What is gem5?

gem5 is a modular discrete event driven computer system simulator platform. That means that:

1. gem5's components can be rearranged, parameterized, extended or replaced easily to suit your needs.
2. It simulates the passing of time as a series of discrete events.
3. Its intended use is to simulate one or more computer systems in various ways.
4. It's more than just a simulator; it's a simulator platform that lets you use as many of its premade components as you want to build up your own simulation system.

gem5 is written primarily in C++ and python and most components are provided under a BSD style license. It can simulate a complete system with devices and an operating system in full system mode (FS mode), or user space only programs where system services are provided directly by the simulator in syscall emulation mode (SE mode). There are varying levels of support for executing Alpha, ARM, MIPS, Power, SPARC, RISC-V, and 64 bit x86 binaries on CPU models including two simple single CPI models, an out of order model, and an in order pipelined model. A memory system can be flexibly built out of caches and crossbars or the Ruby simulator which provides even more flexible memory system modeling.

There are many components and features not mentioned here, but from just this partial list it should be obvious that gem5 is a sophisticated and capable simulation platform. Even with all gem5 can do today, active development continues through the support of individuals and some companies, and new features are added and existing features improved on a regular basis.

Building gem5

Supported operating systems and environments

gem5 has been designed with a Linux environment in mind. We test regularly on **Ubuntu 18.04** and **Ubuntu 20.04** to ensure gem5 functions well in these environments. Though **any Linux based OS should function if the correct dependencies are installed**. We ensure that gem5 is compilable with both gcc and clang (see Dependencies below for compiler version information).

Mac OS should work when compiling using the clang compiler, with all other dependencies installed. However, at present, we do not officially test our builds on Mac OS. **We therefore cannot guarantee the same stability for those wishing to compile and run gem5 in Mac OS as we can in Linux-based systems**. In later versions of gem5, we hope to more effectively support Mac OS through improved testing.

As of gem5 20, **building and running gem5 in both Python 2 and Python 3 are supported**. When testing gem5 we primarily test using Ubuntu 18.04 with Python 2, and Ubuntu 20.04 with Python 3. We would therefore advice, if possible, for users to **utilize Python 2 in Ubuntu 18.04 and Python 3 in Ubuntu 20.04**. Though we shall eventually drop support for Python 2, (due to its retirement), we do not intend to do so until we receive ample evidence our community has migrated to Python 3.

If running gem5 in a suitable OS/environment is not possible, we have provided pre-prepared Docker images which may be used to compile and run gem5. Please see our Docker section below for more information on this.

Dependencies

- **git** : gem5 uses git for version control.
- **gcc**: gcc is used to compiled gem5. **Version >=5 must be used**. We support up to gcc Version 10.
- **Clang**: Clang can also be used. At present, we support Clang 3.9 to Clang 9 (inclusive).
- **SCons** : gem5 uses SCons as its build environment. SCons 3.0 or greater must be used.
- **Python** : gem5 relies on Python development libraries. As of gem5 version 20, gem5 can be compiled and run in environments using either Python 2.7 or Python 3
- **protobuf 2.1+** (Optional): The protobuf library is used for trace generation and playback.
- **Boost** (Optional): The Boost library is a set of general purpose C++ libraries. It is a necessary dependency if you wish to use the SystemC implementation.

Setup on Ubuntu 18.04

If compiling gem5 on Ubuntu 18.04, or related Linux distributions, you may install all these dependencies using APT:

```
sudo apt install build-essential git m4 scons zlib1g zlib1g-dev \
    libprotobuf-dev protobuf-compiler libprotoc-dev libgoogle-perftools-dev \
    python-dev python-six python libboost-all-dev pkg-config
```
This will create an environment which uses Python 2.

Setup on Ubuntu 20.04

If compiling gem5 on Ubuntu 20.04, or related Linux distributions, you may install all these dependencies using API:

```
sudo apt install build-essential git m4 scons zlib1g zlib1g-dev \
    libprotobuf-dev protobuf-compiler libprotoc-dev libgoogle-perftools-dev \
    python3-dev python3-six python-is-python3 libboost-all-dev pkg-config
```
This will create an environment which uses Python 3.

Getting the code

```
git clone https://gem5.googlesource.com/public/gem5
```

Building with SCons

gem5's build system is based on SCons, an open source build system implemented in Python. You can find more information about scons at http://www.scons.org. The main scons file is called SConstruct and is found in the root of the source tree. Additional scons files are named SConscript and are found throughout the tree, usually near the files they're associated with.

Within the root of the gem5 directory, gem5 can be built with SCons using:

```
scons build/{ISA}/gem5.{variant} -j {cpus}
```

For example:
```
scons build/X86/gem5.opt -j 4
```

run the hello world program:

```
build/X86/gem5.opt configs/example/se.py --cmd=tests/test-
progs/hello/bin/x86/linux/hello
```