1. What is Deadlock? List necessary conditions for deadlock. Method to recover from deadlock.
2. How logical address to physical address mapped. Explain in detail.
3. Numerical on finding the physical address given the table with segment number, base, length. And find the address for given segment no. and logical address.
4. Question on paging and how it is mapped from logical address to physical address.
5. What “myprogram.o” file contains. It is generated after compiling using “gcc myprogram.c -o myprogram.o”.
6. Question on synchronization and mutex given the pseudo code.
7. Find the number of child processes given the pseudo code with fork.
   ```
   For i=0; i<10; i++:
           If i %2 == 0:
             fork();
   ```
8. Numerical on finding the best algorithm between FCFS-fit and Best-fit given the size of five memory blocks and size of four processes.
   Given, Block Size = {100, 500, 200, 300, 600}
   Process Size = {212, 417, 112, 426}
9. Is there any fragmentation possible depending on the fixed size of the block? When size of block is not fixed, is there possibility of no fragmentation? Give explanation with reason.
10. Consider a system running ten I/O-bounds tasks and one CPU-bound task. Assume that the I/O bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 millisecond and all processes are long-running tasks. What is the CPU utilization for a round-robin scheduler when:
    a. The time quantum is 1 millisecond
    b. The time quantum is 10 millisecond

    Answer:

    (a) The time quantum is 1 millisecond: Irrespective of which process is scheduled, the scheduler incurs a 0.1 millisecond context-switching cost for every context-switch. This results in a CPU utilization of 1/1.1 * 100 = 91%.

    (b) The time quantum is 10 milliseconds: The I/O-bound tasks incur a context switch after using up only 1 millisecond of the time quantum. The time required to cycle through all the processes is therefore 10*1.1 + 10.1 (as each I/O-bound task executes for 1 millisecond and then incur the context switch task, whereas the CPU-bound task executes for 10 milliseconds before incurring a context switch). The CPU utilization is therefore 20/21.1 * 100 = 94%.