LAB 0 CSCE 313 · Intro to Computer Systems

Setup Day

One Ubuntu VM. Eight steps. We do them together.

You will not write a line of C today. You will end the hour with a Linux machine you can log into — the machine every lab this semester runs on.

What we are building

Two computers, one desk. Your **laptop** is the host. Inside it runs a second, pretend computer — the **VM**, or guest. You talk to the guest through a terminal.

HOST — YOUR LAPTOP

macOS or Windows

- Hypervisor (VMware, UTM, VirtualBox)
- Terminal / PowerShell — where you type `ssh`
- VS Code, your browser, your notes

`ssh · port 22`

private network

GUEST — THE VM

Ubuntu Server LTS

- No desktop, no mouse — text only
- `gcc`, `make`, `gdb`, Wireshark
- Every assignment you submit is built here

First: which laptop are you on?

Before Step 1

Find your row and remember two words from it: your **hypervisor** and your **architecture**. Everything else today is the same for everyone.

Your machine	Hypervisor	ISO	How to check
Mac, Apple Silicon (M1–M4)	VMware Fusion, or UTM	`arm64`	Apple menu → About This Mac
Mac, Intel	VMware Fusion	`amd64`	Same place — says “Intel”
Windows / Linux, Intel or AMD	VMware Workstation Pro, or VirtualBox	`amd64`	Settings → System → About
Windows on ARM (Snapdragon)	Hyper-V, or VirtualBox ARM	`arm64`	About → “ARM-based processor”

ASK A TA Not sure which row is yours? That is a completely normal question — raise your hand now rather than after a 30-minute download.

STEP 1 OF 8 · DO THIS NOW

1 Install a hypervisor

A hypervisor is the app that runs a computer inside your computer. Download and install it, then stop — do not create anything yet.

VMware Fusion

Mac · recommended

Free for personal use. Needs a Broadcom account at `support.broadcom.com`, then My Downloads → search "Fusion".

Workstation Pro

Windows / Linux · recommended

Same free Broadcom download, search "Workstation". Windows has two pre-flight checks — next slide.

UTM · VirtualBox

Escape hatch · no account

UTM (Mac) and VirtualBox (Windows) are free with no signup. A little slower, perfectly fine for this course.

Storage check while it downloads: you want **80–100 GB free** on the host. An external APFS or exFAT SSD works too — never NTFS on a Mac, and never unplug it while the VM runs.

Where the download actually lives

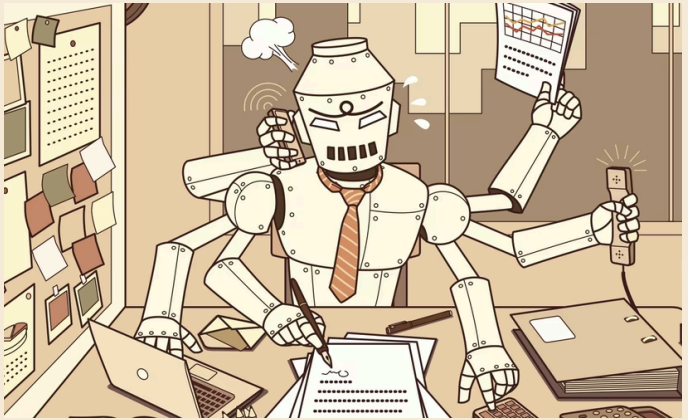
VMware is free for personal use, but it now sits behind a Broadcom account. The path is not obvious — follow these in order. **Workstation Pro** for Windows and Linux, **Fusion Pro** for Mac; the steps are identical apart from the product name.

THE SIX CLICKS

1. **Log in at** `support.broadcom.com` — create a free account if you do not have one.
2. **Use the direct link** Download VMware Workstation Pro (or Fusion Pro) once you are logged in.
3. **Or navigate there:** My Dashboard → My Downloads → Free Software Downloads available HERE → the product.
4. **Expand the product row** named “VMware Workstation Pro <version> for <your OS>”.
5. **Pick a version** from the Release column.
6. **Tick the Terms and Conditions** — this is what activates the download link. Nothing downloads until it is checked.

IF THE LINK LOOKS DEAD

It almost always means step 6. The Terms and Conditions box has to be reviewed and checked before the link becomes clickable — it is not a broken page.



Windows only: two blockers

Side quest · Mac users, relax

If your VM refuses to power on, it is almost always one of these two. Neither is your fault, and both are a settings toggle plus a reboot.

BLOCKER 1 · FIRMWARE

Virtualization is off in BIOS

```
This host supports Intel VT-x,
but VT-x is disabled
```

Reboot into BIOS/UEFI — **F2**, **F10** or **Del** at power-on, or Settings → System → Recovery → Advanced startup → UEFI Firmware Settings. Enable **Intel VT-x** / **AMD-V** (sometimes “SVM Mode”).

BLOCKER 2 · WINDOWS SECURITY

Hyper-V got there first

```
VMware Workstation and Device/
Credential Guard are not compatible
```

Settings → Privacy & Security → Windows Security → Device Security → Core Isolation → **Memory integrity off**, then reboot. Using WSL2 or Docker? Keep Hyper-V and live with the slower mode instead.

STEP 2 OF 8 · DO THIS NOW

2 Download the Ubuntu Server ISO

Go to `ubuntu.com/download/server` and take an **LTS** release. Server, not Desktop — there is no graphical desktop and that is deliberate.

WHY SERVER?

Less to download, less to break, and the terminal is the whole point of this course. You will add a lightweight desktop later, in Step 6, only for Wireshark.

The file is about 3 GB. Start it now and keep listening — the next steps do not need it finished.

https://ubuntu.com/download/server ---> Intel / AMD
https://ubuntu.com/download/server/arm ---> ARM

THE #1 MISTAKE OF SETUP DAY

Get the correct ISO File

It refuses to boot, or crawls so slowly you will blame your laptop. Thirty minutes lost. Apple Silicon takes **arm64**; everyone else takes **amd64**.

```
$ uname -m
aarch64  # Apple Silicon
x86_64   # Intel / AMD
```

You will run this inside the VM later. If it disagrees with your laptop, you downloaded the wrong ISO — delete the VM and start over. It happens; it is a ten-minute redo, not a disaster.

3

STEP 3 OF 8 · DO THIS NOW

Create the virtual machine

In your hypervisor: **New** → point it at the ISO you just downloaded → set these three numbers → do not start the install yet.

4 GB

RAM, minimum

Give it 8 GB if your laptop has 16 GB or more.

64 GB

disk, minimum

Thin-provisioned: it grows to 10–20 GB in practice.

NAT

networking

The default. Leave every other setting alone.

Laptop fans taking off? Close your browser tabs during the install. After setup, 4 GB is comfortable — and **suspend** the VM between sessions instead of shutting it down, so it comes back in seconds.

Make an SSH key while you wait

Before Step 4 · pays off twice

Put a key on GitHub now and the Ubuntu installer can pull it in for you — the "Import SSH identity: from GitHub" box on the next slide. Same key gets you passwordless `git clone` later. Ten minutes now, no passwords all semester.

1 · MAKE THE PAIR

On your **laptop** — Terminal on Mac, PowerShell on Windows. Press Enter at all three prompts:

```
ssh-keygen -t ed25519 \
  -C "netid@tamu.edu"
```

Two files appear in `~/.ssh`: `id_ed25519` (private — never share it) and `id_ed25519.pub` (public — share freely).

2 · COPY THE PUBLIC HALF

```
# Mac / Linux
cat ~/.ssh/id_ed25519.pub

# Windows PowerShell
type
$env:USERPROFILE\.ssh\id_ed25519.pub
```

It is one long line starting `ssh-ed25519 AAAA…` and ending with your email. Select the whole thing.

3 · PASTE IT INTO GITHUB

github.com → your avatar → **Settings** → **SSH and GPG keys** → **New SSH key**.

Title it anything ("my laptop"), leave the type as **Authentication key**, paste, save.

Check it: `ssh -T git@github.com` should greet you by name.

Only ever paste the `.pub` file. If what you copied contains the words `PRIVATE KEY`, you have the wrong file — start again with the `.pub`.

Already have a key on GitHub? Nothing to do — just have your GitHub username ready for the installer, or run `ssh-import-id gh:yourname` in the VM afterwards.

STEP 4 OF 8 · DO THIS NOW

4 Install Ubuntu — accept the defaults

The installer is a blue text screen. Arrow keys move, **Enter** selects, **Tab** jumps to Done. Press Enter through language, keyboard, network, proxy, mirror and storage.

WRITE THESE DOWN

The **username** and **password** you pick on the profile screen. You will need both in about ten minutes, and there is no “forgot password” link.

The install takes a few minutes and then asks to reboot. If it boots back into the installer, the virtual disc is still attached — VM Settings → CD/DVD → disconnect, reboot.

THE ONE SCREEN THAT MATTERS — SSH SETUP

[X] Install OpenSSH server ← check this

[X] Import SSH identity: **from GitHub**
optional — type your GitHub username and it pulls your public keys in for you

[ ] Disallow password authentication
leave this UNCHECKED — see the next slide

Missed the SSH screen entirely? Nothing is lost — you can install the server afterwards from inside the VM. We will cover that too.

If you checked that last box

Recoverable · nobody is locked out

“Disallow password authentication” means SSH will only accept your key. With the right password, you still get this — and it is the clearest error in Linux once you know how to read it.

```
you@host ~ % ssh student@192.168.80.164
student@192.168.80.164: Permission denied (publickey).
```

FIX A · USE THE MATCHING KEY

The private key paired with the GitHub key you imported:

```
ssh -i ~/.ssh/id_ed25519 user@<vm-ip>
```

FIX B · TURN PASSWORDS BACK ON

From the hypervisor console, which always works:

```
sudo nano /etc/ssh/sshd_config.d/50-cloud-
init.conf
# set: PasswordAuthentication yes
sudo systemctl restart ssh
```

While that installs: nine commands

Terminal primer

The mental model: you are always *somewhere* in the filesystem, and commands act on where you are. That is genuinely most of it.

`pwd`

Where am I?

`ls -la`

What is here, hidden files included

`cd src-cloud/`

Go there. `cd ..` goes back up

`cat file`

Print a file to the screen

`nano file`

Edit it. `Ctrl+O` save, `Ctrl+X` quit

`sudo …`

Do it as administrator. Asks your password

`Tab`

Autocomplete. Use it constantly — it prevents typos

`Ctrl+C`

Stop whatever is running. Your escape key

`↑ ↓`

Previous commands. Never retype a long one

Try it: run `yes` and watch it print forever, then stop it with **Ctrl+C**. You just sent your first signal — a topic we come back to properly in a few weeks.

STEP 5 OF 8 · DO THIS NOW

5 Reboot and log in

```
Ubuntu 26.04 LTS csce313 tty1

csce313 login: student
Password:

(nothing appears as you type —
that is normal, keep going)

student@csce313:~$
```

That prompt is the whole course

Your name, the machine's name, where you are (~ = your home folder), and a $ meaning “your turn”.

Booted into the installer again?

The virtual disc is still in the drive. VM Settings → CD/DVD → disconnect the ISO → reboot.

Paste does not work here

This console has no clipboard yet, limited scrollback, and it traps your cursor. Type the next commands by hand — or jump ahead to SSH, which fixes all three.

STEP 6 OF 8 · DO THIS NOW

6 Install the course packages

Six lines, in this order, inside the VM. One at a time — read what each one says before the next.

```
$ curl -o src-cloud.zip https://seed.nyc3.
  cdn.digitaloceanspaces.com/src-cloud.zip
$ sudo apt update
$ sudo apt -y install unzip
$ unzip src-cloud.zip
$ cd src-cloud/
$ ./install.sh
```

The URL is one line with no space — the wrap above is just this slide being narrow.

DO NOT WALK AWAY

It runs for several minutes and stops twice to ask you a question. Answers on the next slide — get them wrong and things break quietly.

WHAT YOU ARE GETTING

The compiler toolchain (`gcc`, `make`, `gdb`), Wireshark for the networking labs, and a lightweight xfce4 desktop so Wireshark has somewhere to draw.

Seeing `Could not get lock`? Ubuntu is doing its own updates in the background. Wait two minutes and retry — never delete a lock file.

The two questions it will ask

DURING WIRESHARK

"Should non-superusers be able to capture packets?"

No — usually the default — just press Enter

Capturing then needs `sudo`, which is the locked-down setup we want.

DURING XFCE4

"Choose the default display manager"

LightDM

Not `gdm3`. Lighter, and it is what the rest of the course assumes.

Never saw the second question? Also fine.

Recent Ubuntu often decides silently. What matters is the result — check it, and fix it if needed:

```
$ cat /etc/X11/default-display-manager
/usr/sbin/lightdm   # what you want to see

$ sudo dpkg-reconfigure lightdm   # run this if it says gdm3
```

STEP 7 OF 8 · DO THIS NOW

7

SSH in from your own terminal

INSIDE THE VM — FIND YOUR ADDRESS

```
$ hostname -I
192.168.80.164
```

Write it down. `ip a` shows the same thing with more detail.

ON YOUR LAPTOP — TERMINAL OR POWERSHELL

```
$ ssh student@192.168.80.164
```

First time it asks about a fingerprint — type `yes`. Windows needs no PuTTY; `ssh` ships built in.

Why bother, when the console works?

Copy and paste. Real scrollback. Your own font size. Several windows at once. And it is how you will work all semester — including on remote machines you cannot see.

Next week it will "break"

NAT hands out a new address after reboots. When SSH suddenly times out, run `hostname -I` in the console first — that is the fix nine times out of ten.

Stop typing your password

Optional · recommended

A key pair is two files: a **private** key that never leaves your laptop, and a **public** key you hand out freely. The VM keeps a list of public keys it trusts.

EASIEST · ALREADY ON GITHUB

Run this *inside the VM* and it pulls in every public key on your GitHub account:

```
ssh-import-id gh:yourname
```

The same thing the installer's GitHub import did — just later.

FROM SCRATCH · MAC / LINUX

On your *laptop*. Press Enter three times to accept the defaults:

```
ssh-keygen -t ed25519 \
  -C "netid@tamu.edu"
ssh-copy-id user@<vm-ip>
ssh user@<vm-ip>
```

No password prompt on that last line = it worked.

FROM SCRATCH · WINDOWS

`ssh-copy-id` does not exist here. After `ssh-keygen`, in PowerShell:

```
type $env:USERPROFILE\.ssh\id_
ed25519.pub | ssh user@<vm-ip>
"mkdir -p ~/.ssh && cat >> ~/.
ssh/authorized_keys"
```

STEP 8 OF 8 · OPTIONAL BUT LOVELY

8 Edit code in VS Code, run it on the VM

If `vi` is not your idea of a good time, this is the alternative. Your files live on the VM; the editor runs on your laptop.

1. VS Code → Extensions (the stacked-squares icon) → search "SSH" → install **Remote – SSH** by Microsoft.
2. `Cmd/Ctrl + Shift + P` → "Remote-SSH: Connect to Host".
3. Enter `user@<vm-ip>`, then your password. A new window opens, attached to the VM.

WORTH FIVE MINUTES: NAME YOUR VM

On your laptop, add this to `~/.ssh/config`:

```
Host csce313
    HostName 192.168.80.164
    User student
    IdentityFile ~/.ssh/id_ed25519
```

Now `ssh csce313` works, and the VM shows up by name in VS Code. When the IP changes, you edit one line.

Compiling still happens on the VM, in the VS Code terminal. Same `gcc`, nicer window.

Git, the four lines that matter today

Bonus round

Run all of this **inside the VM** — that is where your code will live. Git is already installed.

1 · TELL GIT WHO YOU ARE

```
git config --global user.name "Your Name"
git config --global user.email "netid@tamu.edu"
```

2 · A KEY GITHUB WILL TRUST

```
ssh-keygen -t ed25519 -C "netid@tamu.edu"
cat ~/.ssh/id_ed25519.pub
```

Copy that whole line into GitHub → Settings → SSH and GPG keys → New SSH key.

3 · CHECK IT TOOK

```
ssh -T git@github.com
Hi yourname! You've successfully
authenticated…
```

It says it does not provide shell access. That is the correct answer, not an error.

4 · CLONE THE COURSE REPO

```
git clone git@github.com:<org>/<repo>.git
cd <repo> && ls
```

Your TA will give you the exact address. Files appear — you are set up.

You are done when all of these pass

Checkpoint · run them in order

`uname -m`	matches your CPU — `aarch64` or `x86_64`
`lsb_release -a`	Ubuntu, an LTS release
`free -h`	total memory ≥ 3.8 Gi
`df -h /`	root filesystem around 60 GB
`ip a`	a non-loopback `inet` address
`systemctl is-active ssh`	`active`
`dpkg -l | grep -c xfce4`	a number greater than zero
`cat /etc/X11/default-display-manager`	`/usr/sbin/lightdm`
`which wireshark tshark`	both print a path
`gcc --version && make --version`	both print versions

`ssh user@<vm-ip> 'echo ok'` — **From your laptop, not the VM.** Prints `ok` — this one proves the whole chain works end to end.

“SSH doesn't work”

Triage · in this order

Before anything else: **read the error out loud**. Four different messages mean four unrelated problems, and the words tell you which.

`Permission denied (before password)`

Your *laptop* is blocking it — macOS Local Network permission, a firewall, a VPN, or campus Wi-Fi blocking port 22. Test on a phone hotspot.

`Connection refused`

Nothing is listening. In the VM: `sudo apt -y install openssh-server && sudo systemctl enable --now ssh`

`Permission denied (publickey)`

Never asked for a password → password auth is off. Use `ssh -i` with the matching key, or re-enable passwords from the console.

`Timeout, or it worked yesterday`

The VM's IP changed. `hostname -I` in the console, then update your command or SSH config.

THE THREE CHECKS THAT SOLVE NINE IN TEN

1. Which error, exactly?

2. `hostname -I` — is the IP what you think?

3. `systemctl is-active ssh` — is it even running?

Appendix: install & network

VM crawls, or the ISO won't boot
Wrong architecture. Delete the VM, get the arm64 ISO, reinstall.

Paste doesn't work in the console
No clipboard without a GUI. SSH in from your own terminal instead.

Boots into the installer again
Installer disc still attached. VM Settings → CD/DVD → disconnect.

`Unable to locate package`
Run `sudo apt update` first. Still failing? Check `ip a` and the VM's network adapter.

`Could not get lock … dpkg`
Ubuntu is updating in the background. Wait 2–5 minutes, retry. Never delete lock files.

`curl: (6) Could not resolve host`
`ping -c1 1.1.1.1` then `ping -c1 ubuntu.com`. DNS only? `sudo systemctl restart systemd-resolved`.

`REMOTE HOST IDENTIFICATION HAS CHANGED`
You rebuilt the VM. On the laptop: `ssh-keygen -R <vm-ip>`, reconnect, accept.

Only ~30 GB of a 64 GB disk
LVM didn't claim it all: `sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv && sudo resize2fs …`

Display-manager prompt never appeared
Fine if the result is right. Check the file; if it says `gdm3`, run `sudo dpkg-reconfigure lightdm`.

Appendix: Windows, disks & drives

(Windows) `VT-x is disabled`
Virtualization is off in firmware. Enable Intel VT-x / AMD-V in BIOS/UEFI, then reboot.

(Windows) `Device/Credential Guard`
Turn Memory Integrity off and reboot — or keep Hyper-V and accept the slower mode if you need WSL2 or Docker.

(Windows) `ssh-copy-id: not found`
It doesn't exist on Windows. Use the `type … | ssh …` one-liner, or `ssh-import-id gh:name` inside the VM.

Laptop out of disk, mid-semester
Delete old snapshots, empty the VM's trash, or move the VM to an external APFS/exFAT SSD while it is powered off.

External-drive VM won't start
NTFS is read-only on macOS, or the drive was unplugged while the VM ran. Reformat APFS or exFAT.

Fans roaring, 4 GB feels tight
Close host browsers during install; suspend rather than shut down between sessions.

Symptom not listed? Bring the exact error text to lab hours — the wording is what tells us which of these it actually is.

Finished early? Go and play.

All harmless, all removable with `apt remove` — and each one teaches you something about the shell.

`apt moo`	Super cow powers. Try adding more `moo`s.
`sudo apt install sl`	Now mistyping `ls` sends a train past.
`fortune | cowsay`	Your first pipe, doing something silly.
`figlet CSCE 313`	ASCII banners. Pipe it into `cowsay -n`.
`fastfetch`	A brag card — and a quick sanity check.
`cmatrix`	Instant credibility. Quit with `q`.

Not finished? Also fine.

Nothing here is graded. Bring your laptop to lab hours and we will pick up exactly where you stopped — the setup document has every command from today, in order.

QR to the setup doc

Everything from today

Installation and Configuration of Linux Ubuntu Server — on the course page.

Next week

Your first C program on the VM: `gcc`, `make`, and what actually happens between your source file and a running process.