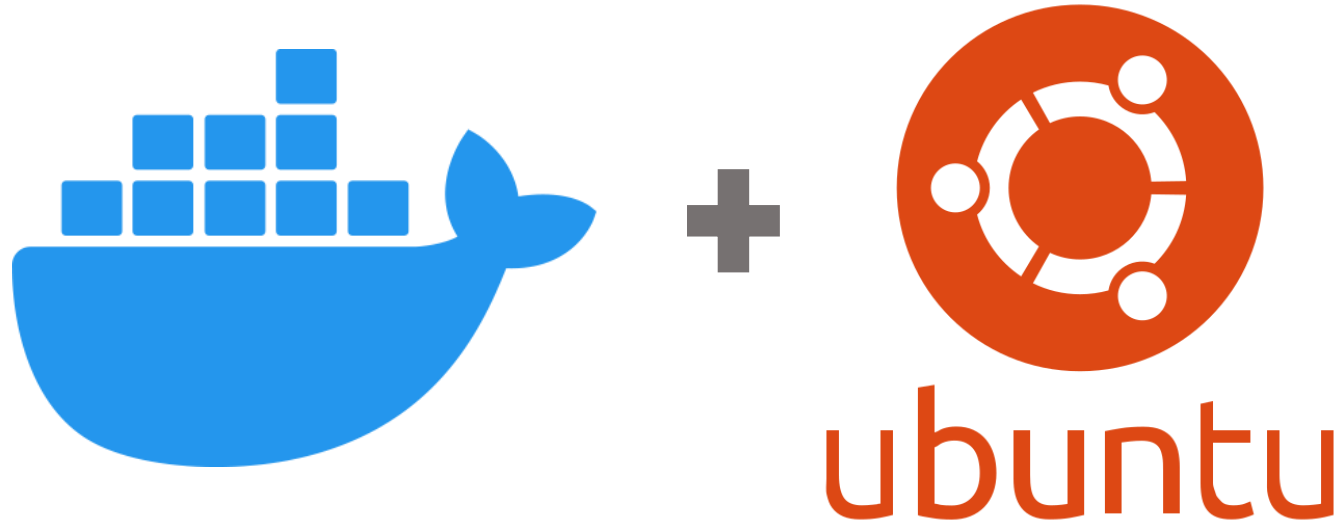
The most popular server Linux in the cloud and data centre, you can rely on Ubuntu Server and its five years of guaranteed free upgrades.

# What is docker?

- Docker is an open platform that allows you to develop, ship, and run applications in containers.
- Containers are lightweight, portable, and self-sufficient environments that enclose an application along with all of its dependencies, such as libraries, runtime, and system settings.
- Docker provides a way to package and distribute applications as containers, ensuring consistent behavior across different environments.



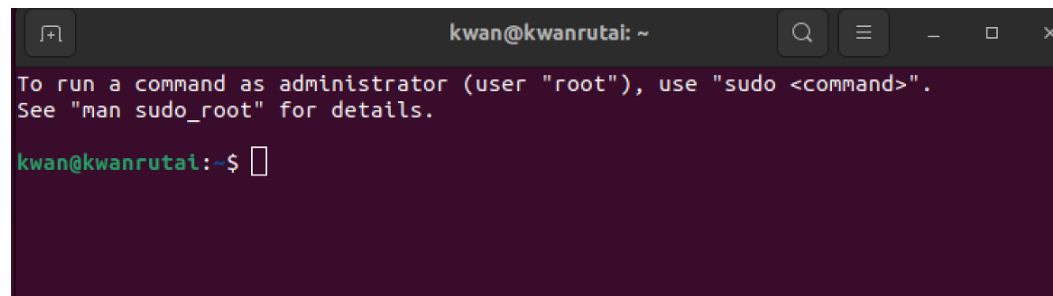
# Creating an Ubuntu container using Docker on Mac and Windows



# What is Shell, Terminal, and Command?

# Linux : Shell and Terminal

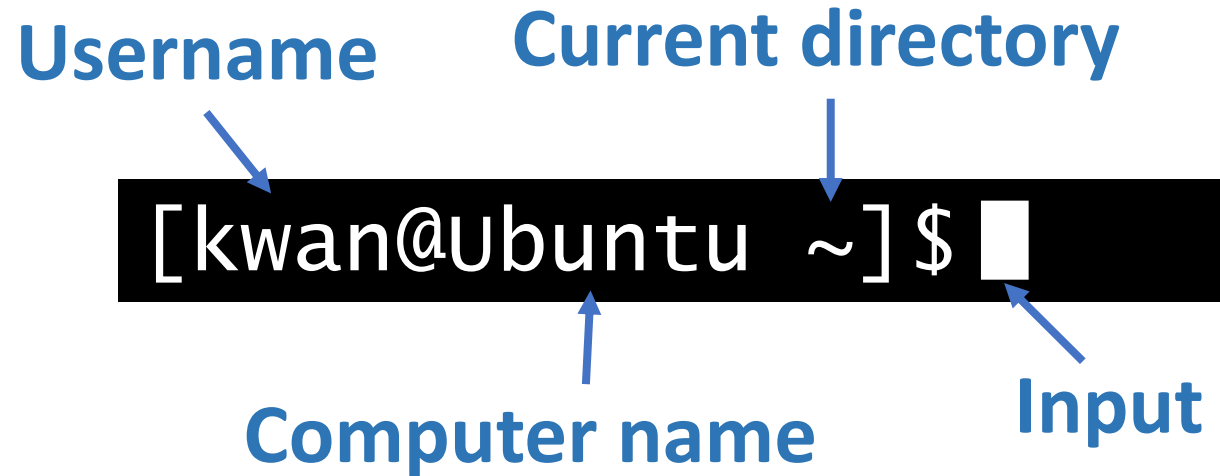
- **Shell** is a Linux command interpreter that provides:
  - Built-in commands
  - Programming control structures
  - Environment variables
  - Bash (Bourne-Again Shell) is a type of shell that is widely used in Unix/Linux systems.
    - Some examples of shell types are Bourne Shell (sh), C shell (csh), and Zsh (zsh).
- A **terminal** , a command-line interface (CLI), is an application that provides users with access to the system shell through text-based commands. It offers a visual interface for entering commands that are then interpreted into binary by shell.



```
kwan@kwanrutai: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
kwan@kwanrutai:~$
```

# Linux : command prompt

The command prompt serves as the input field within the command line interface.



The diagram shows a Linux command prompt on a black background. The prompt text is "[kwan@Ubuntu ~]\$". Four blue arrows point to different parts of the prompt: "Username" points to "kwan", "Current directory" points to "~", "Computer name" points to "Ubuntu", and "Input" points to the empty space after the dollar sign.

Username      Current directory

[kwan@Ubuntu ~]\$

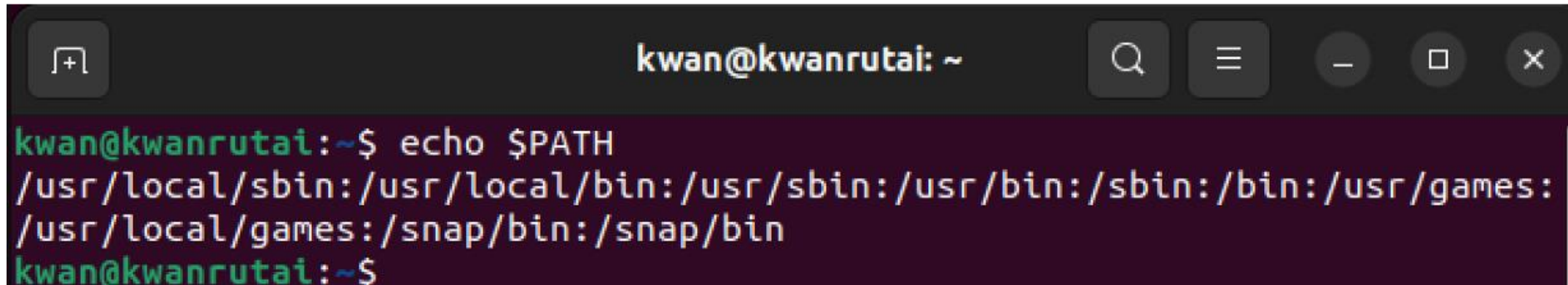
Computer name      Input

~ equal to the HOME directory

# How Shell Understands the entered command?

- All executable commands are located in PATH variable.
- To view all the folders located in the PATH variable, use the command:

```
echo $PATH
```

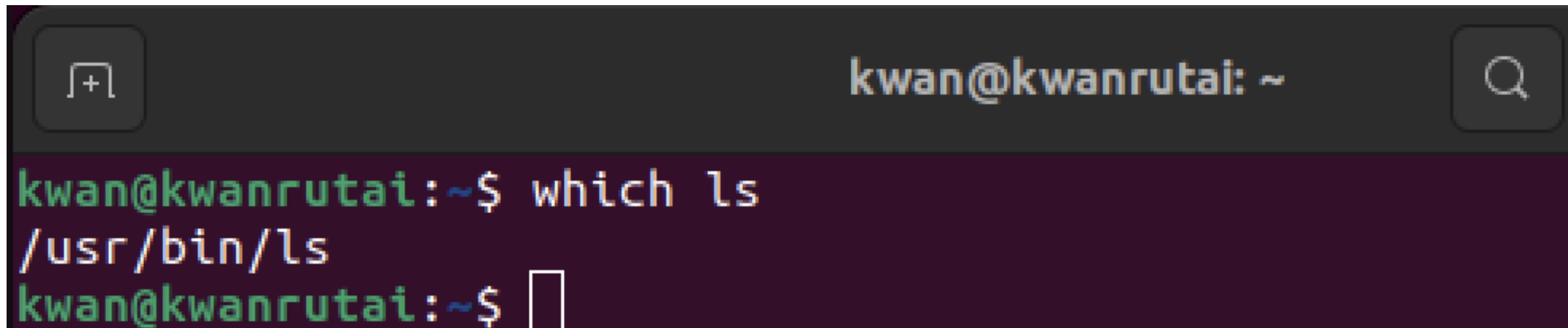
A screenshot of a terminal window with a dark background. The window title bar shows 'kwan@kwanrutai: ~' and standard window controls. The terminal text shows the command 'echo \$PATH' being entered and executed, resulting in a multi-line output of directory paths. The prompt 'kwan@kwanrutai:~\$' is visible before and after the command.

```
kwan@kwanrutai:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:
/usr/local/games:/snap/bin:/snap/bin
kwan@kwanrutai:~$
```

# How Shell Understands the entered command?

To locate the folder where a specific command is located, use the command:

**which** **command**



```
kwan@kwanrutai: ~  
kwan@kwanrutai:~$ which ls  
/usr/bin/ls  
kwan@kwanrutai:~$
```



# Linux Command

- A Linux command is a program or utility that operates within the command-line interface. The commands can be categorized as follows:
  - **System commands:** the commands used to perform tasks related to the system.
  - **File management commands:** the commands are used to manage files and directories.
  - **Networking commands:** the commands are used to manage and troubleshoot network connections.
  - **Process management commands:** the commands are used to manage processes running on the system.
  - **Shell built-in commands:** the commands are built into the shell and are used to perform tasks related to the shell.
  - **Advanced Linux commands:** these are more specialized commands used for advanced tasks such as system programming and network security.

# Example of Linux commands

Command	Function
<b>System Commands:</b>	
useradd	Create a new user account
userdel	Delete a user account
passwd	Change a user's password
shutdown	Shut down the system
reboot	Restart the system
<b>File Management Commands:</b>	
ls	List the files and directories in a directory
cd	Change the current directory
mkdir	create a new directory
rmdir	delete an empty directory
cp	copy a file
mv	move or rename a file
rm	Delete a file

Command	Function
<b>Networking Commands:</b>	
ifconfig	View and configure network interfaces
ping	Test connectivity to a network host
ssh	Remote host securely using the SSH protocol
<b>Process Management Commands:</b>	
ps	View a list of processes running on the system
top	Display a real-time view of the processes running on the system
kill	terminate a process
<b>Shell Built-in Commands:</b>	
echo	display a message on the terminal
alias	create aliases for other commands
history	view the command history

# Command for getting information about the linux computer

- The following commands have the capability to retrieve information about your Linux computer.

Commands	Function
uname	Print system information
uptime -p	Tell how long the system has been running
lscpu	Display information about CPU architecture
lshw	Report detail and brief information about hardware of the machine
lsblk	List information about available block devices (except RAM disk) in tree like format.
df	Display the amount of disk space available.
du	Estimate file space usage

# Getting help command...Man utility

## Manual pages

- All core programs will have a manual page to document the options for the command
- Manual pages are accessible using the man program followed by the program name you want to look up.
- All manual pages have a common structure

Open manual page:

The command "man" is used to open the manual page.



```
[kwan@Ubuntu ~]$ man ls
```

Command for accessing the manual.

# Manual page

There are three main parts.

## 1. NAME:

A short description of the command.

## 2. SYNOPSIS:

Command usage

## 3. DESCRIPTION:

Full description of all options and argument used the command.

Type “q” to exit manual page

```
LS(1)                                General Commands Manual                                LS(1)

NAME
    ls - list directory contents ①

SYNOPSIS
    ls [-@ABCFGHILOPRSTUWabcdefghiklmnopqrstuvwxy1%,] [--color=when] [-D format] [file ...] ②

DESCRIPTION
    ③ For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information. For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any requested, associated information.

    If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

    The following options are available:

    -@      Display extended attribute keys and sizes in long (-l) output.

    -A      Include directory entries whose names begin with a dot ('.') except for . and .. Automatically set for the super-user unless -I is specified.

    -B      Force printing of non-printable characters (as defined by ctype(3) and current locale settings) in file names as \xxx, where xxx is the numeric value of the character in octal. This option is not defined in IEEE Std 1003.1-2008 ("POSIX.1").

    -C      Force multi-column output; this is the default when output is to a terminal.

    -D format
        When printing in the long (-l) format, use format to format the date and time output. The argument format is a string used by strftime(3). Depending on the choice of format string, this may result in a different number of columns in the output. This option overrides the -T option. This option is not defined in IEEE Std 1003.1-2008
```

# Command Options and Arguments

```
[kwan@Ubuntu ~]$ command -options arguments
```

**Command** : command/program that operates within the CLI

**Options** : pass the parameters to a program that will change the way that command does

**Arguments** : Provides the input/output that the command interacts with. When program/command starts execution, arguments are used to pass values or files to it. Argument is input from user.

# Options (Switches)

- Options change the behavior of a program.
- Most options have both short and long formats
  - Single dashes (- options) = short format
  - Double dashes (--options) = long format
- Options are case sensitive
- The short format can be written in a combined form.

**Executing these  
commands**

Command	Options: Short format	Options: Long format
ls	-	-
ls	ls -a	ls --all
ls	ls -l	No long format available.
ls	ls -la , ls -l -a	ls -all -l

# Arguments

- Some commands can be executed without any arguments or options, e.g.,
  - `ls`, `top`, `htop`, `hostname`, `uname`
- Some commands require arguments to execute.
- Let's try running these commands :
  - **`mkdir`** : Create directories
  - **`cd`** : Change the current directory to DIR
  - **`rm`** : Remove file or directories

Open manual page:

```
[kwan@Ubuntu ~]$man cd
```



# Arguments for option

- Some options have their own arguments.(arguments for option)
  - Let's open manual page of command **date**

Try execute these command

```
[kwan@Ubuntu ~]$man date
```

```
date
```

```
date -d "19820208"
```

# Processes in Linux

# What are processes in Linux?

- Processes are tasks that Linux is currently executing after a user executes a command.
- The processes that start quickly and are stopped shortly afterward are referred to as “Short processes”.
  - ls, mkdir, rm, cd, ...
- The processes that continue running while you are working on Linux, such as the Bash shell process, are referred to as “Long processes”.

# List Running Processes

- Use ps command to report the information about current processes.

## Long running process:

Shell continues to run since it started working. Process IDs do not change.

## Short running process:

PS start and stop process after done its job. A new process ID is generated when starting a new process.

```
kwan@kwanrutai: ~  
kwan@kwanrutai:~$ ps  
  PID TTY          TIME CMD  
 1619 pts/0        00:00:00 bash  
 1666 pts/0        00:00:00 sh  
 1667 pts/0        00:00:00 bash  
 3499 pts/0        00:00:00 ps  
kwan@kwanrutai:~$ ps  
  PID TTY          TIME CMD  
 1619 pts/0        00:00:00 bash  
 1666 pts/0        00:00:00 sh  
 1667 pts/0        00:00:00 bash  
 3500 pts/0        00:00:00 ps  
kwan@kwanrutai:~$
```

1 First run

2 Second run

By default, ps report only processes running by current user.  
Try command ps with option -A or -e

# Starting addition process and killing processes

- Use TOP command to display linux processes.

**PID:** Process ID.

**USER:** The owner of the process.

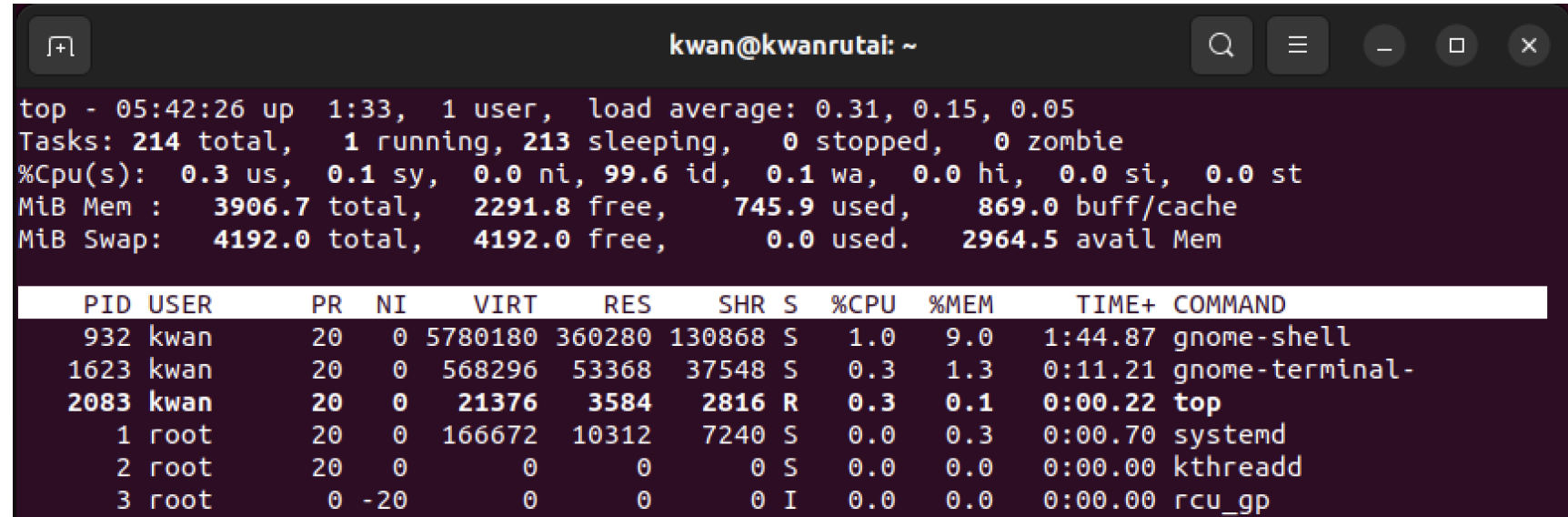
**S:** Status of the process.

**%CPU:** The share of CPU time used by the process since the last update.

**%MEM:** The share of physical memory used.

**TIME+:** Total CPU time used by the task in hundredths of a second.

**COMMAND:** The command name or command line (name + options).



```
top - 05:42:26 up 1:33, 1 user, load average: 0.31, 0.15, 0.05
Tasks: 214 total, 1 running, 213 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.1 sy, 0.0 ni, 99.6 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3906.7 total, 2291.8 free, 745.9 used, 869.0 buff/cache
MiB Swap: 4192.0 total, 4192.0 free, 0.0 used. 2964.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
932	kwan	20	0	5780180	360280	130868	S	1.0	9.0	1:44.87	gnome-shell
1623	kwan	20	0	568296	53368	37548	S	0.3	1.3	0:11.21	gnome-terminal-
2083	kwan	20	0	21376	3584	2816	R	0.3	0.1	0:00.22	top
1	root	20	0	166672	10312	7240	S	0.0	0.3	0:00.70	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp

## Status of process

**D:** Uninterruptible sleep

**R:** Running

**S:** Sleeping

**T:** Traced (stopped)

**Z:** Zombie

To terminate the process, execute the 'kill' command.

**Command**



```
[kwan@Ubuntu ~]$ kill PID
```

**Process ID**

# Htop command

- “htop” is a command-line tool that enables interactive real-time monitoring of system resources and server processes.
- Htop is more visually appealing utility compared to 'top', but it needs to be installed as it is not included by default in Ubuntu systems.
- Installing htop package using apt install

Let's install htop :

1. Update the package index files on the system. These files hold information about available packages and their versions.

```
sudo apt update
```

2. Install htop

```
sudo apt install htop
```

# Run htop

#Run htop in Ubuntu docker

Run htop: **htop**

```
kwan — root@2c4ea9ba5a04: /data — com.docker.cli ◀ docker start -a -i 2c4ea9ba5a04 — 99...

0[ | 0.7%] 3[ | 0.7%]
1[ 0.0%] 4[ 0.0%]
2[ 0.0%] Tasks: 2, 0 thr; 1 running
Mem[|||||] 786M/7.67G Load average: 0.00 0.02 0.03
Swp[ 0K/1024M] Uptime: 00:14:27

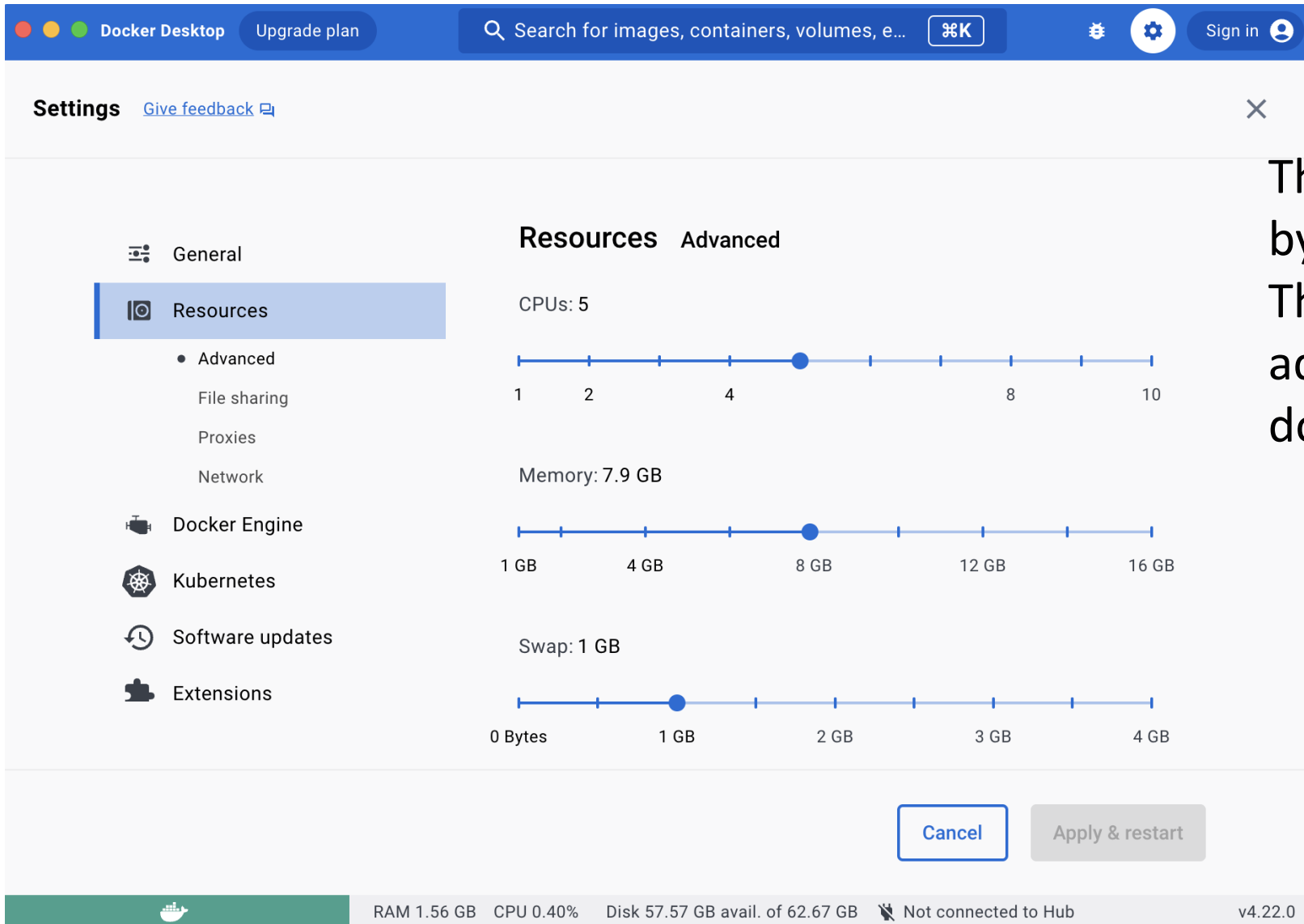
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
  1 root  20  0  4136 3320 2828 S  0.0  0.0 0:00.02 /bin/bash
255 root  20  0  4164 2764 2384 R  0.0  0.0 0:00.80 htop

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

CPU limit to 5  
cores

Ram limit to 7.6G

# Docker setting



The screenshot shows the Docker Desktop Settings window, specifically the 'Resources' section under 'Advanced'. The left sidebar lists settings categories: General, Resources (selected), File sharing, Proxies, Network, Docker Engine, Kubernetes, Software updates, and Extensions. The main area displays three resource sliders: CPUs (set to 5), Memory (set to 7.9 GB), and Swap (set to 1 GB). At the bottom, there are 'Cancel' and 'Apply & restart' buttons. The status bar at the very bottom shows system metrics: RAM 1.56 GB, CPU 0.40%, Disk 57.57 GB avail. of 62.67 GB, and version v4.22.0.

Resource	Current Value	Range
CPU	5	1 to 10
Memory	7.9 GB	1 GB to 16 GB
Swap	1 GB	0 Bytes to 4 GB

The performance was limited by Docker program.  
The performance can be adjusted in setting page of docker program



# Run htop

Run new command in running container

**`docker exec -it container_ID bash`**

kwan — root@2c4ea9ba5a04: /data — com.

```
Last login: Fri Aug 18 09:37:19 on ttys002
(base) kwan@MBPkhxnwanrutai ~ % docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
2c4ea9ba5a04   ubuntu:latest  "/bin/bash"             3 hours ago   Up About an hour                stupefied_einstein
(base) kwan@MBPkhxnwanrutai ~ % docker exec -it 2c4ea9ba5a04 bash
root@2c4ea9ba5a04:/data#
```

kwan — root@2c4ea9ba5a04: /data — com.docker.cli < docker start -a -i 2c4ea9ba5a04 — 99...

0[

1[

2[|

Mem[|||||]

Swp[

0.0%]

0.0%]

0.7%]

783M/7.67G]

0K/1024M]

3[

4[||

Tasks: 3, 0 thr; 1 running

Load average: 0.00 0.06 0.06

Uptime: 00:09:54

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	4136	3320	2828	S	0.0	0.0	0:00.02	/bin/bash
255	root	20	0	4164	2764	2384	R	0.0	0.0	0:00.51	htop
256	root	20	0	4136	3248	2876	S	0.0	0.0	0:00.00	bash

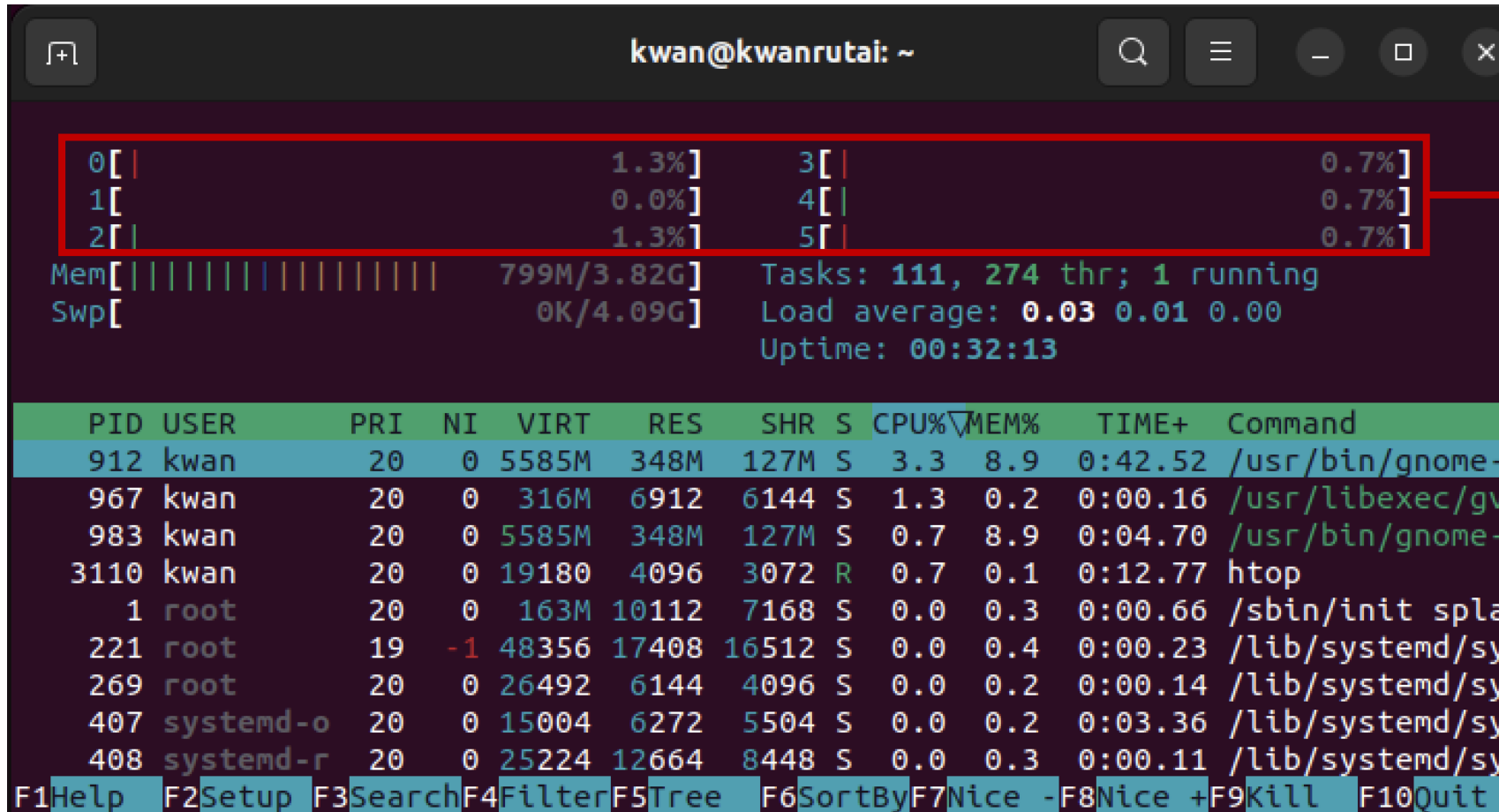
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

New process of “bash” command start after htop command

# Using htop utility

Run htop: **htop**

#Run htop in Ubuntu virtual machine



CPU used

Try running the following key:

F9 : Kill process

F2 : Setup htop display

F4 : Filter

F6 : Sort by selecting column

F10 : Quit htop

Access the manual for the 'htop' command to explore the various options that can be used with it.

Let's try using the '-u' option to filter and display only specific users.

# Program installation and repository

# What are Repositories?

- Thousands of Ubuntu programs are readily available to satisfy users' demands.
- Repositories simplify the installation of new software, ensuring a high level of security by thoroughly testing and tailoring the software for each Ubuntu version.
- Four main repositories are:
  1. **Main** – Canonical-supported free open-source software
  2. **Universe** – Community-maintained free and open-source software
  3. **Restricted** - Proprietary drivers for devices. This includes tools and drivers that are necessary for the proper functioning of the operating system.
  4. **Multiverse** - Software restricted by copyright or legal issues.

# Edit software repositories using the command line

- The list of software repositories is present in the software repositories configuration file.  
“[/etc/apt/sources.list](#)”
- You can add, remove, or temporarily disable software repositories.
- Before doing anything with the software repositories file, make a backup of the file first.

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
```

# Edit software repositories using the command line

- Open file for modify:

```
sudo nano /etc/apt/sources.list
```

- Adding repositories using command “**add-apt-repository**”

```
sudo add-apt-repository \  
"deb http://us.archive.ubuntu.com/ubuntu/ jammy universe"
```

↑  
Repositories contain binaries  
or precompiled packages

**URI**  
Uniform  
Resource  
Identifier

↑  
Release name or version of Linux distro

↑  
The section names or components.

- Run update repositories after modify.

```
sudo apt-get update
```

# Install, Update and Upgrade program from repositories

- The apt command is a powerful command-line tool that work with Ubuntu's Advanced Packaging Tool (APT).
- The following are the examples of apt utility:

- Install a package

```
sudo apt install program-name
```

- Remove a package

```
sudo apt remove program-name
```

- Update the package index

```
sudo apt update
```

- Upgrade packages

```
sudo apt upgrade
```

# Quiz

- The quiz for this session will be conducted at the beginning of the next class.