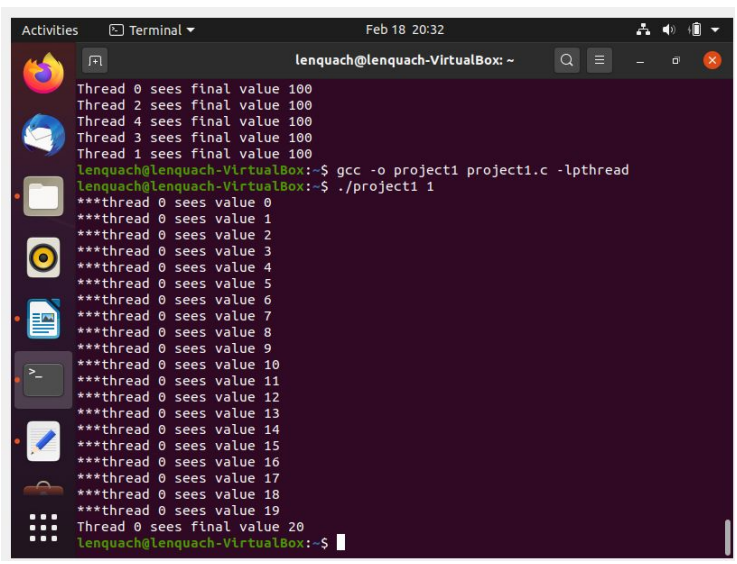
Project 1 Report: Multithreaded Programming and Synchronization
CECS 326 - Operating Systems

Group members: Jorel Caoile
Len Quach

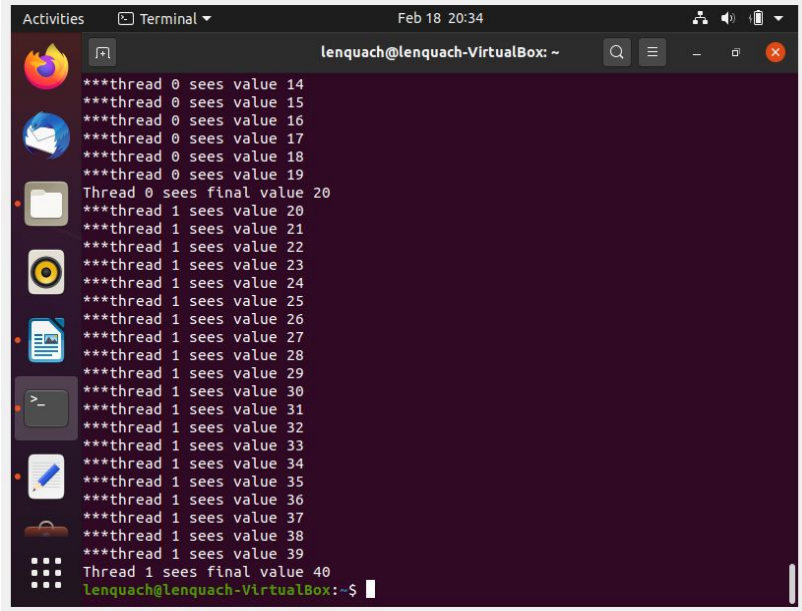
Abstract
The project focuses on process management while using the pthreads library in Linux. This part of Linux allows multiple concurrent activities within the kernel. This project has two parts: creating multiple threads without synchronization and creating multiple threads with synchronization. There is a problem when performing multithreaded programming without synchronization. The final result, once the program has finished executing, has it so that the threads will not have the same final values. Synchronization solves that problem.

1. Part 1: Simple Multi-Thread Programming without Synchronization

Create 1 process
Fig.1

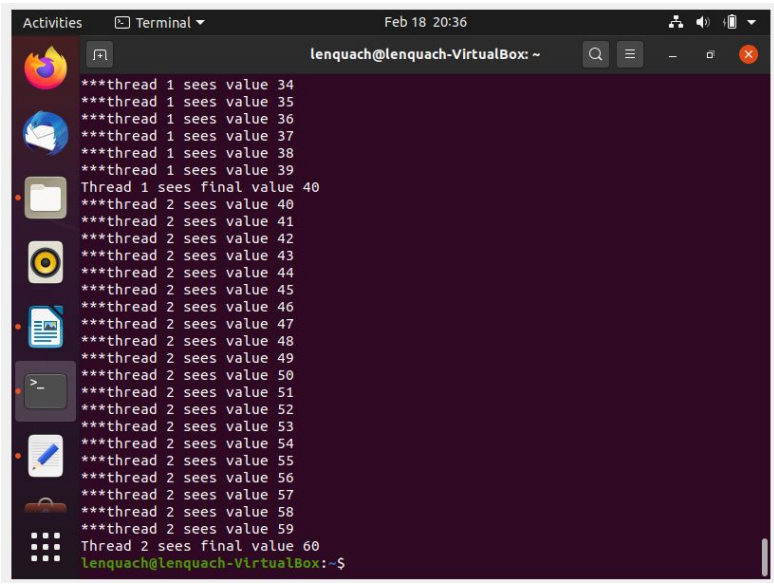


Create 2 processes
Fig. 2



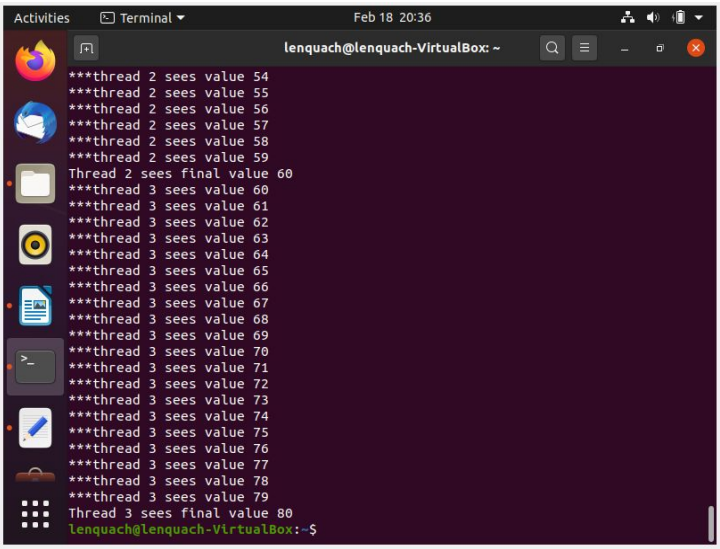
Create 3 processes

Fig.3

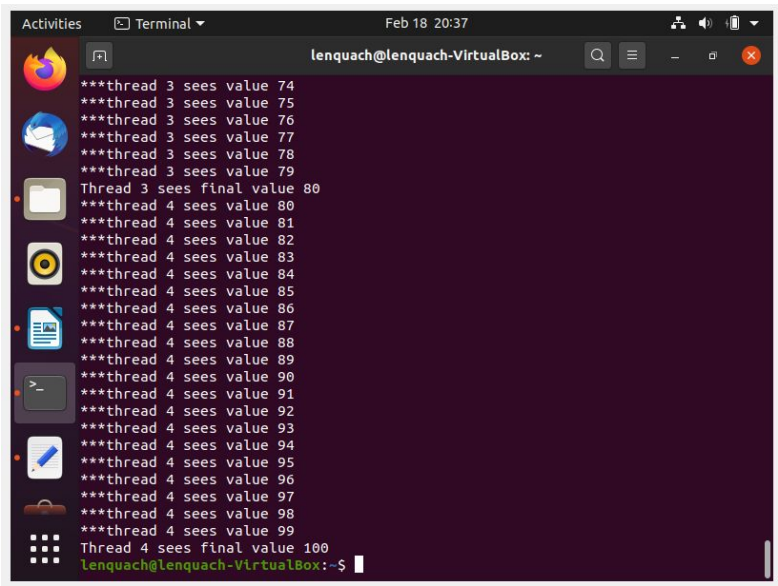


Create 4 processes

Fig. 4



Create 5 processes
Fig.5



The program in part 1 runs without Synchronization, and therefore, the number of threads does not depend on the other one or the threads do not see the same final value. The last thread will see a final value 20 times the number of threads.

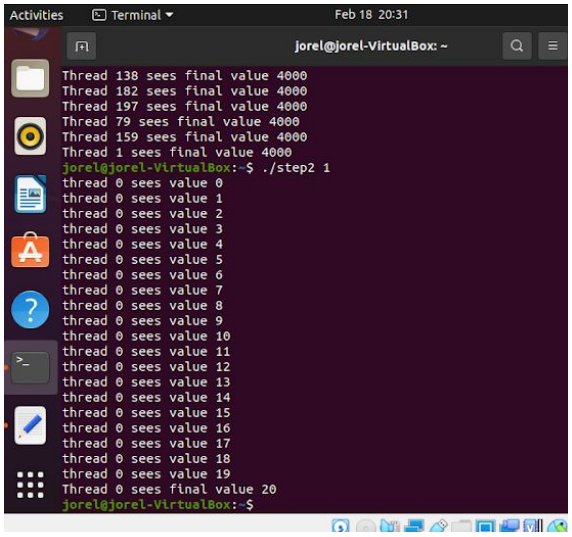
2. Part 2: Simple Multi-Thread Programming with Proper Synchronization

The mutex lock prevents multiple threads from executing simultaneously. This ensures synchronized access of shared resources in the code. The mutex barrier causes a certain number of threads to be at the barrier before starting the next part of the program. Using both,

the program runs in a more organized way and helps prevent errors in the shared variable. Without synchronization, there can be a loss of data and race conditions may occur. However, with the synchronization, the threads will all see a value of 20 times the number of threads.

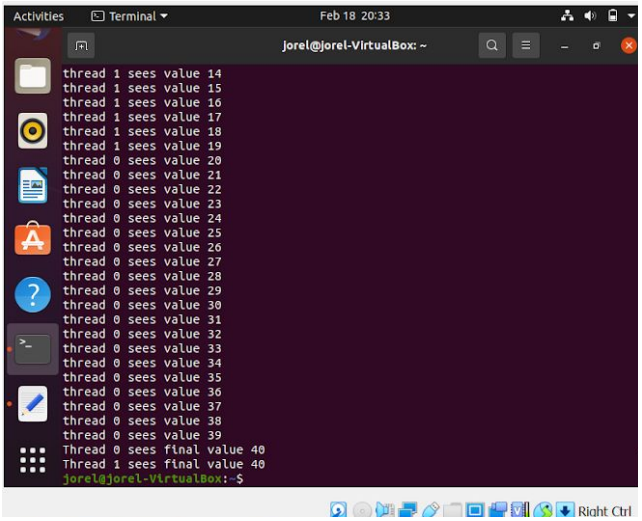
1 thread

Fig.1



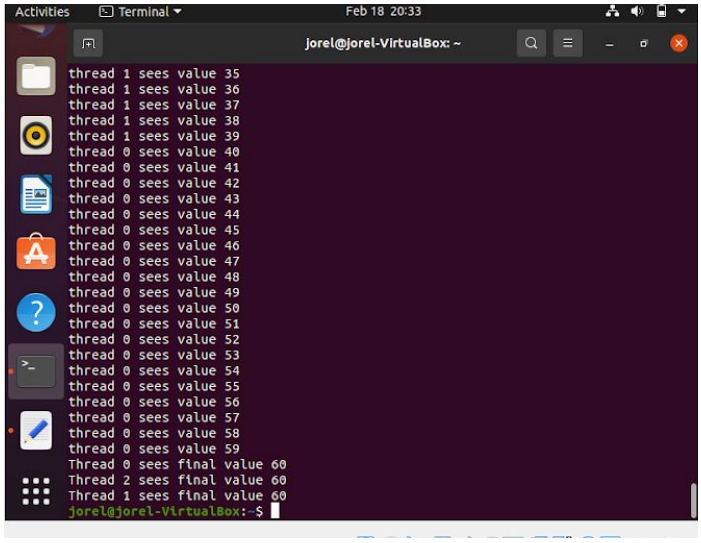
2 threads

Fig. 2

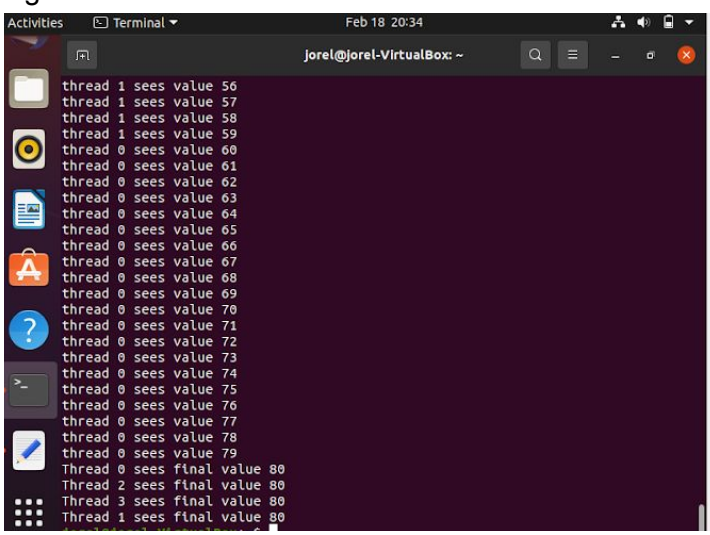


3 threads

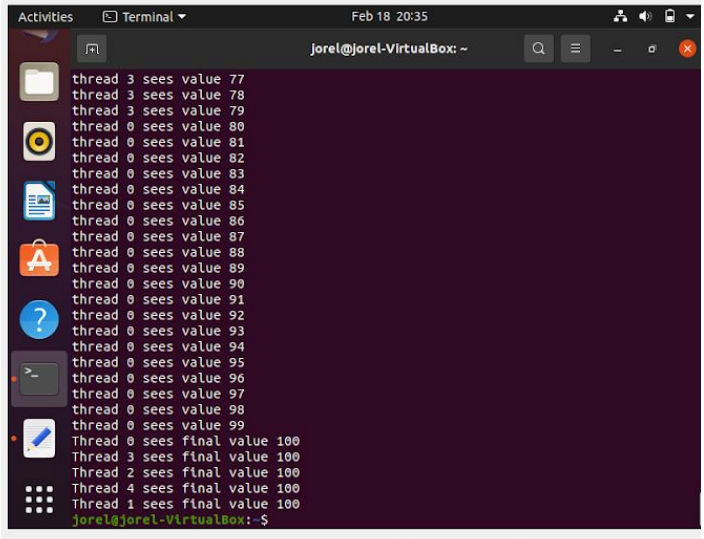
Fig.3



4 threads
Fig. 4



5 threads
Fig.5



3. Individual Contributions

Jorel Caoile:

- Write code for part 2 and write the report
- Do ReadMe file

Len Quach:

- Write code for part 1 and write the report
- Record video

Video Link: https://youtu.be/kppKK4N6QHk