

Software Development Practice 1 (Section RSP)

Linux Environments for Software Development

Objective

- Learn how to install and configure Linux environments.
- Learn how to use the VS Code IDE under Linux (local and remote).
- Learn basic Bash shell commands on Linux.

Expected Learning Outcomes

- Students will be able to install, configure, and operate Linux environments to perform basic system administration tasks.
- Students will be able to develop, build, and deploy software in Linux environments.

Task 1: WSL2 + Ubuntu 24.04 on Microsoft Windows

- Install and configure **Ubuntu 24.04** on your native OS using **WSL2**.
- Refer to "*How to install Linux on Windows with WSL*" (Update: 11/20/2024)
🔗 <https://learn.microsoft.com/en-us/windows/wsl/install>
- Write a summary of the installation process and
- Capture some screenshots during software installation process for inclusion in the lab report.
 - Any problems encountered and their solutions should be included in the report.

Task 2: Oracle VirtualBox VM + Linux

- Install and configure **Linux distributions** on **virtual machines (Vms)**.
 - For example: **Ubuntu**, **Debian**, etc.
 - Refer to "**VirtualBox + VirtualBox Extension Pack Downloads (for personal and educational use)**"
 -  <https://www.virtualbox.org/wiki/Downloads>
 - "**VirtualBox Online Manuals**"
 -  <https://www.virtualbox.org/manual/ch01.html>
 -  <https://www.virtualbox.org/manual/ch02.html>
 - Make sure that your Ubuntu VM can access the Internet to update software package using "sudo apt update && sudo apt upgrade".
 - Write a summary of the installation process and capture some screenshots during software installation process for inclusion in the lab report.
 - Any problems encountered and their solutions should be included in the report.
 - **Question:** How do you configure your **Ubuntu VM and VirtualBox network settings** so that it can be accessed via an SSH client from Windows 11? Describe the steps required to complete this task in detail.
-

Task 3: Basic & Useful Linux Commands

- Run the following **Linux commands in a Linux bash shell** and observe the text output.
- Compare the outputs on both **Ubuntu VM and WSL2 Ubuntu** platforms.
- Explain the purpose or function of each command. Write a short comment for each command.

Commands for Linux / Ubuntu

```
$ sudo apt update && sudo apt upgrade -y && sudo apt dist-upgrade -y
$ uname -a
$ cat /etc/os-release | head -5
$ lsb_release -all
$ df -h
$ free -h
$ hostname
$ whoami
$ echo $USER
$ ip addr show dev eth0
$ ip route
$ lscpu | grep -e "^CPU(s):" | awk '{print "CPU cores: " $2}'
$ lsblk
$ uptime
$ ping -c 10 8.8.8.8
$ python3 -V
$ which python3
$ sudo apt install python3 python3-pip \
    python3-venv python3-setuptools python3-dev
$ which nvidia-smi && "$(which nvidia-smi)"
$ sudo reboot -h now
```

Commands for WSL2 Ubuntu

```
$ /mnt/c/WINDOWS/System32/wsl.exe --version
$ /mnt/c/WINDOWS/System32/wsl.exe --list
$ /mnt/c/WINDOWS/System32/wsl.exe --ubuntu
```

Questions

- Which command is used to determine the disk storage capacity?
 - Which command is used to find the number of CPU cores available?
 - Which command is used to find the IPv4 address of the Ubuntu machine?
 - Explain the purpose or function of the `~/.bashrc` file.
-

Task 4: VS Code IDE on Linux

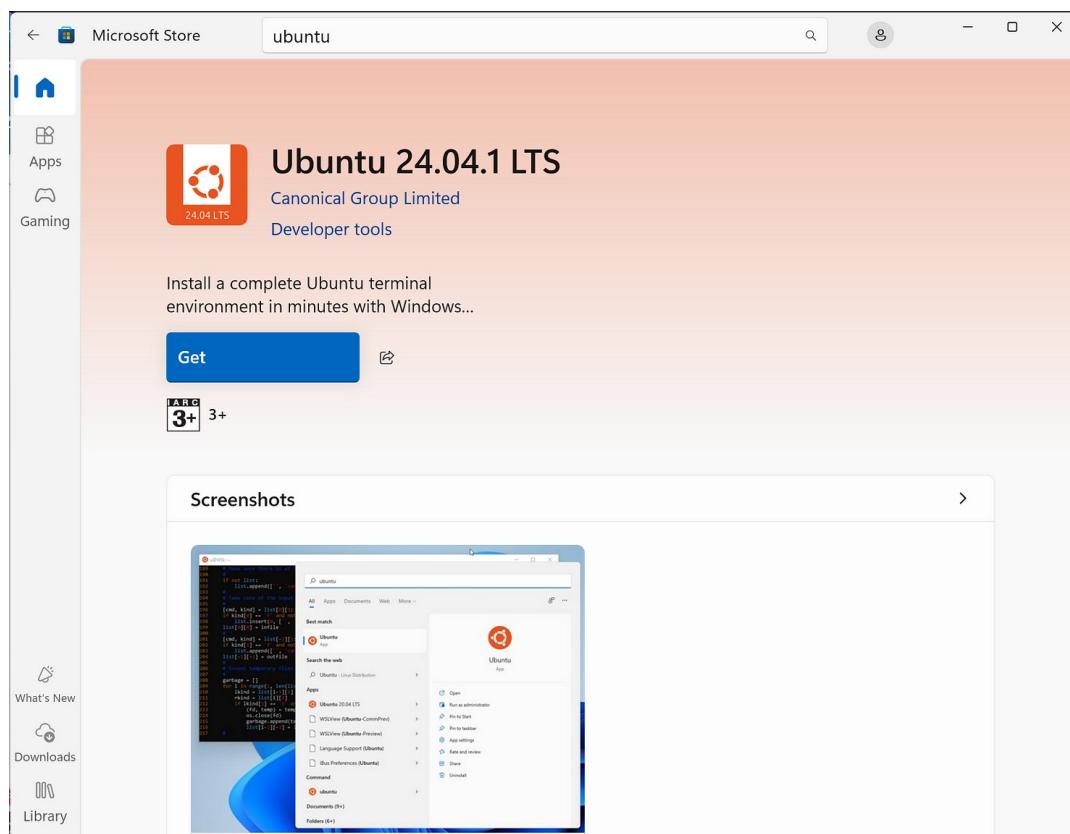
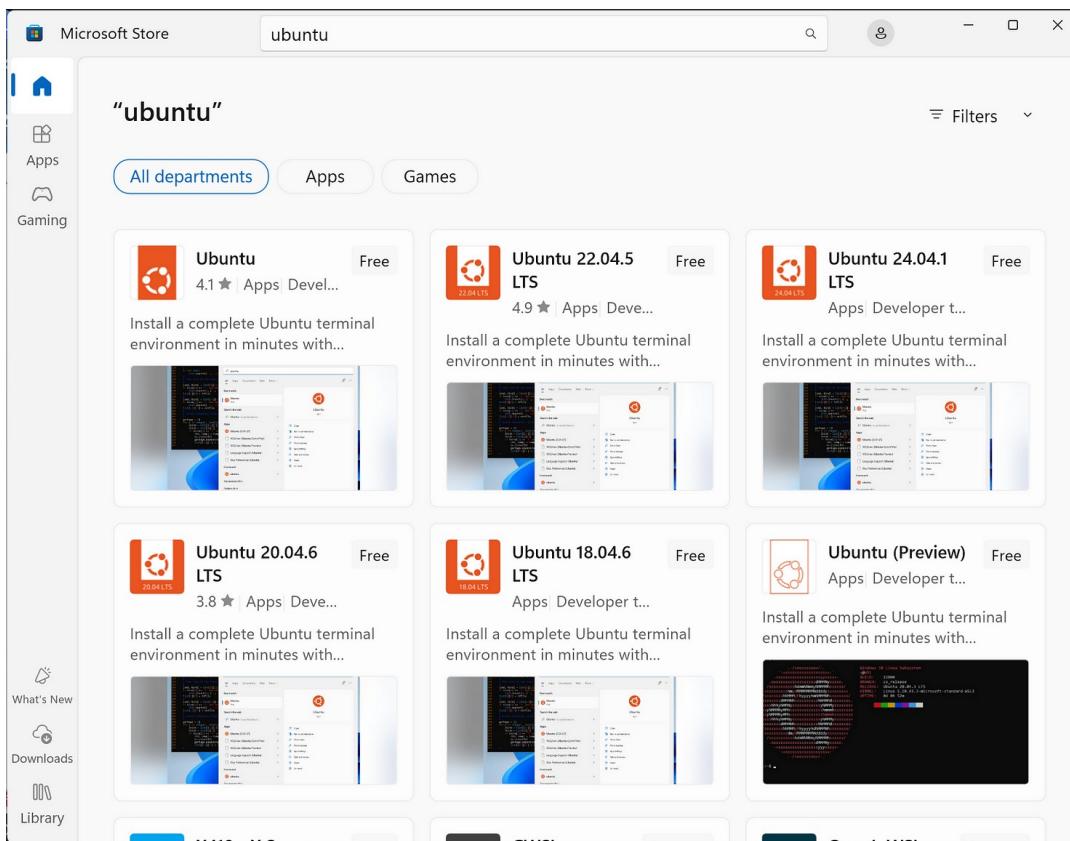
- Install **VS Code IDE** and the "**Microsoft Python Extension Pack**" on the following platforms.
 1. WSL2 Ubuntu 24.04 on Windows
 2. Ubuntu 24.04 / Virtual Box VM (remote access via SSH).
- Examples of extensions: **Python Formatter**
 -  <https://code.visualstudio.com/docs/python/formatting>
- Write Python scripts and run them **on Ubuntu remote machine while running the VS Code IDE on a local machine.**
- Install packages such as `matplotlib` and `numpy` (under a **Python virtual environment**).
 - Write **Python test scripts** that utilize these packages.
 - Demonstrate your work to the lab instructor for **both WSL2 Ubuntu 24.04** and **Ubuntu 24.04 / Virtual Box VM**.

Useful online documentations

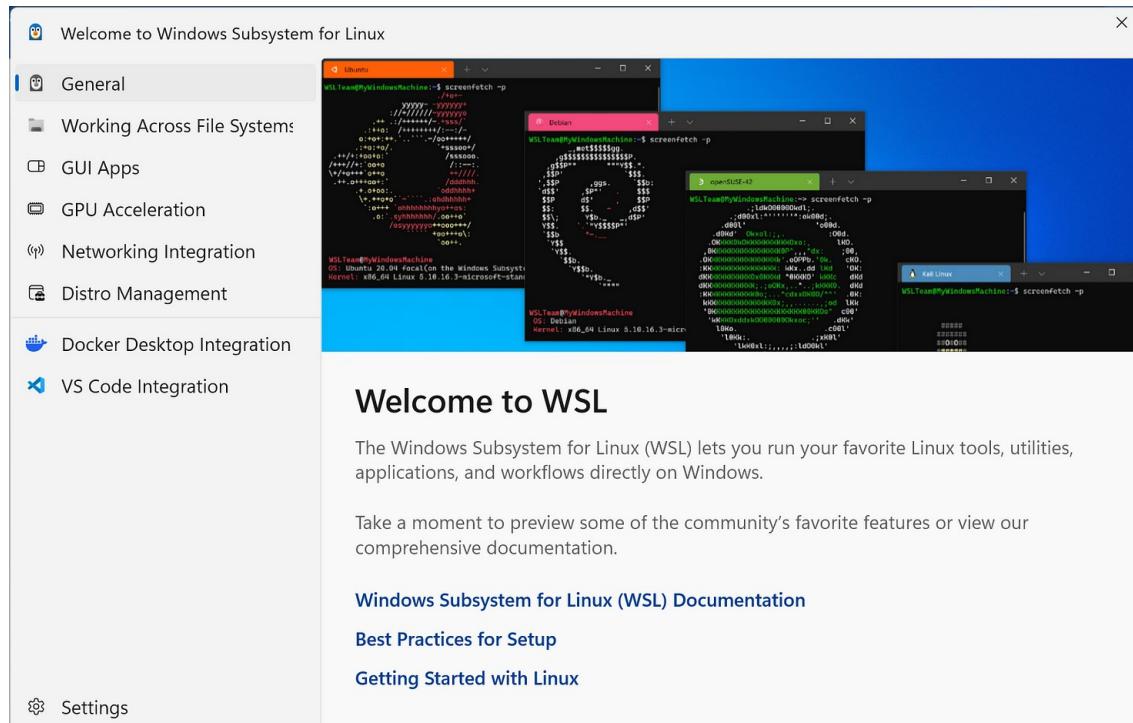
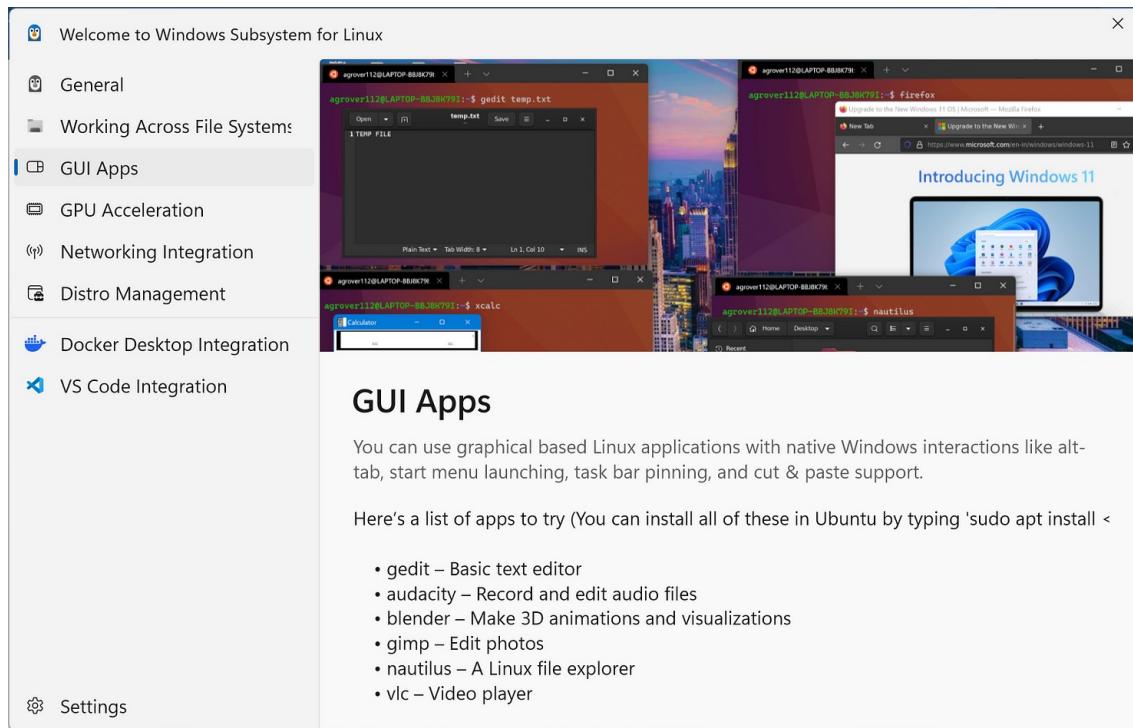
- https://iot-kmutnb.github.io/blogs/teaching/sw_dev_2024-1/handout-1_2024-06-29.pdf
- https://iot-kmutnb.github.io/blogs/teaching/sw_dev_2024-1/handout-2_2024-06-29.pdf
- https://iot-kmutnb.github.io/blogs/teaching/sw_dev_2024-1/handout-3_2024-07-05.pdf

Last update: 2025-06-23

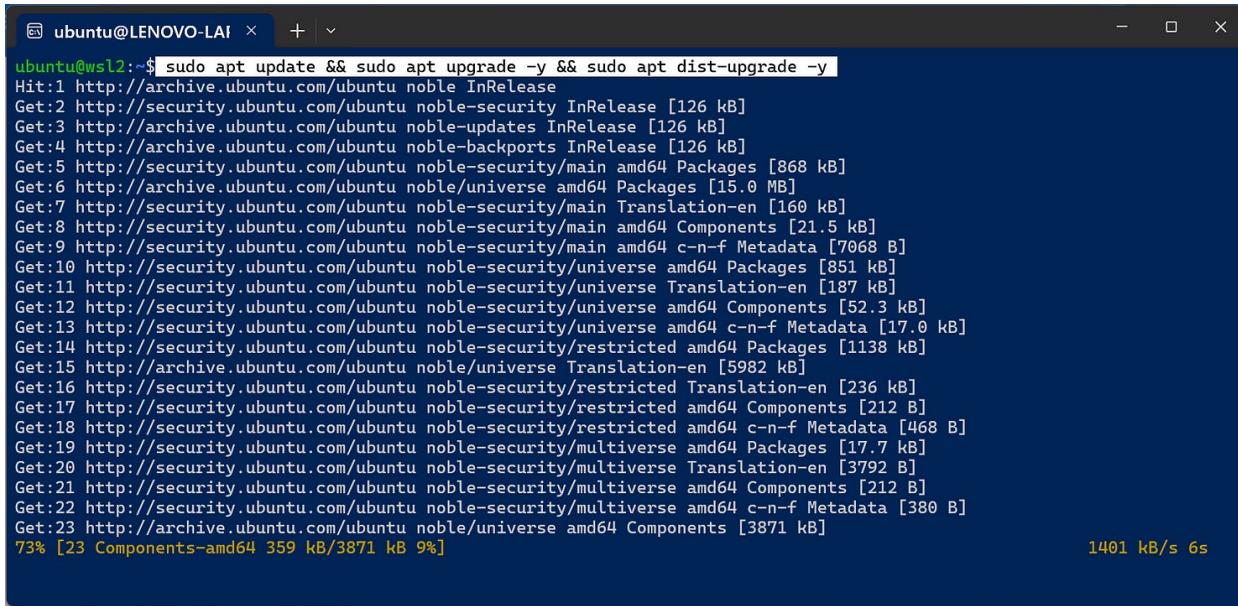
MS Store on Windows 11: Ubuntu Distributions for WSL 2



Note that WSL2 Ubuntu also support GUI-based applications.



Linux commands to upgrade Ubuntu software packages



```
ubuntu@ws12:~$ sudo apt update && sudo apt upgrade -y && sudo apt dist-upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [868 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [160 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [7068 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [851 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [187 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1138 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [236 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [468 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [17.7 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [3792 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [380 B]
Get:23 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
73% [23 Components-amd64 359 kB/3871 kB 9%] 1401 kB/s 6s
```

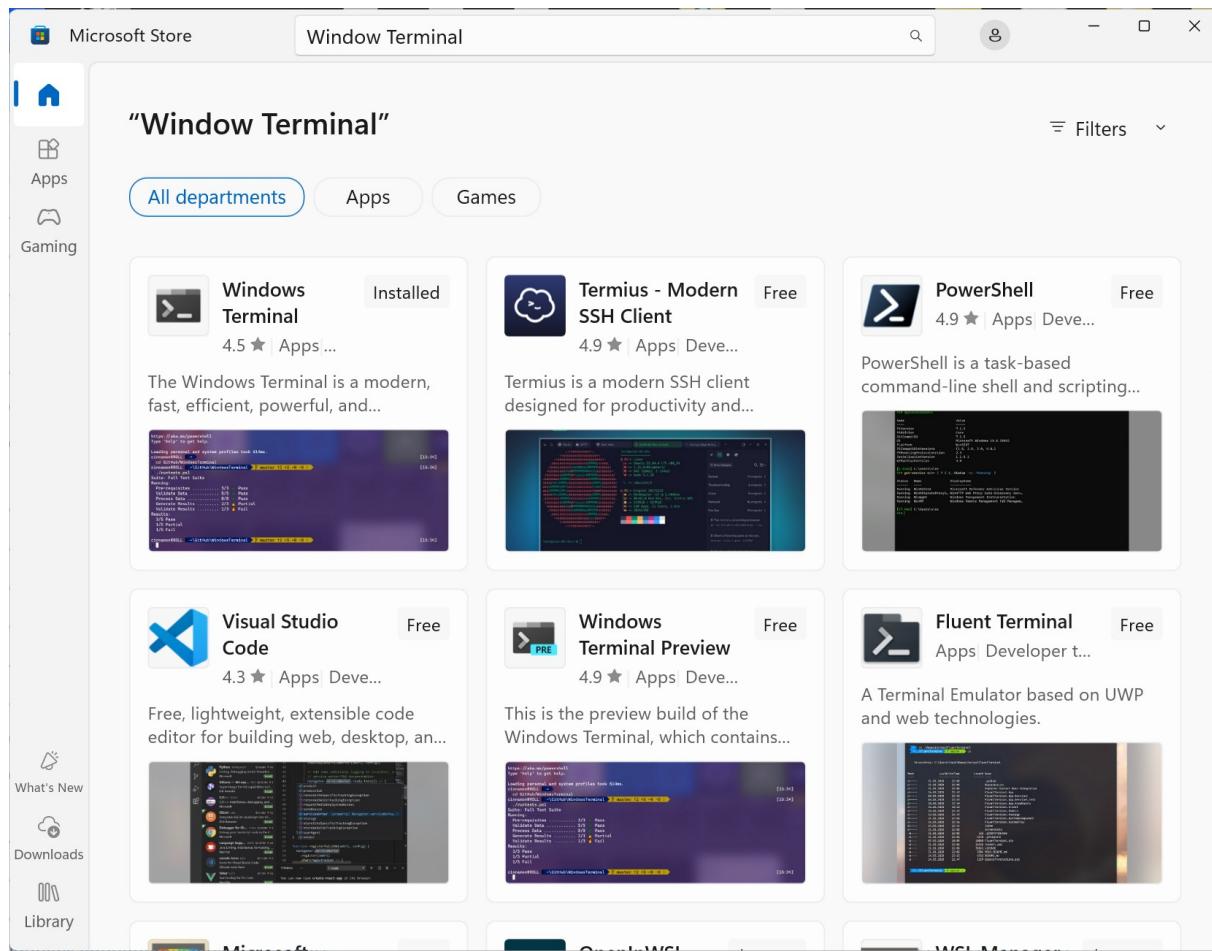
Linux commands for Creating a Python Virtual Environment

```
$ sudo apt install python3 python3-pip \
    python3-setuptools python3-venv python3-dev

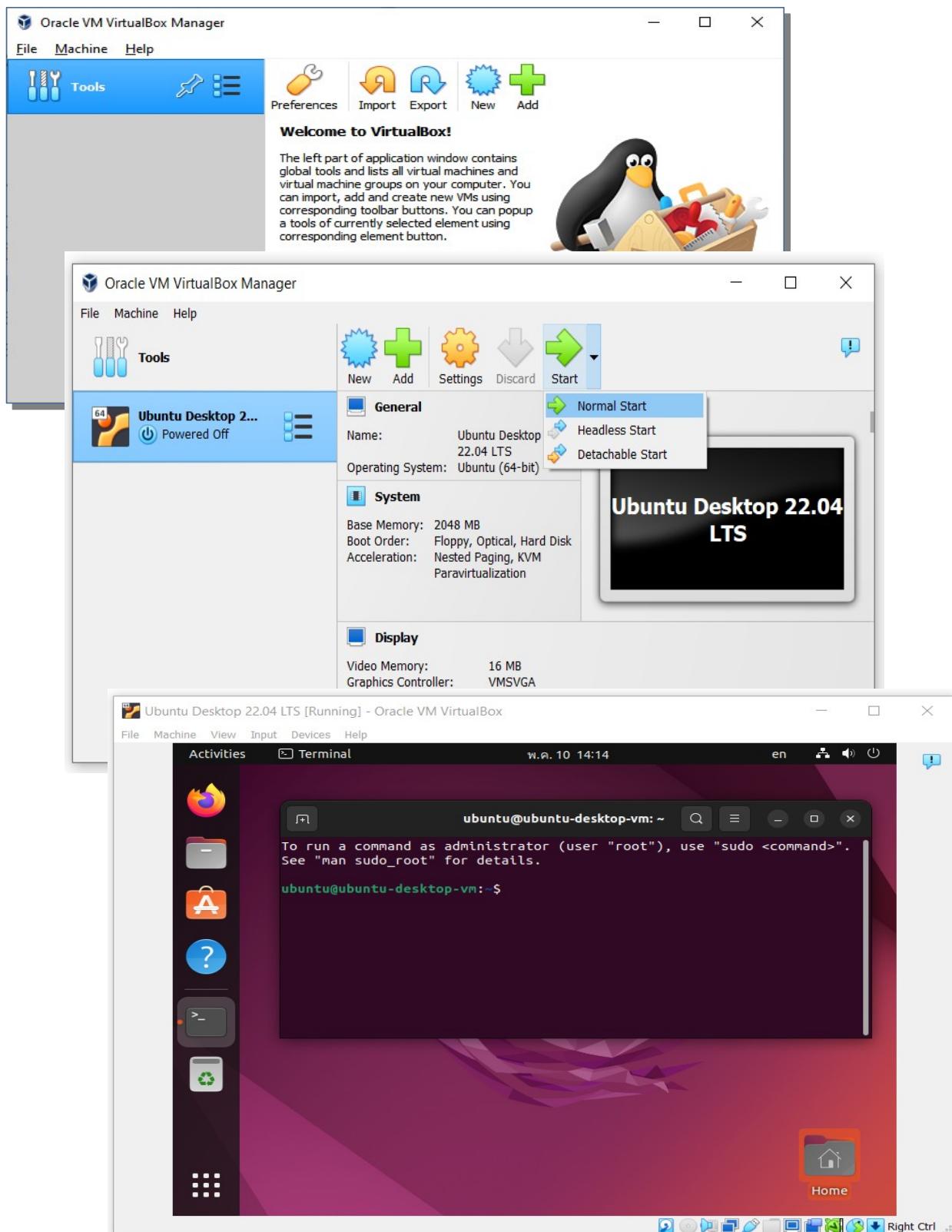
$ python3 -m venv ~/.venv

$ source ~/.venv/bin/activate
```

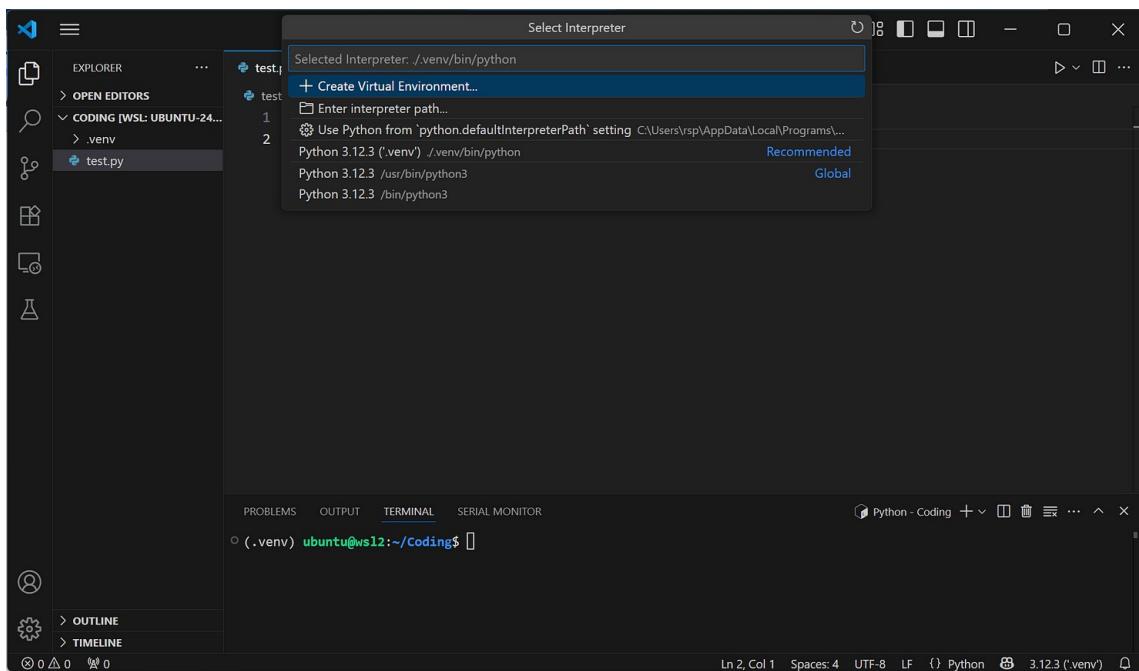
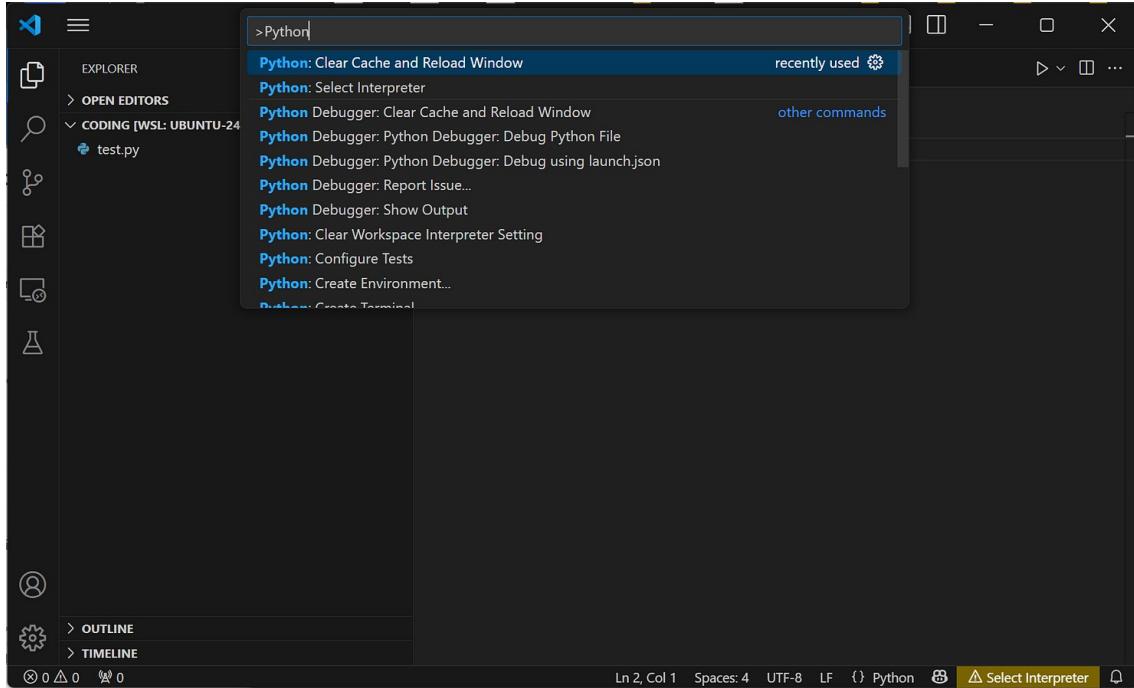
Installing Software via Microsoft Store



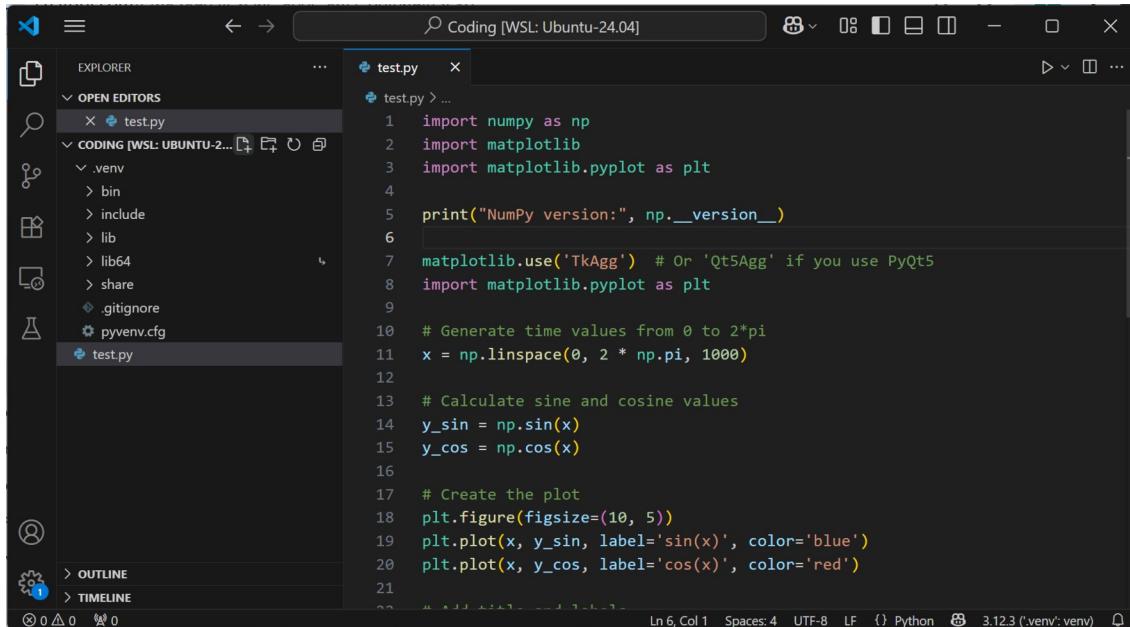
Ubuntu under Oracle VirtualBox



Creating Python Virtual Environment Under VS Code IDE (WSL2 Ubuntu).



Testing a Python script using VS Code IDE on WSL 2 Ubuntu



A screenshot of the VS Code interface on a WSL 2 Ubuntu 24.04 system. The left sidebar shows a file tree with a .venv folder containing bin, lib, lib64, share, .gitignore, and pyvenv.cfg. The main editor window displays the following Python script:

```
test.py
1 import numpy as np
2 import matplotlib
3 import matplotlib.pyplot as plt
4
5 print("NumPy version:", np.__version__)
6
7 matplotlib.use('TkAgg') # Or 'Qt5Agg' if you use PyQt5
8 import matplotlib.pyplot as plt
9
10 # Generate time values from 0 to 2*pi
11 x = np.linspace(0, 2 * np.pi, 1000)
12
13 # Calculate sine and cosine values
14 y_sin = np.sin(x)
15 y_cos = np.cos(x)
16
17 # Create the plot
18 plt.figure(figsize=(10, 5))
19 plt.plot(x, y_sin, label='sin(x)', color='blue')
20 plt.plot(x, y_cos, label='cos(x)', color='red')
21
```

The status bar at the bottom indicates the file is 3.12.3 (.venv: venv) and shows other details like line 6, column 1, and encoding.

