Assignment -2  
  
Vijaya Krishna Sameeraj Jonnavithula   
  
Date : September 22, 2024

1. gem5, a widely-used open-source computer system simulator, offers a highly customizable platform for simulating diverse system architectures. It is extensively utilized in computer architecture research and education, enabling users to model and assess the performance of various system components such as CPU, memory systems, and complete system simulations.

**Purpose of gem5 Simulator:**

* Architectural Research: gem5 is specifically designed to simulate both high-level and detailed architectural models, aiding researchers in comprehending the performance implications of different hardware design choices.
* System-Level Simulation: The simulator can replicate complete system workloads, including operating systems, which proves valuable for examining system-wide performance characteristics and optimizations.
* ISA Simulation: gem5 allows users to investigate the behavior and performance of processors that implement various Instruction Set Architectures (ISAs).
* Customizability: One of gem5's main advantages is its modularity, which enables users to easily experiment with different designs, simulate various cache hierarchies, and alter or add new components.

**gem5 and x86 ISA:**

Most contemporary desktop and server processors are powered by the ISA known as x86, which is one of the most popular. Because gem5 supports x86, users can use the architecture to simulate both user-level programs and entire system environments.  
  
**Main characteristics of gem5's x86 support:**

* Full-System Simulation: Gem5 can run whole operating systems (like Linux) in a virtualized environment because it supports full-system simulation for the x86 ISA. A thorough analysis of the system's performance is now possible.
* User-Level Simulation: Gem5 can simulate x86 user-level programs in addition to full-system simulation. This allows users to concentrate on the behavior of individual apps without having to engage with an entire operating system.
* Timing Models: Gem5 gives users the option to choose from a variety of CPU models with differing precision and efficiency, including:  
  atomic
* SimpleCPU: A straightforward CPU model primarily used for testing, with an emphasis on quick simulations.  
  When  
  SimpleCPU: An increasingly realistic model that replicates the latency and timing of an instruction's execution.
* O3CPU: An out-of-order CPU model that is perfect for in-depth architectural research, offering incredibly detailed performance simulations.

**Applications of gem5 with x86:**

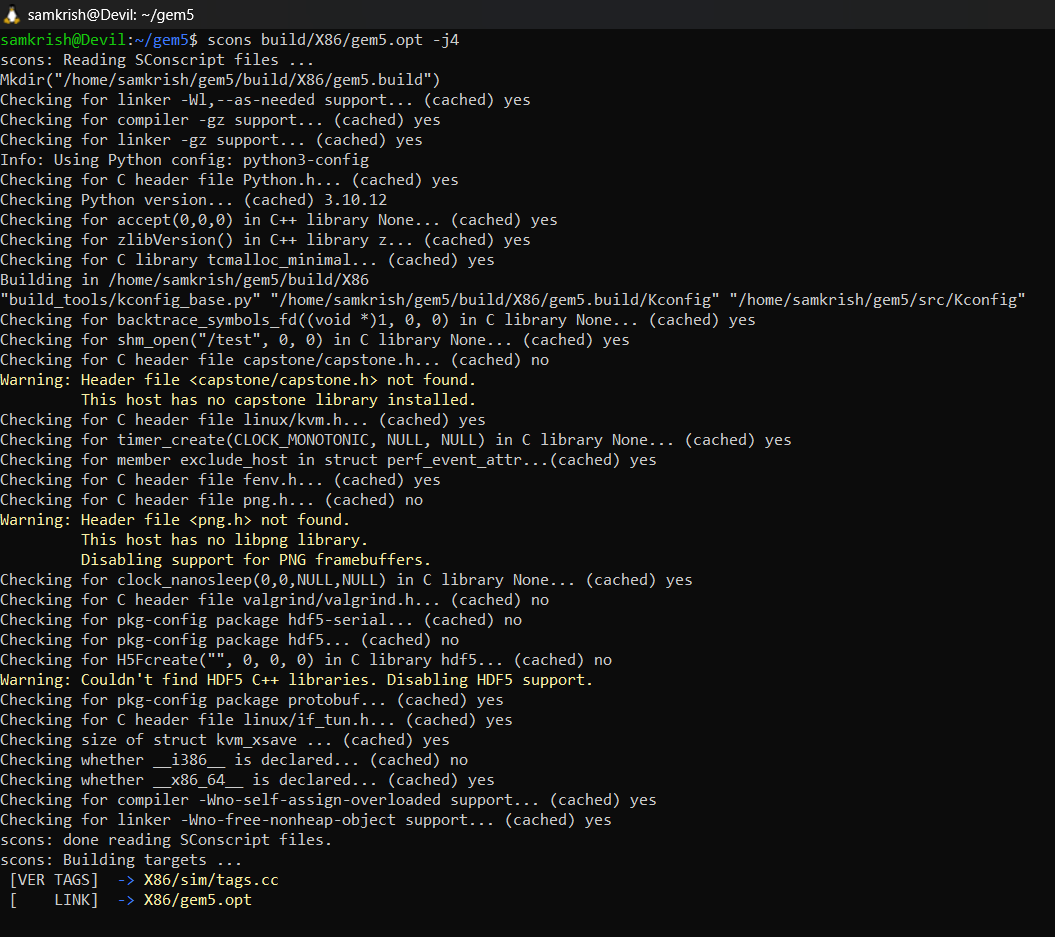
* Performance Analysis: Researchers can compare in-order and out-of-order execution, assess the performance of various x86 microarchitectures, and optimize for particular workloads using gem5.
* Cache and Memory Studies: Cache hierarchy studies and tuning are made possible by gem5's x86 support, which aids in the comprehension of the effects of cache sizes, latencies, and coherence protocols.
* OS and System Software Research: Gem5's complete x86 system support is useful for researching device drivers, system software optimizations, and operating system behavior.

1. Login and Update dependencies and clone gem 5 (gem 5 already cloned on the system):   
     
   A screen shot of a computer

   Description automatically generated  
     
   A screen shot of a computer program

   Description automatically generated
2. Building gem5 for x86:

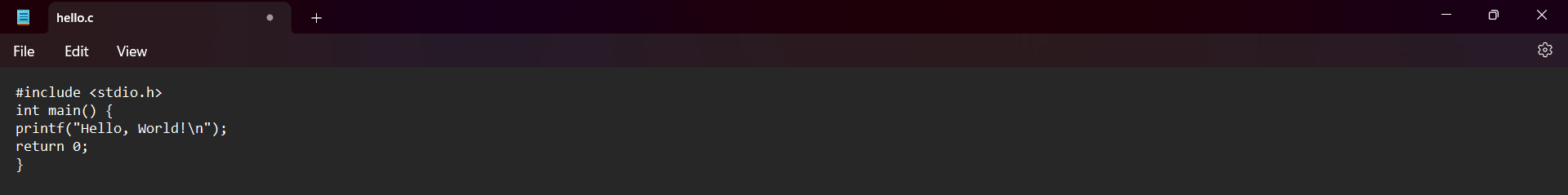
Code : scons build/X86/gem5.opt -j4



A screen shot of a computer

Description automatically generated

1. Hello World Program:



1. Compile :

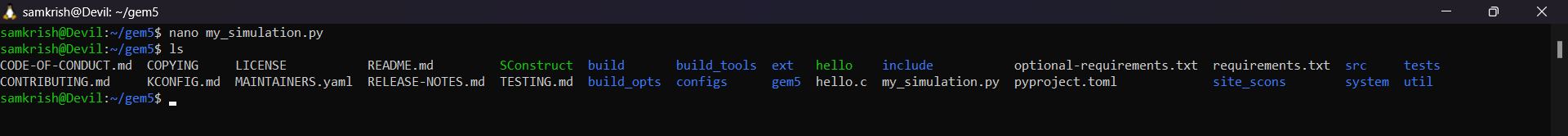
A screenshot of a computer

Description automatically generated

1. Python script to run :

A computer screen with a black background

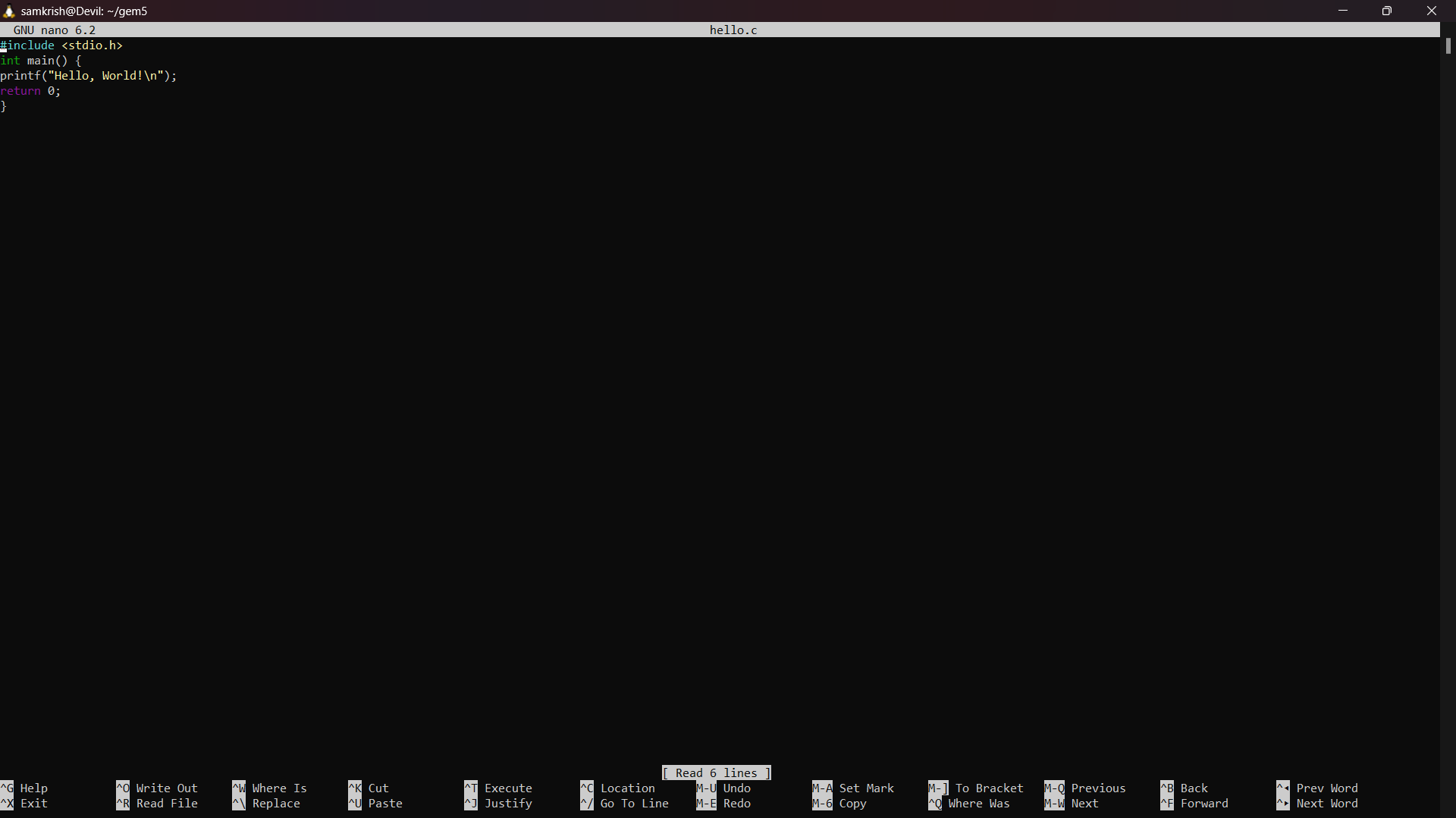
Description automatically generated



1. Output:  
     
   First we start by checking the build and allocating all the execute permissions and then we compile the c file and execute it to get the output.   
     
      
   A screenshot of a computer

   Description automatically generated
2. Troubleshooting :

* After writing the hello.c file , could not find it anywhere in the folder, faced the problem of finding it in the gem5 folder.   
    
  To troubleshoot this, I moved it manually, but it did not work,even after checking hidden files, could not find it, so had to create a new file using nano so that we create a file in the required folder and directory.   
  A black screen with many colorful text

  Description automatically generated with medium confidence
* 
* After moving, I has issues with the permissions, so added the +x for the files.
* Also faced issue with memory allocation, handled it with the following commands :   
    
  sudo fallocate -l 4G /swapfile # Create a 4 GB swap file
* sudo chmod 600 /swapfile # Set the correct permissions
* sudo mkswap /swapfile # Set up the swap file
* sudo swapon /swapfile # Enable the swap file