Assignment 1: GitHub Account and gem5 Installation

Name: Bijay KC

Computer Architecture and Design (MSCS-531-M50)

Date: January 15th 2026

1. Introduction to gem5

gem5 is an open-source platform used for computer architecture research. It's a simulator that allows us to model different hardware designs in great detail. I learned that it actually came from two different projects, M5 and GEMS, which merged to create a tool that handles both the CPU and the memory system really well.

It supports different architectures like X86 and ARM. In research, it is used to test new ideas for processor designs (like caches or branch predictors) without needing to build real hardware. It can run in Syscall Emulation (SE) mode for simple programs or Full System (FS) mode to run a whole operating system.

2. Environment Setup

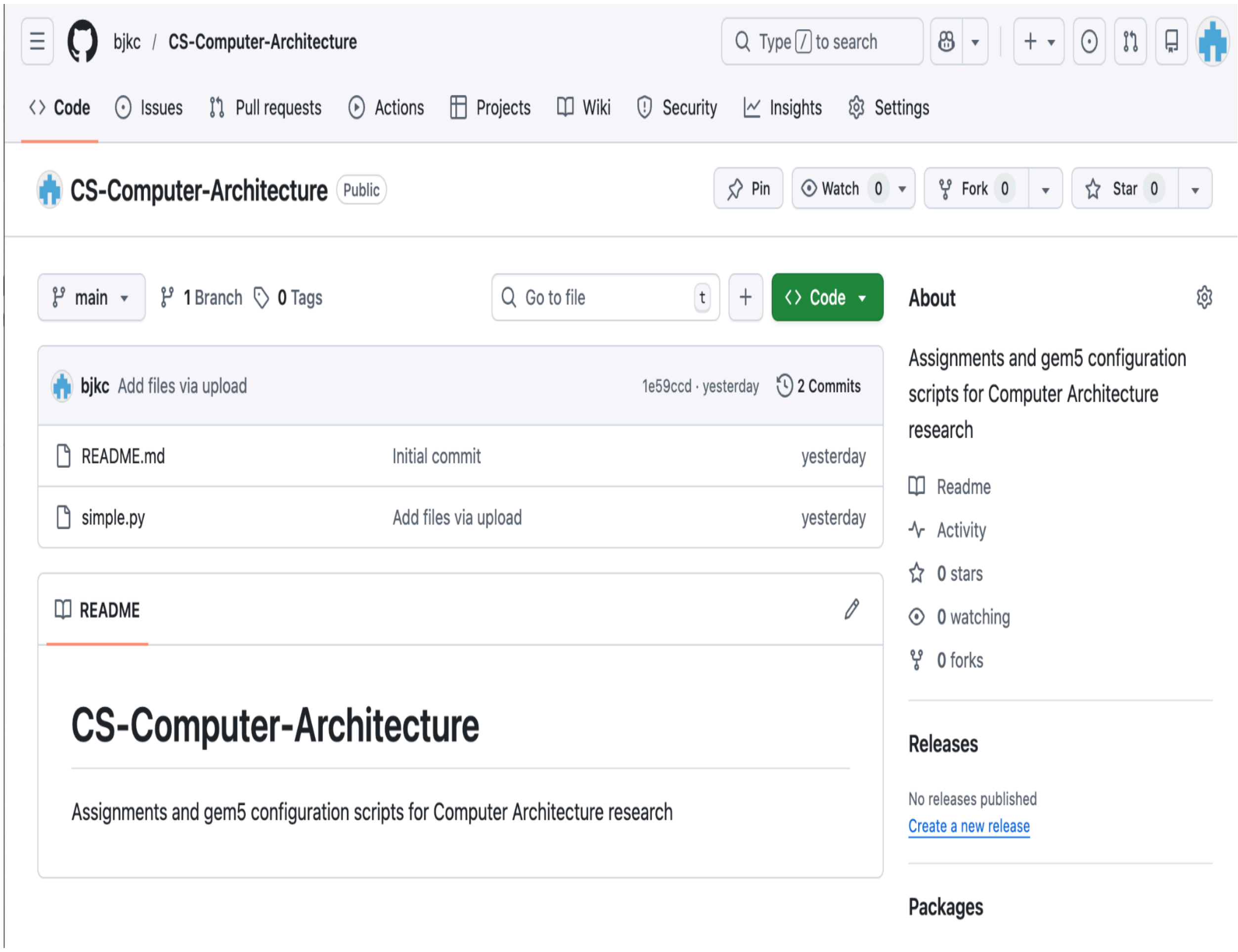
Since I am using a Mac, I had to set up my environment using Homebrew. I followed these steps:

- Prerequisites: I installed the necessary tools using the command:
  ```
  brew install scons python3 boost hdf5 protobuf google-perftools libpng
  ```
- Cloning the Repository: I downloaded the gem5 source code from GitHub:
  ```
  git clone https://github.com/gem5/gem5.git
  ```
- Git Configuration: I also linked my terminal to my GitHub account by running:
  ```
  git config --global user.email "[kcb17@yahoo.com]"
  git config --global user.name "Bijay KC"
  ```

---

3. GitHub Documentation

I created a GitHub account and set up a repository called **CS-Computer-Architecture** to organize my work.



bjkc Add files via upload 1e59ccd · yesterday 2 Commits

README.md	Initial commit	yesterday
simple.py	Add files via upload	yesterday

README

CS-Computer-Architecture

Assignments and gem5 configuration scripts for Computer Architecture research

About: Assignments and gem5 configuration scripts for Computer Architecture research

No releases published

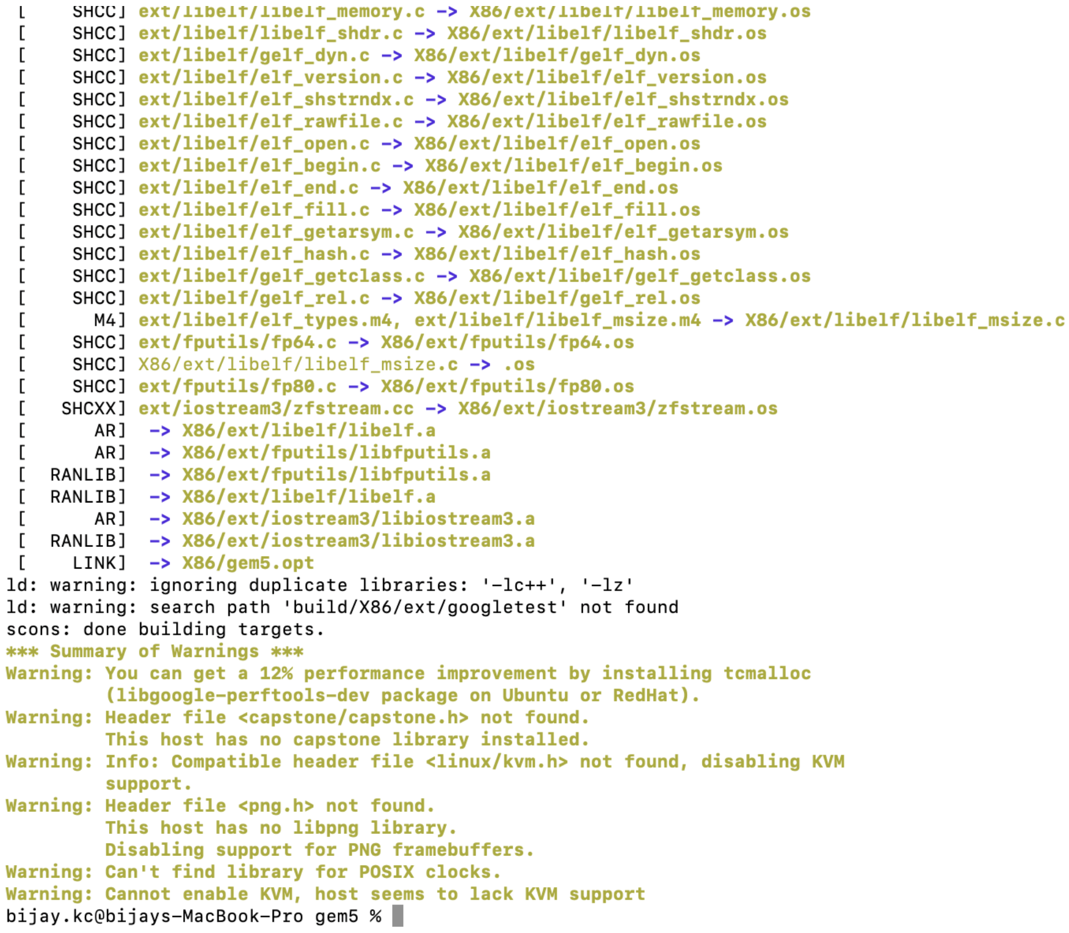
I also uploaded the `simple.py` configuration script to this repository to ensure my simulation parameters are version-controlled and accessible for future research.

4. Building gem5

I moved into the gem5 directory and started the build for the **X86** architecture. I used the `gem5.opt` version because it is optimized for performance while still being useful for debugging.

- **The Command:** `scons build/X86/gem5.opt -j 4`

The build process took about 23 minutes. Even though I saw some warnings about KVM and PNG support, these are normal for macOS and didn't stop the build from finishing.



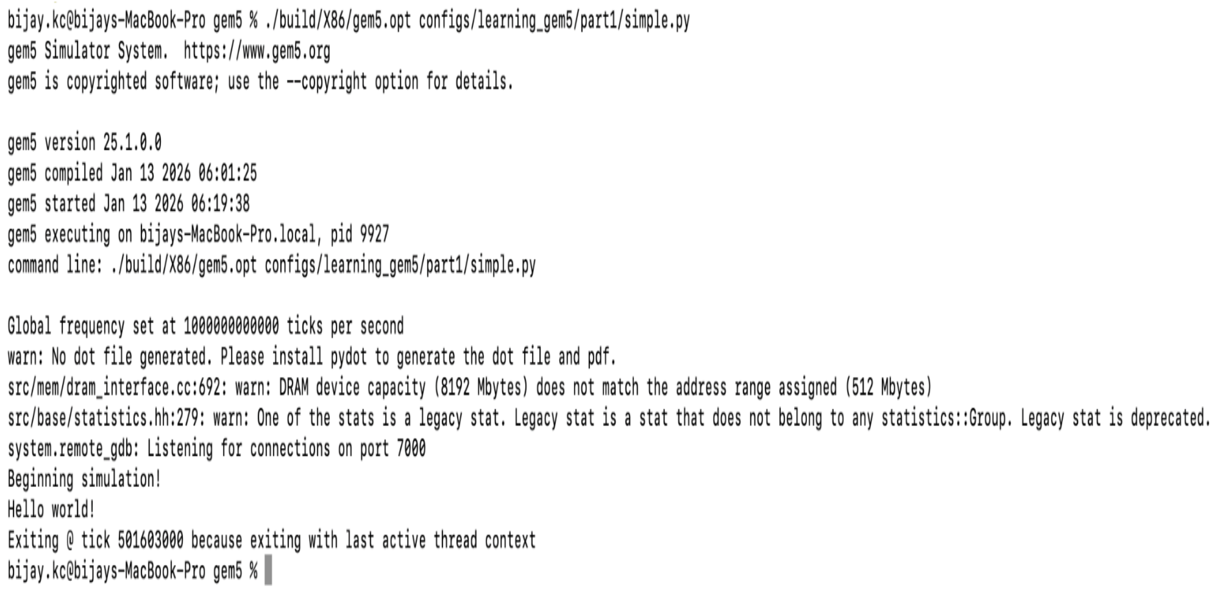
```
[     SHCC] ext/libelf/libelf_memory.c -> X86/ext/libelf/libelf_memory.os
[     SHCC] ext/libelf/libelf_shdr.c -> X86/ext/libelf/libelf_shdr.os
[     SHCC] ext/libelf/gelf_dyn.c -> X86/ext/libelf/gelf_dyn.os
[     SHCC] ext/libelf/elf_version.c -> X86/ext/libelf/elf_version.os
[     SHCC] ext/libelf/elf_shstrndx.c -> X86/ext/libelf/elf_shstrndx.os
[     SHCC] ext/libelf/elf_rawfile.c -> X86/ext/libelf/elf_rawfile.os
[     SHCC] ext/libelf/elf_open.c -> X86/ext/libelf/elf_open.os
[     SHCC] ext/libelf/elf_begin.c -> X86/ext/libelf/elf_begin.os
[     SHCC] ext/libelf/elf_end.c -> X86/ext/libelf/elf_end.os
[     SHCC] ext/libelf/elf_fill.c -> X86/ext/libelf/elf_fill.os
[     SHCC] ext/libelf/elf_getarsym.c -> X86/ext/libelf/elf_getarsym.os
[     SHCC] ext/libelf/elf_hash.c -> X86/ext/libelf/elf_hash.os
[     SHCC] ext/libelf/gelf_getclass.c -> X86/ext/libelf/gelf_getclass.os
[     SHCC] ext/libelf/gelf_rel.c -> X86/ext/libelf/gelf_rel.os
[       M4] ext/libelf/elf_types.m4, ext/libelf/libelf_msize.m4 -> X86/ext/libelf/libelf_msize.c
[     SHCC] ext/fputils/fp64.c -> X86/ext/fputils/fp64.os
[     SHCC] X86/ext/libelf/libelf_msize.c -> .os
[     SHCC] ext/fputils/fp80.c -> X86/ext/fputils/fp80.os
[    SHCXX] ext/iostream3/zfstream.cc -> X86/ext/iostream3/zfstream.os
[       AR]  -> X86/ext/libelf/libelf.a
[       AR]  -> X86/ext/fputils/libfputils.a
[   RANLIB]  -> X86/ext/fputils/libfputils.a
[   RANLIB]  -> X86/ext/libelf/libelf.a
[       AR]  -> X86/ext/iostream3/libiostream3.a
[   RANLIB]  -> X86/ext/iostream3/libiostream3.a
[     LINK]  -> X86/gem5.opt
ld: warning: ignoring duplicate libraries: '-lc++', '-lz'
ld: warning: search path 'build/X86/ext/googletest' not found
scons: done building targets.
*** Summary of Warnings ***
Warning: You can get a 12% performance improvement by installing tcmalloc
         (libgoogle-perftools-dev package on Ubuntu or RedHat).
Warning: Header file <capstone/capstone.h> not found.
         This host has no capstone library installed.
Warning: Info: Compatible header file <linux/kvm.h> not found, disabling KVM
         support.
Warning: Header file <png.h> not found.
         This host has no libpng library.
         Disabling support for PNG framebuffers.
Warning: Can't find library for POSIX clocks.
Warning: Cannot enable KVM, host seems to lack KVM support
bijay.kc@bijays-MacBook-Pro gem5 % 
```

5. Running the Simulation

To test the build, I ran a "Hello World" simulation.

- **The Command:** `./build/X86/gem5.opt configs/learning_gem5/part1/simple.py`

Results: The simulation successfully initialized the virtual CPU and printed "Hello world!" as expected. It reached the exit event at **tick 501603000**.



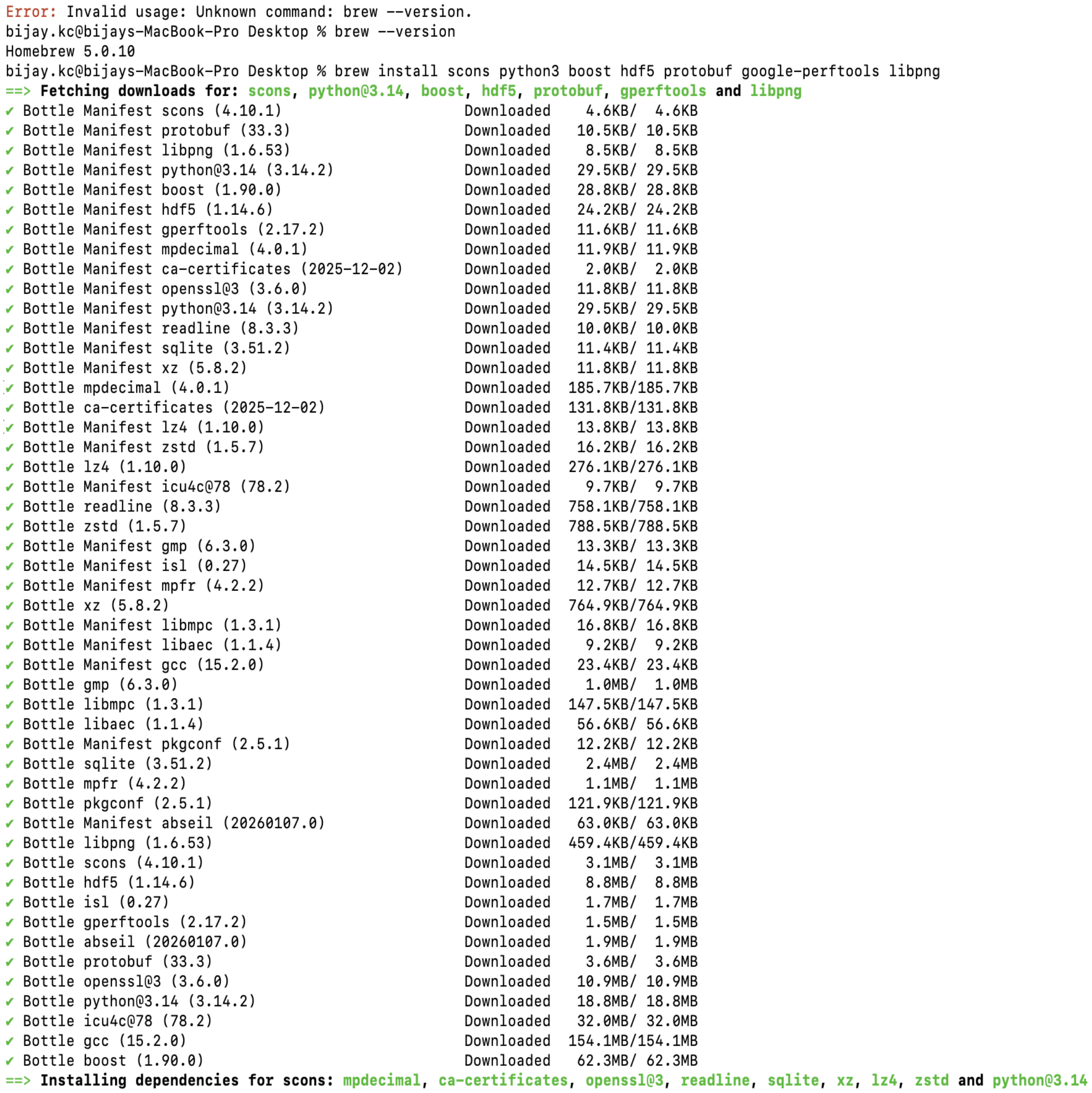
```
bijay.kc@bijays-MacBook-Pro gem5 % ./build/X86/gem5.opt configs/learning_gem5/part1/simple.py
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 25.1.0.0
gem5 compiled Jan 13 2026 06:01:25
gem5 started Jan 13 2026 06:19:38
gem5 executing on bijays-MacBook-Pro.local, pid 9927
command line: ./build/X86/gem5.opt configs/learning_gem5/part1/simple.py

Global frequency set at 1000000000000 ticks per second
warn: No dot file generated. Please install pydot to generate the dot file and pdf.
src/mem/dram_interface.cc:692: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
src/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy stat is deprecated.
system.remote_gdb: Listening for connections on port 7000
Beginning simulation!
Hello world!
Exiting @ tick 501603000 because exiting with last active thread context
bijay.kc@bijays-MacBook-Pro gem5 %
```

6. Troubleshooting and Reflection

- When I was first installing Homebrew, I got an error saying it didn't recognize the `brew --version` command. I thought I might have messed up the installation. But after I found out, the terminal just needed a second to update itself. Once I refreshed the settings, it worked perfectly and showed I had version 5.0.10. This was a good lesson for me as I learned that even if a command fails once, it doesn't always mean the process will stop; sometimes the computer just needs a moment to find the new paths.



```
Error: Invalid usage: Unknown command: brew --version.
bijay.kc@bijays-MacBook-Pro Desktop % brew --version
Homebrew 5.0.10
bijay.kc@bijays-MacBook-Pro Desktop % brew install scons python3 boost hdf5 protobuf google-perftools libpng
==> Fetching downloads for: scons, python@3.14, boost, hdf5, protobuf, gperftools and libpng
✔ Bottle Manifest scons (4.10.1)                     Downloaded    4.6KB/  4.6KB
✔ Bottle Manifest protobuf (33.3)                    Downloaded   10.5KB/ 10.5KB
✔ Bottle Manifest libpng (1.6.53)                    Downloaded    8.5KB/  8.5KB
✔ Bottle Manifest python@3.14 (3.14.2)               Downloaded   29.5KB/ 29.5KB
✔ Bottle Manifest boost (1.90.0)                     Downloaded   28.8KB/ 28.8KB
✔ Bottle Manifest hdf5 (1.14.6)                      Downloaded   24.2KB/ 24.2KB
✔ Bottle Manifest gperftools (2.17.2)                Downloaded   11.6KB/ 11.6KB
✔ Bottle Manifest mpdecimal (4.0.1)                  Downloaded   11.9KB/ 11.9KB
✔ Bottle Manifest ca-certificates (2025-12-02)       Downloaded    2.0KB/  2.0KB
✔ Bottle Manifest openssl@3 (3.6.0)                  Downloaded   11.8KB/ 11.8KB
✔ Bottle Manifest python@3.14 (3.14.2)               Downloaded   29.5KB/ 29.5KB
✔ Bottle Manifest readline (8.3.3)                   Downloaded   10.0KB/ 10.0KB
✔ Bottle Manifest sqlite (3.51.2)                    Downloaded   11.4KB/ 11.4KB
✔ Bottle Manifest xz (5.8.2)                         Downloaded   11.8KB/ 11.8KB
✔ Bottle mpdecimal (4.0.1)                           Downloaded  185.7KB/185.7KB
✔ Bottle ca-certificates (2025-12-02)                Downloaded  131.8KB/131.8KB
✔ Bottle Manifest lz4 (1.10.0)                       Downloaded   13.8KB/ 13.8KB
✔ Bottle Manifest zstd (1.5.7)                       Downloaded   16.2KB/ 16.2KB
✔ Bottle lz4 (1.10.0)                                Downloaded  276.1KB/276.1KB
✔ Bottle Manifest icu4c@78 (78.2)                    Downloaded    9.7KB/  9.7KB
✔ Bottle readline (8.3.3)                            Downloaded  758.1KB/758.1KB
✔ Bottle zstd (1.5.7)                                Downloaded  788.5KB/788.5KB
✔ Bottle Manifest gmp (6.3.0)                        Downloaded   13.3KB/ 13.3KB
✔ Bottle Manifest isl (0.27)                         Downloaded   14.5KB/ 14.5KB
✔ Bottle Manifest mpfr (4.2.2)                       Downloaded   12.7KB/ 12.7KB
✔ Bottle xz (5.8.2)                                  Downloaded  764.9KB/764.9KB
✔ Bottle Manifest libmpc (1.3.1)                     Downloaded   16.8KB/ 16.8KB
✔ Bottle Manifest libaec (1.1.4)                     Downloaded    9.2KB/  9.2KB
✔ Bottle Manifest gcc (15.2.0)                       Downloaded   23.4KB/ 23.4KB
✔ Bottle gmp (6.3.0)                                 Downloaded    1.0MB/  1.0MB
✔ Bottle libmpc (1.3.1)                              Downloaded  147.5KB/147.5KB
✔ Bottle libaec (1.1.4)                              Downloaded   56.6KB/ 56.6KB
✔ Bottle Manifest pkgconf (2.5.1)                    Downloaded   12.2KB/ 12.2KB
✔ Bottle sqlite (3.51.2)                             Downloaded    2.4MB/  2.4MB
✔ Bottle mpfr (4.2.2)                                Downloaded    1.1MB/  1.1MB
✔ Bottle pkgconf (2.5.1)                             Downloaded  121.9KB/121.9KB
✔ Bottle Manifest abseil (20260107.0)                Downloaded   63.0KB/ 63.0KB
✔ Bottle libpng (1.6.53)                             Downloaded  459.4KB/459.4KB
✔ Bottle scons (4.10.1)                              Downloaded    3.1MB/  3.1MB
✔ Bottle hdf5 (1.14.6)                               Downloaded    8.8MB/  8.8MB
✔ Bottle isl (0.27)                                  Downloaded    1.7MB/  1.7MB
✔ Bottle gperftools (2.17.2)                         Downloaded    1.5MB/  1.5MB
✔ Bottle abseil (20260107.0)                         Downloaded    1.9MB/  1.9MB
✔ Bottle protobuf (33.3)                             Downloaded    3.6MB/  3.6MB
✔ Bottle openssl@3 (3.6.0)                           Downloaded   10.9MB/ 10.9MB
✔ Bottle python@3.14 (3.14.2)                        Downloaded   18.8MB/ 18.8MB
✔ Bottle icu4c@78 (78.2)                             Downloaded   32.0MB/ 32.0MB
✔ Bottle gcc (15.2.0)                                Downloaded  154.1MB/154.1MB
✔ Bottle boost (1.90.0)                              Downloaded   62.3MB/ 62.3MB
==> Installing dependencies for scons: mpdecimal, ca-certificates, openssl@3, readline, sqlite, xz, lz4, zstd and python@3.14
```

- Pre-commit Warning: When I started the build, I got a warning about `pre-commit` being missing. During the setup on macOS, I encountered several warnings that required research. First, the build system notified me that `pre-commit` hooks were missing. I chose to bypass this by entering 'y', as it is a non-essential tool for basic simulation.
- Second, warnings regarding **KVM** and **PNG** support appeared. I determined that these were due to architectural differences between macOS and Linux; since gem5 defaults to software-based simulation for X86, these did not hinder my progress.
- Mac-Specific Warnings: My terminal mentioned that `linux/kvm.h` was missing. I researched this and realized it's because KVM is a Linux-only feature and doesn't apply to my Mac setup.
- Observation: I was surprised that even a simple 'Hello World' took over 500 million ticks. This shows how much work a CPU does for even a basic task. The separation of Python for config and C++ for the engine is a very smart design choice.

Additional observation:

One interesting detail I noticed in the terminal was that the 'Global frequency' was set to one trillion ticks per second. I did some research and found out that this means gem5 is simulating time at a picosecond level. It's impressive that the simulator is precise enough to track events happening at one-trillionth of a second, which explains why the 'Hello World' run took over 500 million ticks to finish.

Conclusion:

This assignment was a successful introduction to the world of computer architecture simulation. By setting up a GitHub workflow and building gem5 on macOS, I've established a solid foundation for more complex hardware experiments. Between fixing the terminal errors and seeing how the simulator tracks time down to the picosecond, I have a much better understanding that how complicated a CPU really is.

References

Homebrew. *Homebrew install script* [Computer software]. GitHub. Retrieved January 14, 2026, from https://github.com/Homebrew/install

The gem5 Project. *Building gem5* [Official Documentation]. Retrieved January 15, 2026, from https://www.gem5.org/documentation/general_docs/building

The gem5 Project. *Learning gem5: Part 1 - Creating a simple configuration script* [Tutorial]. Retrieved January 15, 2026, from https://www.gem5.org/documentation/learning_gem5/part1/simple_config/

GitHub. *The gem5 Simulator System* [Source code]. GitHub. Retrieved January 15, 2026, from https://github.com/gem5/gem5