Assignment 7

Problem (A):

Write two C programs named *program1.c* and *program2.c* which will be responsible for reading the contents of a file (single file). Now you use the concept of semaphore to provide synchronization between the processes of program1 and program2, so that program2 can read the contents of the specified file only after reading the contents of the file by program1. Again if the program1 runs for n times, then program2 can read the contents of the file at most n times only, not more than that. Moreover you have to remove the semaphore that is created for your above mentioned operations properly by using proper system call. Try to demonstrate properly.

Hints:

* For creating a semaphore set and accessing an existing semaphore set, you need a system call - **semget(key_t key, int nsems, int oflag)**

* For performing operations on one or more of the semaphores in the set of semaphores you need a system call - **semop(int semid, struct sembuf *opsptr, size_t nops)**

* For performing various control operations like setting the value of a semaphore in the semaphore array, you need a system call -**semctl(int semid, int semnum, int cmd, arguments)**

  The *arguments* parameter is optional., *cmd* may be macro IPC_RMID for removing semaphore set specified by the *semid* and macro SETVAL for setting the value of a semaphore.

* For better understanding about the system call, go through the man pages.

* For better understanding about the system call API, go through the man pages and

  UNIX NETWORK PROGRAMMING by W.RICHARD STEVENS.

* Check with command ipcs

Problem (B) ~~(Optional)~~:

Write two C programs named *prog1.c* and *prog2.c* which will be responsible for reading some value (i.e. say a string “Hello” or any other value of your choice) in a shared memory block. First allow the respective processes of prog1 and prog2 to read and print the content controlled by semaphore. Next do the necessary modification in the code such that both the processes reflect the *Busy-Wait lock* (i.e. semaphore based Spin lock). To implement this program use two semaphores variables (say A and B) and implement busy-wait scenario as we discussed in theory). Finally you have to remove the semaphore that is created for your above mentioned operations properly by using proper system call. Try to demonstrate properly.