CE 705 Real time systems
Practical Assignment 1

Objectives:

- Review some LINUX commands.
- Be familiar with the C compiler under LINUX and make refreshment in C language.
- Understand thread synchronization in standard LINUX.

Tasks:

- Start your session in LINUX, then enter the following commands at the prompt, and remember the functionality of each one:
    a. $ ls
    b. $ls –a
    c. $ls -l
    d. $man ls
    e. $ hostname
    f. $ cd ..
    g. $ pwd
    h. $ exit

- Study "Employee code" in C and use the gcc compiler to compile and run (The code is attached to this file). Write your observation.

- Study "Thread code" in C and use the gcc compiler to compile and run (The code is attached to this file). Write your observation.

- Study "Semaphore Code" in C and use the gcc compiler to compile and run (The code is attached to this file). Write your observation.

- Write a C program to create two threads, both sharing a global variable count, initialized to zero.

  Thread A: Increments count variable and displays the following message at every second for 30 times.

  "The value of count is NNNN in thread A"

  Thread B: Increments the same count variable and displays the following message at every second for 30 times.

  "The value of count is NNNN in thread B"

  Compile and run the program. Execute the program and observe the display sequence of count variable.

  Variation: Modify your program to protect the "incrementing and displaying of variable operation" in such a manner that after completion of thread A, the thread B execution continues.
  Hint: Use POSIX mutex synchronization method, or Semaphores.

  Now observe the display sequence of count variable.

  Write your observation.