
Platform setup: Windows

Key takeaways

1. WSL2 is the recommended Windows path: it matches the Linux grading model.
2. MSYS2 gives bash + git + pacman on native Windows — five steps, one UCRT64 shell.
3. When Windows setup blocks progress, SSH to a department host instead.

Windows is fine; the shell must be Linux

- Windows can be a good laptop platform for this course, but the course command model is Linux-like.
- Use WSL2 if your laptop allows it. It gives you the same shell, paths, and home-directory model as the department hosts.
- Choose MSYS2 when your files and tools must live on the Windows side, or when WSL2 is blocked. It is a real course shell — if you follow its recipe exactly.

Windows decision tree

Pick the path that preserves one Linux-like workspace

Fastest in class

A department Linux host needs no install. SSH in from any terminal you already have and work there today.

SSH now; finish the laptop after class.

Best local path

WSL2 Ubuntu gives you the same shell, paths, and home directory the graders and the department hosts use.

Run every course command inside Ubuntu.

Native Windows path

MSYS2 is bash + git + pacman on native Windows — the choice when your files and tools live on the Windows side.

Five-step recipe; daily shell only — HW0 submits from WSL2.

Do not run the course setup from a random mix of PowerShell, Git Bash, WSL, and MSYS2.

WSL2: install, clone, run ./setup

One PowerShell command; everything after it happens inside Ubuntu

```
# In Windows PowerShell – installs the Ubuntu WSL2 distro:
wsl --install -d Ubuntu

# In the Ubuntu shell: bootstrap only – ./setup pins the rest
sudo apt update && sudo apt install -y git gh tmux curl
gh auth login          # one-time browser login
# clone your private homework repository
gh repo clone utac-f26/dotfiles-<login> ~/dotfiles
cd ~/dotfiles && ./setup  # installs pinned toolchain, links config
./self-check           # HW0 completion checklist
```

- Store course repos under /home/<you>, not /mnt/c/Users/<you>: Linux tools are faster and simpler there.
- Deck 901 is the full Linux / WSL2 walkthrough — from here on, the commands are identical to Linux.

MSYS2: bash + git + pacman on native Windows

Choose it when your files and tools must live on the Windows side

- MSYS2 is the complete package: pacman installs git, gh, tmux, and ripgrep, and its bash runs them against your real Windows files — no VM boundary.
- Git Bash cannot substitute: Git for Windows strips pacman, so it can never install the course toolchain. Once MSYS2 is set up, Git Bash goes unused.
- Five setup steps make it solid. Each one fails SILENTLY when skipped, so follow guides/windows.md in your dotfiles repo exactly.

One daily shell: git, tmux, and agents run in UCRT64 bash. The graded ./setup does not — HW0 submits from WSL2 or a department host.

The MSYS2 recipe — five steps, five silent failures

Exact commands live in your repo: `guides/windows.md` — your agent can read it too

Step	Skipping it silently gives you
1 · Install MSYS2; open 'UCRT64' shell	Same-looking bash, different runtime
2 · pacman installs the toolchain	No pacman: the toolchain never arrives
3 · db_home: /c/Users/%U (nsswitch.conf)	A shadow home under C:\msys64\home
4 · Dev Mode + winsymlinks:nativestrict	In -s copies the file and exits 0
5 · CLAUDE_CODE_GIT_BASH_PATH → MSYS2	Bash tool runs Git Bash; TMP/TEMP dropped

- A skipped step does not error — the shell mostly works and diverges later. Steps 3–5 change host state: restart every terminal (and agent).

MSYS2: verify, then clone

Each check catches one silently-skipped step — run them in a FRESH shell

```
echo $MSYSTEM                # UCRT64
cygpath -w ~                  # C:\Users\<you> — one home
# make a file, symlink it, confirm the link is real (not a copy)
touch t && ln -s t l && test -L l && echo real symlink

gh auth login                  # one-time browser login
gh repo clone utac-f26/dotfiles-<login> ~/dotfiles # your repo
# do NOT run ./setup here — it exits on MSYS2; submit HW0 from WSL2
```

- A stale terminal fails checks that a fresh one passes: setx values reach only process trees started after they were set.
- Deck 903 is the deep dive on why each check exists: two homes, crossed runtimes, silent symlink copies.

MSYS2 symlinks: enable Developer Mode first

Without it, `ln -s` silently copies — exit 0, no link, and the files diverge later

In -s under MSYS2	Developer Mode off	Developer Mode on
Default (MSYS unset)	Silent copy, exit 0	Silent copy, exit 0
MSYS=winsymlinks:nativestrict	Loud failure	Real native symlink

- Both switches, then a fresh shell: Developer Mode alone still copies, and the MSYS variable only counts at shell startup.
- Prefer the loud failure: a silent copy diverges the first time either file changes.
- Deck 903 tells the full story of this bug — the shadow home, the silent copy, and the re-test that exposed it.

Native Windows tools: use them as launchers

PowerShell and Windows Terminal are fine; the course shell should be WSL2 or MSYS2

- Windows Terminal
 - Good launcher for Ubuntu, PowerShell, and MSYS2 profiles.
- PowerShell
 - Good for installing WSL2 and checking Windows state; not the default place to run course setup.
- Git Bash
 - Another POSIX-on-Windows runtime, with no pacman — it can never install the course toolchain. Do not mix it with MSYS2: crossing msys runtimes silently drops TMP/TEMP.
- GitHub Desktop / Windows git.exe
 - Fine for other projects, but the course quickstart assumes gh inside the shell where setup runs.

Windows gotchas

Symptoms that usually mean you crossed runtime boundaries

Symptom	Likely cause	First check
Repo appears twice	Two homes or WSL plus Windows clone	<code>pwd; echo \$HOME; cygpath -w ~</code>
Command works in one terminal only	PATH differs by runtime	<code>which gh; echo \$PATH</code>
Agent cannot find files	Started from wrong shell or directory	<code>pwd; git status</code>
Slow WSL file operations	Repo under <code>/mnt/c</code>	Move repo to <code>/home/<you></code>
In <code>-s</code> made a copy, not a link	Developer Mode off or MSYS var unset	<code>test -L <link></code>
Version prints; command fails	Sandbox or cache one layer down	Run the real command; read its error

Use the department host when local Windows setup is burning class time. The first goal is a working course shell.

Key Takeaways

1. WSL2 is the recommended Windows path because it matches the Linux grading model.
2. MSYS2 is bash + git + pacman on native Windows: five steps, one UCRT64 shell for daily work — the graded receipt still comes from WSL2.
3. When local Windows setup blocks progress, SSH to the department host and keep moving.