Computer Science Fundamentals and Career Pathways

Assignment Number 02: Basics of Linux and Open-Source Tools

Made By-

Bhavya Shany

Roll No. 2501730024

Section A

Step 1- Installation

Option B — Install Ubuntu (WSL 2) on Windows 10/11 Prerequisites: Windows 10 or Windows 11. PowerShell (Admin) access.

Steps (PowerShell/Command Prompt as Administrator) 1.

#Enable WSL and Virtual Machine Platform features:

wsl --install

(This installs WSL and will install a default Linux distribution (usually Ubuntu)).

#If you already have WSL but not WSL2 enabled,

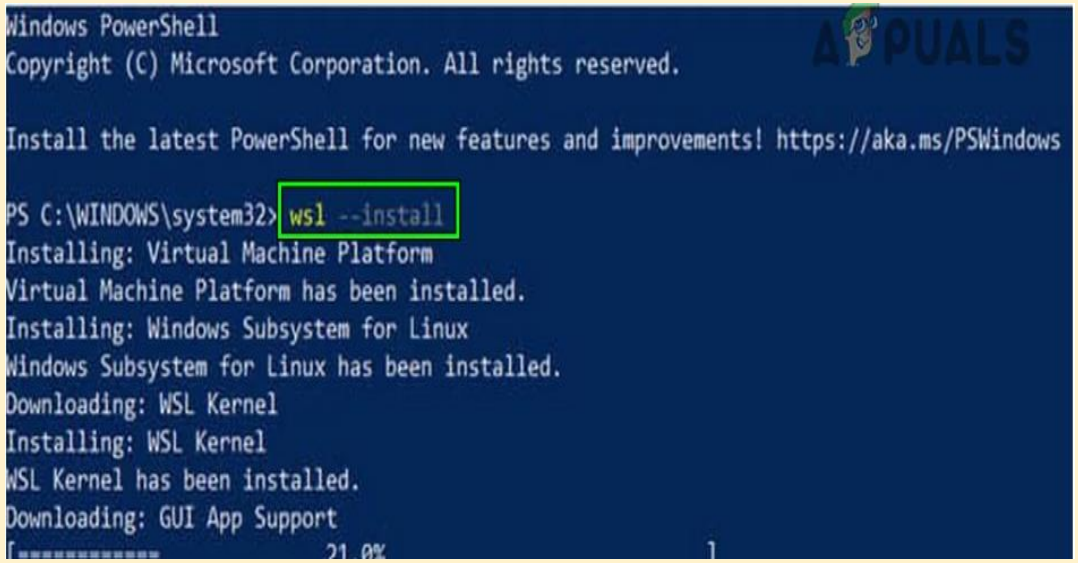
run: **wsl --update wsl --set-default-version 2**

#Install Ubuntu from Microsoft Store: **search "Ubuntu"** and click Install (e.g., Ubuntu 22.04 LTS).

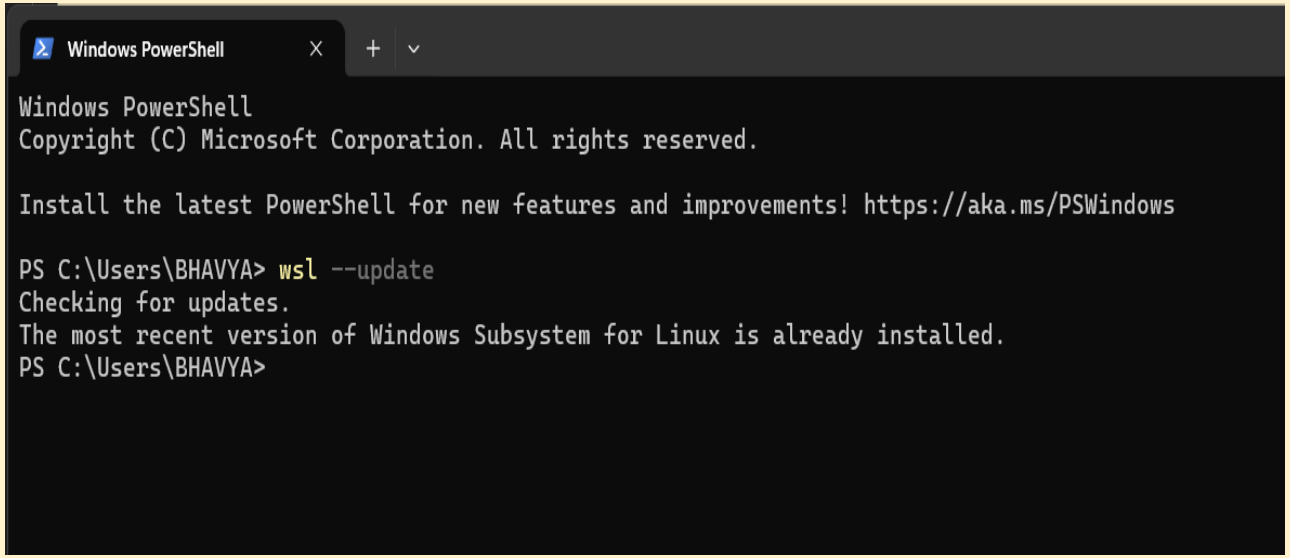
#Launch Ubuntu from Start Menu → set username and password. Screenshots to capture: - PowerShell command running

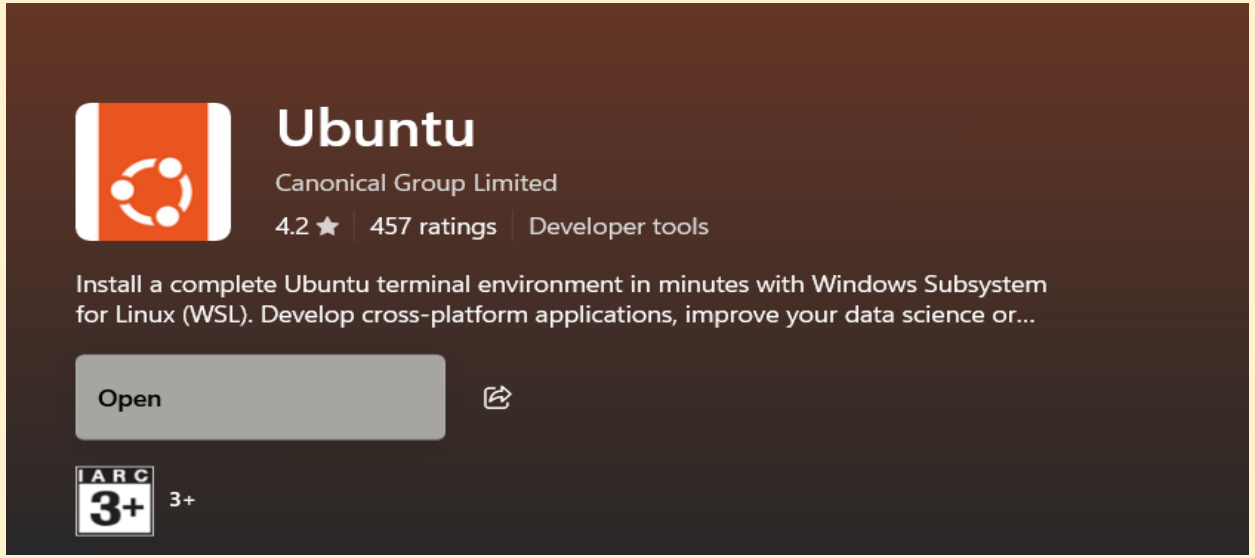
wsl --install (show success message) .

wsl--install



wsl--update





Hardware Configuration:

WSL example - Host OS: Windows 11 - WSL version: WSL 2 - Ubuntu distro: Ubuntu 22.04 LTS - Memory/ disk: WSL uses host resources

Step 2 — Shell command implementations and documentation

A) File Navigation

```
PS C:\Users\BHAVYA> pwd

Path
----
C:\Users\BHAVYA
```

1) pwd

Syntax: pwd

Description: Print working directory — shows current directory path.

When/Why: Use to confirm your current directory before running file operations.

2) ls

Syntax: ls

Description: List directory contents.

When/Why: Quick view of files; use(-l for long format, -a to include hidden files).

```
PS C:\Users\BHAVYA> ls


    Directory: C:\Users\BHAVYA


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        28-10-2025     22:55                .anaconda
d-----        28-10-2025     23:07                .conda
d-----        28-10-2025     14:04                .continuum
d-----        24-09-2025     01:50                .idlerc
d-----        28-10-2025     22:55                .ipynb_checkpoints
d-----        28-10-2025     22:59                .ipython
d-----        28-10-2025     22:59                .jupyter
d-----        22-09-2025     10:13                .vscode
d-----        28-10-2025     20:52                anaconda3
d-----        28-10-2025     22:55                anaconda_projects
d-r---        21-09-2025     21:29                Contacts
d-----        28-09-2025     22:33                Documents
d-r---        12-11-2025     17:07                Downloads
d-r---        21-09-2025     21:29                Favorites
d-----        14-11-2025     23:23                Folder2
d-----        14-11-2025     23:23                Folder3
d-----        14-11-2025     23:23                ImpFolder
```

3) Cd

```
PS C:\Users\BHAVYA> cd "C:\Users\BHAVYA\OneDrive\Desktop\IMP"
PS C:\Users\BHAVYA\OneDrive\Desktop\IMP> mkdir hello


    Directory: C:\Users\BHAVYA\OneDrive\Desktop\IMP


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        14-11-2025     23:57                hello


PS C:\Users\BHAVYA\OneDrive\Desktop\IMP>
```

Syntax: cd [directory]

Description: Change directory.

When/Why: Navigate filesystem to the folder where you want to operate.

4) tree

```
PS C:\Users\BHAVYA\OneDrive\Desktop\IMP> tree
Folder PATH listing
Volume serial number is 6C29-4AE7
C:.
├───Chemistry
│   ├───Fuels
│   └───Water Treatment
├───CSFP
├───hello
├───LabAssigmentKRMUWebDev
│   └───Portfolio
├───Mathematics
│   ├───pythonmaths
│   ├───Unit 1
│   ├───Unit 2
│   └───Unit 3
├───Python
│   └───Python Assignments
├───Random
│   └───Study-Sphere
├───Tinker Cad
└───WebDev
    ├───Unit1
    └───Unit2
PS C:\Users\BHAVYA\OneDrive\Desktop\IMP>
```

Syntax: tree [options] [directory]

Description: Display directory tree (hierarchical).

When/Why: Visualize directory structure — useful for documentation and debugging. (If tree is missing: sudo apt update && sudo apt install tree .)

B) File and Directory Management

1) mkdir

```
PS C:\Users\BHAVYA> mkdir Folder1, Folder2, Folder3


    Directory: C:\Users\BHAVYA


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        14-11-2025     23:23                Folder1
d-----        14-11-2025     23:23                Folder2
d-----        14-11-2025     23:23                Folder3
```

Syntax: mkdir [options] directory_name

Description: Create a directory.

When/Why: Create directories to organize files.

2)touch

Syntax: touch filename

Description: Create an empty file or update timestamp.

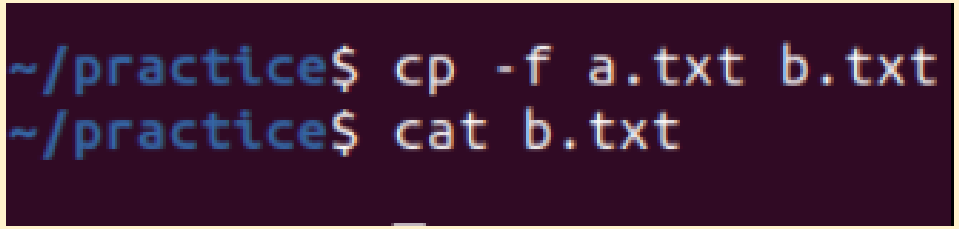
When/Why: Quickly create placeholder files.

3) cp

Syntax: cp [options] source destination

Description: Copy files or directories (-r for recursive).

When/Why: Duplicate files or folder trees.



4) mv

Syntax: mv [options] source destination

Description: Move or rename files/directories.

When/Why: Organize or rename items.

5) rm

Syntax: rm [options] file

Description: Remove files. Use (-r to remove directories recursively and -f to force).

When/Why: Delete unnecessary files. Be careful — deletions are permanent.

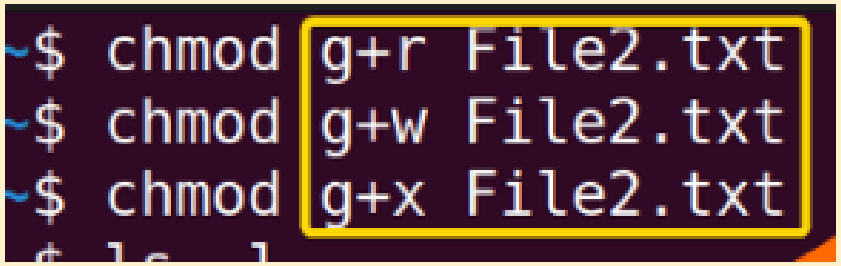
C) Permissions Management

1) chmod

Syntax: chmod [options]

Description: Change file mode (permissions)

When/Why: Secure files by restricting write/execute permissions.

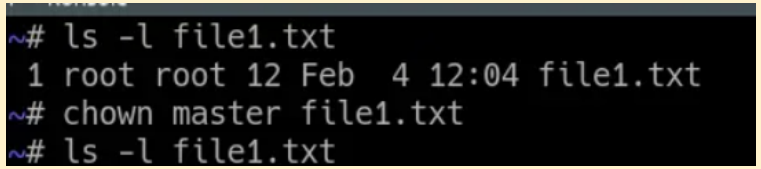


2) chown

Syntax: chown [owner][:group] file

Description: Change file owner and/or group.

When/Why: Set correct ownership for scripts or service files.



D) Process Monitoring

1) ps

Syntax: ps [options]

Description: Report snapshot of current processes. Common: (ps aux).

When/Why: Check running processes, PIDs for troubleshooting.

2) top

Syntax: top

Description: Interactive process viewer with CPU/memory usage.

When/Why: Monitor system in real-time; kill misbehaving processes.

3) kill

Syntax: kill [signal] PID or kill -9 PID

Description: Send signals to processes (terminate, etc.).

When/Why: Stop misbehaving processes when graceful shutdown fails.

```
~$ kill -l
T        3) SIGQUIT      4) SIGILL
S        8) SIGFPE       9) SIGKILL
R2      13) SIGPIPE     14) SIGALRM
LD      18) SIGCONT     19) SIGSTOP
```

E) Networking Tools

1) Ping

Syntax: ping [options] destination

Description: Send ICMP echo requests to check reachability and latency. When/Why: Basic network connectivity test.

```
[root@localhost ~]# ping www.ipcisco.com
PING www.ipcisco.com (178.128.175.110): 56 data bytes
64 bytes from 178.128.175.110: seq=0 ttl=56 time=376.290 ms
64 bytes from 178.128.175.110: seq=1 ttl=56 time=362.500 ms
64 bytes from 178.128.175.110: seq=2 ttl=56 time=362.860 ms
64 bytes from 178.128.175.110: seq=3 ttl=56 time=362.220 ms
64 bytes from 178.128.175.110: seq=4 ttl=56 time=357.825 ms
64 bytes from 178.128.175.110: seq=5 ttl=56 time=365.660 ms
```

2) Ifconfig / ip

Syntax: ifconfig or ip addr show

Description: Show network interface configuration. (ifconfig may be deprecated in favor of ip .)

When/Why: See assigned IPs and interface states.

3) netstat (or ss)

Syntax: netstat -tuln or ss -tuln

Description: List network sockets and listening ports.

When/Why: Debug network services and port conflicts.

Step 3 — Shell scripts

- Script A — Backup a Directory (backup_dir.sh)

```
#!/bin/bash
# backup_dir.sh
# Purpose: Copy a specified directory to a backup folder with a timestamp.
#Author-Bhavya Shany
# Date: 2025-11-09

# Usage: ./backup_dir.sh /path/to/source /path/to/backup_root

SOURCE_DIR="$1"
BACKUP_ROOT="$2"

if [[ -z "$SOURCE_DIR" || -z "$BACKUP_ROOT" ]]; then
  echo "Usage: $0 SOURCE_DIR BACKUP_ROOT"
  exit 1
fi

if [[ ! -d "$SOURCE_DIR" ]]; then
  echo "Source directory does not exist: $SOURCE_DIR"
  exit 2
fi

TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
BACKUP_DIR="$BACKUP_ROOT/backup_$(basename "$SOURCE_DIR")_$TIMESTAMP"

mkdir -p "$BACKUP_DIR"

# Copy files recursively while preserving attributes
cp -a "$SOURCE_DIR/" "$BACKUP_DIR/"

if [[ $? -eq 0 ]]; then
  echo "Backup successful: $BACKUP_DIR"
  exit 0
else
  echo "Backup failed"
  exit 3
fi
```

Explanation & usage: - The script accepts two arguments — the source path and a backup root directory. - It creates a timestamped backup directory and uses cp -a to preserve permissions and symbolic links.

- Script B — CPU/Memory Monitoring (resource_monitor.sh)

```
#!/bin/bash
# resource_monitor.sh
# Purpose: Log CPU and memory usage to a file at regular intervals.
#Author-Bhavy Shany
# Date: 2025-11-09

# Usage: ./resource_monitor.sh /path/to/logfile interval_seconds
LOGFILE="$1"
INTERVAL=${2:-60}  # default to 60 seconds if not provided
```

```
if [[ -z "$LOGFILE" ]]; then
  echo "Usage: $0 LOGFILE [INTERVAL_SECONDS]"
  exit 1
fi

mkdir -p "$(dirname "$LOGFILE")"

# Header for logfile
echo "timestamp,cpu_percent,mem_total,mem_used,mem_free" >> "$LOGFILE"

while true; do
  TIMESTAMP=$(date +"%Y-%m-%d_%H:%M:%S")
  # cpu usage (percentage) using top in batch mode
  CPU_PERCENT=$(top -b -n1 | grep "%Cpu(s)" | awk '{print 100 - $8}')
  # memory using free
  read -r MEM_TOTAL MEM_USED MEM_FREE <<< $(free -m | awk '/Mem:/ {print $2"
"$3" "$4}')
  echo "$TIMESTAMP,$CPU_PERCENT,$MEM_TOTAL,$MEM_USED,$MEM_FREE" >> "$LOGFILE"
  sleep "$INTERVAL"
done
```

- Script C — Automated Download Task (auto_download.sh)

```
#!/bin/bash
# auto_download.sh
# Purpose: Download a file from the internet using wget or curl and store it in
a predefined directory.
#Author- Bhavya Shany
# Date: 2025-11-09

# Usage: ./auto_download.sh URL /path/to/download_dir

URL="$1"
DOWNLOAD_DIR="$2"
```

```
if [[ -z "$URL" || -z "$DOWNLOAD_DIR" ]]; then
  echo "Usage: $0 URL DOWNLOAD_DIR"
  exit 1
fi

mkdir -p "$DOWNLOAD_DIR"
FILENAME=$(basename "$URL")
DEST="$DOWNLOAD_DIR/$FILENAME"

# Prefer wget if available
if command -v wget >/dev/null 2>&1; then
  wget -O "$DEST" "$URL"
elif command -v curl >/dev/null 2>&1; then
  curl -L -o "$DEST" "$URL"
else
  echo "Neither wget nor curl is installed. Install one and retry."
  exit 2
fi

if [[ $? -eq 0 ]]; then
  echo "Downloaded $URL to $DEST"
  exit 0
else
  echo "Download failed"
  exit 3
fi
```


THANK
YOU