[GEM5 Configuration]

GEM5 (https://www.gem5.org/) is a modular platform that enables the flexible simulation of different computer architectures. As this simulator will be used over the course of the semester to simulate and test different prospective computer architectures for *free* (vs. designing and fabricating a single computer architecture for approximately $3B-$4B, where B = billion), the objective of this exercise is to install the GEM5 simulation environment so you can use it in future homework assignments.

Follow the instructions below to install GEM5 on your linux system

(a) Run the commands below to create your own GEM5 workspace and to clone the GEM5 repository into it.

```
$ mkdir gem5_workspace && cd gem5_workspace
$ git clone https://gem5.googlesource.com/public/gem5
```

(b) Build GEM5 using SCons, an open-source build system implemented in Python. However, because SCons is not available on the CS@VT compute resources, you need to install it in your user space, as shown below.

```
$ mkdir gem5env
```

Create a virtual Python environment

```
$ python3 -m venv ./gem5env
$ source gem5env/bin/activate
```

Install SCons

```
(gem5env) $ pip install scons
```

Now that scons is installed, build GEM5 for the POWER instruction set architecture (ISA). This step should take approximately 10 minutes.

```
(gem5env) $ cd gem5
(gem5env) $ scons build/POWER/gem5.opt -j 8
```

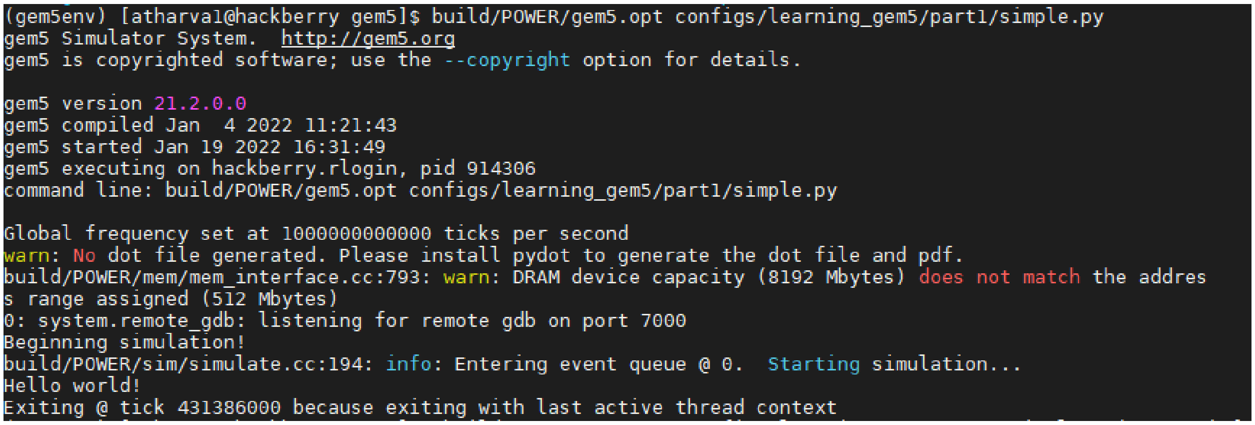
Copy the executable `hello32` from `gem5/tests/test-progs/hello/bin/power/linux` and name it `hello` in the same directory. Why? The configuration file that runs the "hello world" program expects the executable name to be `hello` (see line 91 of simple.py)

```
(gem5env) [atharva1@hackberry gem5]$ cp tests/test-
progs/hello/bin/power/linux/hello32 tests/test-progs/hello/bin/power/linux/hello
```

Run the configuration script `simple.py,` as shown below.

```
(gem5env) $ build/POWER/gem5.opt configs/learning_gem5/part1/simple.py
```

After running the aforementioned configuration script `simple.py`, the output of the command prompt should look similar to the screenshot below



```
(gem5env) [atharva1@hackberry gem5]$ build/POWER/gem5.opt configs/learning_gem5/part1/simple.py
gem5 Simulator System.  http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 21.2.0.0
gem5 compiled Jan  4 2022 11:21:43
gem5 started Jan 19 2022 16:31:49
gem5 executing on hackberry.rlogin, pid 914306
command line: build/POWER/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.
build/POWER/mem/mem_interface.cc:793: warn: DRAM device capacity (8192 Mbytes) does not match the addres
s range assigned (512 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7000
Beginning simulation!
build/POWER/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 431386000 because exiting with last active thread context
```

If you are able to see the output shown in the image above, you can then proceed with your simulations.
Note: To run workloads in POWER ISA for GEM5, you will need to compile your code with a cross-compiler for PowerPC

[Cross-compilation for POWER]

Install the cross compiler using the command below

```
sudo apt install gcc-powerpc64-linux-gnu
```

On the Ubuntu terminal, compile your code using the command below:

```
powerpc64-linux-gnu-gcc -O0 -ggdb3 -std=c99 -static <your_code>.c
-o <exe_name>.out
```