Installing and Using WSL2 on Windows

Setting up **WSL2** (Windows Subsystem for Linux 2) and **Windows Terminal**

What You'll Install

1. **WSL2** - Run Linux directly on Windows
2. **Windows Terminal** - Modern terminal application
3. **Ubuntu** - The Linux distribution we'll use

What is WSL2?

- **W**indows **S**ubsystem for **L**inux version **2**
- Lets you run a real Linux environment on Windows
- No need for dual boot or virtual machine
- Full Linux kernel with better performance
- Works great with development tools

WSL2 is not Linux Emulation, it's a real Linux kernel running on Windows!

- Runs real Linux binaries on Real Linux kernel
- Full system call compatibility
- Supports Docker and other advanced tools
- Seamless file sharing between Windows and Linux
- It's just like having Linux installed on your machine!

For ASE courses and projects, WSL2 is the recommended way to run Linux on Windows!

- UNIX/Linux is widely used in software development
- Use Linux command line tools
- Run development environments natively
- Run scripts and programs for automation
- All examples and instructions will assume Linux or Mac environment (not Windows CMD or PowerShell)

Prerequisites

Before starting, make sure you have:

- ✅ Windows 10 (Version 2004 or higher) **OR** Windows 11
- ✅ Administrator access on your computer
- ✅ Internet connection
- ✅ At least 4GB of free disk space

Step 1: Install WSL2

Quick Installation (Recommended)

1. **Open PowerShell as Administrator**
   - Press `Windows Key`
   - Type "PowerShell"
   - Right-click → "Run as administrator"
2. **Run this command:**

```
wsl --install
```

Note: This command automatically installs WSL2 (not WSL1) on Windows 10 version 2004+ and Windows 11.

3. **Wait for installation to complete** (may take 5-10 minutes)

Step 2: Restart Your Computer

- After installation finishes, **restart your computer**
- This is required for WSL2 to work properly

Step 3: Set Up Ubuntu

After restarting:

1. **Ubuntu will launch automatically**
   - If not, search for "Ubuntu" in Start Menu
2. **Create your Linux username and password**
   - Enter a username (lowercase, no spaces)
   - Enter a password (you won't see it as you type - this is normal!)
   - Re-enter password to confirm

⚠️ **Important:** Remember this password! You'll need it for `sudo` commands.

Step 4: Verify WSL2 Installation

1. **Open PowerShell** (doesn't need to be as Administrator)

2. **Check your installation:**

```
wsl -l -v
```

3. **Expected output:**

```
  NAME      STATE           VERSION
* Ubuntu    Running         2
```

✅ Make sure VERSION shows **2** (not 1)

If VERSION Shows 1

If your Ubuntu shows VERSION 1, upgrade it:

```
wsl --set-version Ubuntu 2
```

Set WSL2 as default for future installations:

```
wsl --set-default-version 2
```

Install Windows Terminal

Method 1: Microsoft Store (Easiest)

1. Open **Microsoft Store**
2. Search for **"Windows Terminal"**
3. Click **Install**

Method 2: Using PowerShell

```
winget install --id Microsoft.WindowsTerminal -e
```

Why Windows Terminal?

- **Multiple Tabs** - Run Linux, PowerShell, and CMD in one window
- **Split Panes** - Work in multiple terminals side-by-side
- **Customizable** - Change colors, fonts, and themes
- **Better Performance** - GPU-accelerated text rendering
- **Modern Interface** - Clean and easy to use

Using WSL2 in Windows Terminal

Launch Ubuntu in Windows Terminal

1. **Open Windows Terminal**
2. **Click the ▼ (down arrow)** next to the + tab
3. **Select "Ubuntu"** from the dropdown menu

Now you're running Linux inside Windows Terminal!

Basic WSL2 Commands

Check WSL version:

```
wsl -l -v
```

Start WSL:

```
wsl
```

Stop all WSL instances:

```
wsl --shutdown
```

Update WSL:

```
wsl --update
```

Accessing Files Between Windows and Linux

WSL2 is an isolated environment, so all the file systems in WSL2 are separate from Windows by default.

However, Microsoft provides **two ways** to access files across the boundary:

1. **From Linux → Access Windows files** via `/mnt/`
2. **From Windows → Access Linux files** via `\\wsl$\` or `\\wsl.localhost\`

Understanding File Systems

Two Separate File Systems:

- **Windows File System:** `C:\`, `D:\`, etc.
  - Your Windows files live here
  - Slower when accessed from Linux
- **Linux File System:** `/home/`, `/var/`, etc.
  - Your Linux files live here
  - Faster when working in Linux
  - Accessed from Windows via special network path

From Linux → Access Windows Files

Windows drives are **automatically mounted** at `/mnt/`:

```
# Go to C: drive
cd /mnt/c/

# Go to your Windows user folder
cd /mnt/c/Users/YourUsername/

# List Windows Desktop files
ls /mnt/c/Users/YourUsername/Desktop/

# Access D: drive (if you have one)
cd /mnt/d/
```

Example: Working with Windows Files from Linux

```
# Create a file on Windows Desktop from Linux
echo "Hello from Linux!" > /mnt/c/Users/YourUsername/Desktop/test.txt

# Copy a Windows file to Linux home directory
cp /mnt/c/Users/YourUsername/Documents/file.txt ~/

# Navigate to your Windows Downloads folder
cd /mnt/c/Users/YourUsername/Downloads/
```

⚠️ **Note:** Windows & Linux WSL are separated file systems and are connected via a virtual network. Accessing Windows files `/mnt/c/` from Linux is slower than working in Linux filesystem!

From Windows → Access Linux Files

Method 1: File Explorer (Easiest)

1. Open **File Explorer**
2. Look for **"Linux"** (penguin icon) in the left sidebar
3. Click to browse your Ubuntu files
4. Navigate to `Ubuntu` → `home` → `yourusername`

From Windows → Access Linux Files

Method 2: Direct Path in File Explorer

Type in the address bar:

```
\\wsl.localhost\Ubuntu\home\yourusername\
```

Or the shorter version:

```
\\wsl$\Ubuntu\home\yourusername\
```

⚠️ **Note:** wsl$ is a special network share that gives access to all your WSL distributions!

⚠️ **Note:** The `Ubuntu` part may differ if you installed a different distribution.

Example: Opening Linux Files in Windows Apps

From **inside Ubuntu terminal**, you can:

```
# Open current directory in Windows File Explorer
explorer.exe .

# Open a specific file in default Windows app
explorer.exe myfile.txt

# Open VS Code in current directory
code .
```

The `.exe` extension tells WSL to run a Windows program!

Where Should You Store Your Files?

✅ RECOMMENDED: Store projects in Linux filesystem

```
# Store your code here (FAST)
~/projects/my-app/
# Full path: /home/yourusername/projects/my-app/
```

Why? Much faster file operations when working in Linux!

Where Should You Store Your Files?

❌ NOT RECOMMENDED: Storing projects in Windows filesystem

```
# Storing here is SLOW when using Linux tools
/mnt/c/Users/YourUsername/projects/my-app/
```

Why? Significant performance penalty when Linux tools access Windows files.

Exception: You can use `/mnt/c/` for sharing files between Windows and Linux occasionally.

File Access Best Practices

Rule:Do not edit Linux files directly from Windows apps!

Why? It can cause file corruption and data loss! Don't forget that accessing Linux files from Windows is done over a virtual network share, which may not handle Linux file attributes properly.

✅ DO:

- **Keep your projects in Linux filesystem** (`~/projects/`)
- **Use `git clone`** inside WSL2 (not in `/mnt/c/`)
- **Install development tools** inside Ubuntu
- **Use `/mnt/c/`** only to access existing Windows files
- **Use `explorer.exe .`** to open Linux folders in Windows

❌ DON'T:

- **Don't work on code in `/mnt/c/`** when possible (performance)
- **Don't edit Linux files directly from Windows apps** (can cause corruption)
- **Don't create symlinks** from Windows to Linux (may not work)
- **Don't forget** that file permissions work differently
- **Don't access Git repos** stored in Windows from Linux or vice versa

File Name Translations

Inside Ubuntu terminal, accessing Windows files involves some name translations:

- c: → `/mnt/c/`
- d: → `/mnt/d/`

Inside Windows File Explorer, accessing Linux files involves:

- Linux root `/` → `\\wsl$\Ubuntu\`
- Linux home `/home/username/` → `\\wsl$\Ubuntu\home\username\`

Understanding File Permissions

Linux and Windows handle file permissions differently:

In Linux (WSL2):

```
# Check file permissions
ls -la myfile.txt

# Change permissions
chmod 755 myscript.sh

# Change ownership
chown username:username myfile.txt
```

Note: Files in `/mnt/c/` may have different permission behavior!

Common File Access Scenarios

Scenario 1: Clone a GitHub repo

```
# ✅ CORRECT: Clone to Linux filesystem
cd ~
git clone https://github.com/username/repo.git

# ❌ WRONG: Don't clone to Windows filesystem
cd /mnt/c/Users/YourUsername/
git clone https://github.com/username/repo.git  # SLOW!
```

Common File Access Scenarios

Scenario 2: Share a file between Windows and Linux

```
# Copy from Windows to Linux
cp /mnt/c/Users/YourUsername/Desktop/data.csv ~/

# Copy from Linux to Windows
cp ~/results.txt /mnt/c/Users/YourUsername/Desktop/
```

Common File Access Scenarios

Scenario 3: Edit Linux files with Windows VS Code

```
# From inside Ubuntu terminal
cd ~/projects/my-app
code .
```

✅ VS Code will open and edit files on WSL2 safely using the **WSL Remote extension**.

⚠️ Not Recommended:

Opening WSL files in Windows apps for editing.

✔️ Reading is OK

❌ Editing & saving can break permissions, Git, and file watchers.

(Almost) Perfect Linux Environment on Windows using GitHub/VSCode/Windows Terminal/Explorer.exe and WSL2

With these tools, we can use **WSL2 Ubuntu** as the main development environment!

1. Use **Windows Terminal** for a great terminal experience.
2. Use **VS Code WSL Remote extension** to edit code directly in WSL2 & Managing files.
3. Use **Explorer.exe** to open Linux folders in Windows Explorer when needed.
4. Use **GitHub** for version control and collaboration.

Quick File Access Commands

```
# Show current directory path
pwd

# List files with details
ls -la

# Go to home directory
cd ~
# or
cd /home/yourusername/

# Go to Windows Desktop from Linux
cd /mnt/c/Users/YourUsername/Desktop/

# Open current Linux directory in Windows Explorer
explorer.exe .
```

Common Issues and Solutions

Issue: "WSL 2 requires an update to its kernel component"

Solution:

1. Download the WSL2 kernel update from: https://aka.ms/wsl2kernel
2. Run the installer
3. Restart your computer

Common Issues and Solutions

Issue: "The attempted operation is not supported"

Solution:

- Make sure Virtualization is enabled in BIOS
- Check Windows version: Run `winver` and ensure version 2004+

Issue: Ubuntu is slow

Solution:

```
wsl --shutdown
```

Then restart Ubuntu

Issue: Can't find files I created in Windows

Solution:

Check the correct `/mnt/` path:

```
# Windows C:\Users\YourName\Desktop\file.txt is at:
ls /mnt/c/Users/YourName/Desktop/file.txt
```

Note: Replace `YourName` with your actual Windows username!

Issue: Permission denied when editing files

Solution:

```
# If file is owned by root, change ownership
sudo chown $USER:$USER myfile.txt

# Or run with sudo (be careful!)
sudo nano myfile.txt
```

Updating Ubuntu

Inside Ubuntu terminal, run:

```
# Update package list
sudo apt update

# Upgrade installed packages
sudo apt upgrade -y
```

Run this periodically to keep your system updated.

Installing Common Development Tools

Use `apt` to install tools like Git, Node.js, Python, etc.

- The -y flag auto-confirms prompts

```
apt install git nodejs python3 -y
```

Enjoy UNIX/Linux Environment on Windows!

Most developers prefer UNIX/Linux for development because:

- You can use any shell such as zsh or bash
- You can make scripts and automate tasks easily using the shell scripts
- You can make python scripts and run them natively
- You can use Docker and other container tools natively

You can do all these things on WSL2!