Setup

Mário Antunes

September 15, 2025

Universidade de Aveiro

Setting Up Your Digital Workspace

Goal for Today: Ensure everyone has a consistent and powerful work environment. This helps us learn faster and avoids the classic “but it works on my machine!” problem.

What is an Operating System (OS)?

Think of an OS as the **manager** of your computer's resources. □□

- It's the software that runs everything else.
- It manages the **CPU** (the brain), **memory** (the workspace), and **storage** (the filing cabinet).
- It provides a **user interface** (UI) for you to interact with the machine.

We'll be focusing on two main families:

- □ **Windows:** The most common desktop OS.
- □ **Linux:** A powerful, open-source OS family, dominant in servers, cloud computing, and scientific research.

What is a Filesystem?

A filesystem is the **library catalog** for your computer. It's how the OS organizes, stores, and finds your files.

Windows (NTFS)

- Uses **drive letters** (e.g., `C:`, `D:`).
- Path separator is a **backslash (`\`)**.
- Example:
  `C:\Users\YourName\Documents\MyFile.txt`

Linux (ext4, Btrfs, etc.)

- Has a single, unified **root directory (/)**.
- Everything, including devices, is treated like a file.
- Path separator is a **forward slash (/)**.
- Example: `/home/yourname/documents/myfile.txt`

Key takeaway: *Understanding the path structure is crucial for finding your files and running programs from the command line!*

Why a Standard Environment? (The “Linux” Choice)

We are standardizing on a **Linux-based command-line environment** because:

- **Industry Standard:** It’s the backbone of the web, cloud computing (AWS, Google Cloud), and scientific computing.
- **Powerful Tooling:** Offers unparalleled tools for programming, automation, and data manipulation.
- **Transparency:** Helps you understand what the computer is *actually* doing.

Now, let’s explore your options for getting this environment set up!

Your Three Paths to Linux

1. **Native Linux Installation:**
    - **What:** Linux is the main OS on your computer.
    - **Best for:** Maximum performance and full immersion.
2. **Virtual Machine (VM):**
    - **What:** A complete Linux computer running inside a window on your current OS.
    - **Best for:** Safe, isolated, and easy to reset.
3. **Windows Subsystem for Linux (WSL):**
    - **What:** A compatibility layer to run a real Linux environment directly inside Windows.
    - **Best for:** Tight integration between Windows and Linux tools.

Option 1: Native Linux Installation

This means you install a Linux distribution (like Ubuntu or Fedora) directly on your computer’s hardware, either replacing or alongside Windows (“dual-booting”).

Pros & Cons

- ✅ **Pro: Best Performance.** No overhead; Linux has direct access to all hardware (CPU, GPU).
- ✅ **Pro: Full Immersion.** Forces you to learn and adapt to the Linux environment.
- ❌ **Con: Complex Setup.** Can be tricky, with risks of data loss if not done carefully (backup is essential!).
- ❌ **Con: Hardware Compatibility.** Some specific hardware (Wi-Fi cards, webcams) might require extra configuration.

Who is this for?

Students who are adventurous, comfortable with computer hardware, or have a spare machine to experiment with.

Setup Steps

1. **Choose a distribution:** We recommend **Ubuntu 22.04 LTS** for its great support.
2. **Create a bootable USB drive:** Use tools like Rufus or BalenaEtcher.
3. **Partition your hard drive:** This is the most critical step if you plan to dual-boot. **BACK UP YOUR DATA FIRST!**
4. **Boot from the USB drive** and follow the installer instructions.

Option 2: Virtual Machine (VM)

A VM uses a **hypervisor** (like VirtualBox or VMWare) to emulate a full computer system inside your existing OS. We provide a pre-configured image to make this easy!

How it Works: Networking

Your VM needs network access to download software (`apt install`) or use `git`.

- The hypervisor creates a virtual network adapter for your VM.
- It usually uses **NAT (Network Address Translation)**, which acts like a router, allowing the VM to share your host computer's internet connection securely.

Pros & Cons

- ✅ **Pro: Safe & Isolated.** The VM is a sandbox. If you break it, it doesn't affect your main OS. You can easily delete it or reset it from a snapshot.
- ✅ **Pro: Easy Setup.** Just install VirtualBox and import the provided course image.
- ❌ **Con: Resource Heavy.** Requires significant RAM (8GB+ recommended for your whole system) and CPU power, as you are running two operating systems at once.
- ❌ **Con: Slower Performance.** Slower than a native install due to the overhead of virtualization.

Who is this for?

Almost everyone! It's the safest, most recommended, and most consistent option for this course.

Setup Steps

1. **Install VirtualBox:** Download and install the latest version of VirtualBox and its “Extension Pack”.
2. **Download the Course VM Image:** Get the `.ova` file from the course website.
3. **Import the Appliance:** In VirtualBox, go to `File > Import Appliance` and select the `.ova` file you downloaded. Follow the on-screen prompts.
4. **Start your VM:** Select the imported machine and click “Start”. That’s it!

Option 3: Windows Subsystem for Linux (WSL) □+□

WSL lets you run a genuine Linux kernel and environment directly on Windows, without the overhead of a full VM. It provides powerful integration between the two systems.

How it Works: Filesystem & Networking

- **Networking:** WSL automatically shares the network connection of your Windows host. It just works!
- **Filesystem Integration:** Your Windows drives (like `C:`) are automatically mounted inside Linux under `/mnt/`. For example, your `C:\Users\YourName` folder is accessible at `/mnt/c/Users/YourName`.

  Important: *For best performance, always work with your files inside the Linux filesystem (`/home/yourname/`), not on the mounted Windows drives (`/mnt/c/`).*

Pros & Cons

- ✅ **Pro: Excellent Performance.** Near-native speed for command-line tools.
- ✅ **Pro: Great Integration.** Easily call Linux tools from Windows and vice-versa. You can use VS Code on Windows to edit files directly inside WSL.
- ❌ **Con: “Headless” by Default.** WSL is primarily a command-line tool. Running Linux GUI apps requires extra setup (WSLg).
- ❌ **Con: Potential for Complexity.** Some advanced networking or hardware access can be more complex than in a VM or native install.

Who is this for?

Windows users who want a fast, integrated command-line environment and are comfortable working primarily in a terminal.

Setup Steps

1. **Enable WSL:** Open PowerShell **as an Administrator** and run this single command: `powershell wsl --install` This command will enable the required Windows features, download the latest Linux kernel, and install **Ubuntu** as the default distribution.
2. **Reboot** your computer when prompted.
3. **Create a User Account:** After rebooting, a terminal window will open to complete the Ubuntu installation. You will be asked to create a username and password. **Remember this password!**
4. **You're Ready!** You can launch your Linux terminal from the Start Menu (search for "Ubuntu").

Summary & Next Steps

You have three great options. Your choice depends on your comfort level and computer.

Feature	Native Install	Virtual Machine (VM)	WSL
Performance	□□□	□□	□□□
Safety/Isolation	□	□□□	□□
Ease of Setup	□	□□□	□□
Recommended For	Experts/Hobbyists	**Everyone (Default)**	Windows Users

Your Task Now:

1. **Choose one** of the three methods.
2. Follow the setup instructions to get it running.
3. Open a terminal and be ready for our next session!

Having trouble? Don't worry! Ask your professors, teaching assistants, or classmates for help. Getting your environment set up is the first important step. Good luck!