

Project Simulator gem5 Setup Instructions

All projects in this course will use the event-based computer architecture simulator 'gem5'. In order to use the 'gem5' simulator for this course, it must be downloaded and run on a virtual machine. In this course, in addition to the instructional PDFs, you will be downloading 2 important zip files. The first file, **gem5.zip**, contains all the code necessary to run the simulator for this course's specifications. Do not download any 'gem5' files from the 'gem5' website or its GitHub, as they are frequently updated and may not match the specifications of this course. The second file, **benchmarks.zip**, contains all of the benchmark code to run tests. While not needed to complete the setup, here is the gem5 website link for further info: <https://www.gem5.org/>.

NOTE: This setup procedure has three components:

- 1) Setting up the VM - this part can be completed in about 15-25 minutes.
- 2) Setting up the gem5 simulator - this part can take as short as 20-30 minutes or as long as 1.5 - 2 hours, depending on the computer and VM configuration. This component must be done in a single, uninterrupted step (can't be stopped and resumed later).
- 3) Testing the gem5 environment - this part can be completed in about 5-10 minutes.

Therefore, please allocate about 3 hours of uninterrupted time to complete setting up gem5. Almost everyone should finish the entire setup process (this document) within 3 hours, with a majority of people finishing much quicker.

Part 1: Setting up the VM

The gem5 simulator for this course must run on Ubuntu 20.04.5. There are multiple options for running gem5 on Ubuntu depending on whether you are running a Windows, MacOS, or Linux operating system. The two recommended options for this course include installing a hypervisor (e.g. VMWare, VirtualBox) on your desktop or using a cloud-based VM (GCP) through your browser. However, many students have been able to complete the projects using these alternative options:

- 1) Install the Docker image from gem5 to run locally and containerized:
https://www.gem5.org/documentation/general_docs/building#docker

- 2) Setting up an Windows Subsystem for Ubuntu with GUI support:
<https://ubuntu.com/wsl>
- 3) Dual Booting Ubuntu from their local Desktop.

Please try the alternative options **only if you are experienced** and prefer these to the recommended VM options. While you will probably be able to complete the projects without incidents, if you do run into any issues, the course staff may not be able to help troubleshoot or assist you with any issues that affect your hardware. This document will provide information to walk you through the steps to set up the VM using the two recommended options.

Recommended VM provisioning instructions:

1. **Installing a Hypervisor**

NOTE: If you already have a hypervisor installed, DO NOT USE AN EXISTING VM OR UBUNTU .iso FILE. You must create a new VM and download and install a NEW Ubuntu 20.04 .iso file.

- i. Download a hypervisor, either VMWare or VirtualBox. This option is compatible with Windows, MacOS (Intel-based), and Linux.
- ii. Download the Ubuntu 20.04 .iso file from Canvas or the ubuntu website:
<https://releases.ubuntu.com/focal/>
- iii. Install the downloaded Ubuntu 20.04 .iso file onto the VM.

If you need a step-by-step walkthrough for the VMWare installation, here is a link:

<https://getlabsdone.com/10-easy-steps-to-install-ubuntu-19-04-on-vmware-workstation-15/>

If you need a step-by-step walkthrough for the VirtualBox installation, here is a link:

<https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview>

- iv. Once downloaded, in the hypervisor settings please allocate at least 40GB of Disk Space (40GB recommended) and at least 4GB of RAM (8GB recommended). More RAM and Disk allocation will result in faster compilation and fewer issues.
- v. Loading Project Files: After starting up the VM, use a web browser to log into the course Canvas page, download all the project files for project 1 (specifically **benchmarks.zip** and **gem5.zip**) and unzip them on the VM itself.

Now the setup of the VM is complete. You can proceed to “**Part 2: Setting up gem5**”.

2. Provisioning a GCP (Google Cloud Platform) VM

- i. Follow the step-by-step walkthrough instructions in the “GCP Setup for gem5” document on Canvas.

Remember to stop your GCP VM when you are not using it to avoid incurring charges. Also, remember to terminate your instance at the end of the course to avoid incurring charges, if you do not want to keep your course materials. You can always download your work from GCP before terminating.

Part 2: Setting up gem5:

DISCLAIMER: the gem5 setup process will likely take a few hours.

After unzipping the files “gem5.zip” and “benchmarks.zip”, you may notice that gem5.zip unzipped to a folder named gem5-21.0.0.0. Don’t be alarmed, as this is the version of gem5 that was zipped and used for this course. You can change its name with the command: `mv gem5-21.0.0.0 gem5`

After accessing the VM, type the following 3 commands to install the necessary dependencies.

*****NOTE***:** Copying and pasting the following commands may not work due to PDF formatting. Either type it manually or carefully check it for line breaks after pasting.

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install build-essential git m4 scons zlib1g zlib1g-dev  
libprotobuf-dev protobuf-compiler libprotoc-dev  
libgoogle-perftools-dev python3-dev python-is-python3  
libboost-all-dev pkg-config
```

Now, enter the gem5 directory with this command:

```
cd gem5
```

Now we must build the gem5 simulator for the target processor. Here, we choose the ARM processor, which will be used for all the projects for this class. gem5 also supports other ISAs such as x86, MIPS etc., which we will not use.

*****NOTE***:** The following single command takes the bulk of the time, as it compiles and builds all of the necessary files for the ARM platform for simulation. It can be as fast as 20 minutes on some VMs or as slow as 2 hours in others. **This step must be done continuously and cannot be interrupted. Please run it only when you are ready.**

Now run:

```
scons build/ARM/gem5.opt
```

Now we wait for the simulator to finish building. During the build phase, you may receive many “warnings”. You can ignore those for now. As long as you do not get any “errors”, you have successfully completed the build for gem5.

Part 3: Testing the gem5 environment and setup:

Once the gem5 simulator is built for the ARM processor, we can simulate the pre-compiled sample test program “hello world” on gem5. To do so, type on the VM terminal:

```
build/ARM/gem5.opt ./configs/example/se.py -c  
tests/test-progs/hello/bin/arm/linux/hello
```

You will see output Hello world! on the console. Output statistics are created in the directory m5out, by default. We will explore later various command-line options for gem5 simulation. Now that the installation, compilation, build, and the testing of the gem5 simulator has been verified, you can begin to work on the projects for this course.