Unix - Shell Installation

By Dhandapani Yedappalli Krishnamurthi Sep 29, 2025

Install & use Unix (Linux) shells on Windows — step-by-step (fast, practical)

Below are the **best ways** (recommended → alternatives), with exact commands you can copy/paste. I focus first on **WSL (Windows Subsystem for Linux)** — the modern, integrated way to run a real Linux shell on Windows — then lighter-weight options (Git Bash, Cygwin, MSYS2) and a full VM option (VirtualBox + Ubuntu).

Option A — WSL (recommended)

Runs *real* Linux (bash, zsh, apt, system tools) inside Windows with minimal overhead. Works best on **Windows 11** or Windows 10 (2004+). Microsoft Learn+1

Quick overview (one-command install on modern Windows)

1. Open **PowerShell as Administrator**.
2. Run (recommended on Windows 11 — also works on many Windows 10 builds):

```
wsl --install
```

This enables the WSL feature, installs the default distro (usually Ubuntu), sets WSL2 as default, and installs the Linux kernel as needed. Restart if prompted. Microsoft Learn

To choose a distro:

```
wsl --list --online          # show available distros
wsl --install -d Ubuntu-22.04
```

To see installed distros and WSL versions: `wsl -l -v`. Microsoft Learn+1

Manual / older Windows 10 steps (if `wsl --install` not available)

1. Enable features (Admin PowerShell):

```
dism.exe /online /enable-feature
/featurename:Microsoft-Windows-Subsystem-Linux /all
/norestart
dism.exe /online /enable-feature
/featurename:VirtualMachinePlatform /all /norestart
```

2. Restart, then install your distro from the Microsoft Store (or use the `wsl --install -d <Distro>` helper if available). If converting existing distro to WSL2:

```
wsl --set-default-version 2
wsl --set-version <DistroName> 2
```

(Manual instructions for older builds are on Microsoft's WSL docs.) Microsoft Learn+1

First-run & essentials inside your new Linux shell

1. Launch the distro (Start menu or `wsl` in PowerShell / Windows Terminal). Create your Linux username/password when asked.
2. Always update packages:

```
sudo apt update && sudo apt upgrade -y
```

3. Install common dev tools:

```
sudo apt install -y build-essential git curl wget
vim zsh htop
```

4. Example — create & run a Hello World script:

```
cat > hello.sh <<'EOF'
#!/bin/bash
echo "Hello from WSL!"
EOF
chmod +x hello.sh
./hello.sh
```

Helpful WSL commands: `wsl --list --verbose`, `wsl --status`, `wsl --update`, `wsl --shutdown`. Microsoft Learn+1

Extra WSL tips & integration

- **Windows files** are mounted under `/mnt/<drive>`, e.g. `C:\Users\You` → `/mnt/c/Users/You`. Use that to access Windows files from WSL. For best performance, **keep heavy dev work inside the WSL filesystem** (e.g., `~/projects`) rather than working directly on `/mnt/c`, unless you need Windows apps to edit them. Microsoft Learn+1
- **GUI apps (WSLg)**: on Windows 11 WSL supports running Linux GUI apps (X11/Wayland) natively (WSLg). You can launch GUI Linux apps and they behave like Windows apps. Microsoft Learn
- **Editor integration**: use VS Code Remote — WSL extension to edit code inside WSL (fast + seamless). Visual Studio Code
- **Windows Terminal**: install Microsoft Windows Terminal for tabs & nice UX and create a profile for your distro. Install from Store or via `winget`. Microsoft Learn+1

Option B — Git Bash (Git for Windows) — lightweight, simple

- Installs a Bash shell (MSYS2-based) mainly intended for Git and basic UNIX tools. Good if you just want a familiar bash prompt on Windows without full Linux.
- Download & install: https://git-scm.com/downloads — then open **Git Bash** from Start menu. Git+1

Example use:

```
# in Git Bash:
echo "Hello from Git Bash"
bash hello.sh    # run script
```

Option C — Cygwin — large POSIX layer on Windows

- Provides many GNU tools and a POSIX environment. Use when you need many Unix utilities but can't use WSL. Installer: `setup-x86_64.exe` from cygwin.com. Cygwin+1

Option D — MSYS2 — developer toolchain (Pacman)

- Good for building native Windows binaries and uses pacman (Arch-like). Download/install from msys2.org; update base packages and then `pacman -Syu`. MSYS2+1

Option E — Full VM (VirtualBox + Ubuntu) — complete Linux desktop

- If you want a full separate Linux desktop, use VirtualBox (or VMware) and install Ubuntu/other distro as a VM. Good for isolated environments or GUI Linux desktop. VirtualBox downloads and Ubuntu + step-by-step guides are available. VirtualBox+1

Quick checklist — exact commands you'll use most

```
# (PowerShell as Admin) install WSL (Windows 11 / supported Win10)
wsl --install

# list available WSL distros
wsl --list --online

# set default version for future installs
wsl --set-default-version 2

# list installed distros + version
wsl -l -v

# update WSL itself (if recommended)
wsl --update
```

Inside Linux (Ubuntu):

```
# update distro packages
sudo apt update && sudo apt upgrade -y

# install dev tools
sudo apt install -y build-essential git curl zsh

# create & run hello script
echo -e '#!/bin/bash\necho Hello WSL' > hello.sh
chmod +x hello.sh
```

```
./hello.sh

# set zsh as your login shell (after installing zsh)
chsh -s $(which zsh)
# install Oh My Zsh (optional)
sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/ma
ster/tools/install.sh)"
```

(References: Microsoft WSL docs, Git for Windows, Cygwin, MSYS2, VirtualBox, VS Code Remote WSL.) Visual Studio Code+5Microsoft Learn+5Git+5

Troubleshooting & notes

- If `wsl --install` fails: ensure virtualization is enabled in BIOS/UEFI and that you have an up-to-date Windows build. For manual install enable `VirtualMachinePlatform` and the WSL optional feature (see docs). Microsoft Learn
- If file I/O is slow when working on projects on `/mnt/c`, move the project into your WSL home (`~/`) for speed. Microsoft Learn
- Want GUI apps? Use Windows 11 + WSLg or a VM. Microsoft Learn

---

Quick decision guide

- Want a **full Linux dev environment (best)** → **WSL2**. Microsoft Learn
- Need just Git + basic bash → **Git Bash**. Git
- Need many Unix tools but Windows-only host → **Cygwin / MSYS2**. Cygwin+1
- Need a full separate Linux desktop → **VirtualBox + Ubuntu**. Ubuntu

---