School of Computing Science and Engineering

Programme: B.Tech Branch: CSE

Course: CSE212-Operating Systems Lab 2012-13 Winter Semester

Common to all Batch's and Slots

Cycle Sheet -2

1. Implement the producer-consumer problem / reader-writer problem using semaphore

   Example scenario:
   Assume there are three processes: Pa, Pb, and Pc. Only Pa can output the letter A, Pb B, and Pc C. Utilizing only semaphores (and no other variables) the processes are synchronized so that the output satisfies the following conditions:
   i. A B must be output before any C's can be output.
   ii. B's and C's must alternate in the output string, that is, after the first B is output, another B cannot be output until a C is output.
   Similarly, once a C is output, another C cannot be output until a B is output.
   iii. The total number of B's and C's which have been output at any given point in the output string cannot exceed the number of A's which have been output up to that point.

   Examples
   AACB -- invalid, violates i.
   ABACAC -- invalid, violates ii.
   AABCABC -- invalid, violates iii.
   AABCAAABC – valid i.
   AAAABCBC – valid ii.
   AB – valid iii.

2. Implement a solution for the classical synchronization problem

   Example scenario:
   In the Cigarette-Smokers Problem, Consider a system with three smoker processes and one agent process. Each smoker continuously rolls a cigarette and then smokes it. But to roll and smoke a cigarette, the smoker needs three ingredients: tobacco,

paper and matches. One of the smoker processes has paper, another has tobacco and the third has matches. The agent has an infinite supply of all three materials. The agent places two of the ingredients on the table. The smoker who has the remaining ingredient then makes and smokes a cigarette, signaling the agent on completion. The agent then puts out another two of the three ingredients and the cycle repeats. Write a program to synchronize the agent and the smokers.

3. Simulate Banker's algorithm for deadlock prevention

4. Simulate Banker's algorithm for deadlock avoidance

5. Write a program to solve the memory fragmentation problems for
   i. Fixed Memory Allocation (Internal)
   ii. Variable Memory Allocation (External)

6. Implement a program to allocate memory by applying the following strategies.
   i. FIRST FIT
   ii. BEST FIT
   iii. WORST FIT

   Example scenario:
   Implement a memory management schemes:
   Free space is maintained as a linked list of nodes with each node having the starting byte address and the ending byte address of a free block. Each memory request consists of the process-id and the amount of storage space required in bytes. Allocated memory space is again maintained as a linked list of nodes with each node having the process-id, starting byte address and the ending byte address of the allocated space. When a process finishes (taken as input) the appropriate node from the allocated list should be deleted and this free disk space should be added to the free space list. [Care should be taken to merge contiguous free blocks into one single block. This results in deleting more than one node from the free space list and changing the start and end address in the appropriate node]. For allocation use first fit, worst fit and best fit.

7. Implement a program for page replacement using the following algorithms
   i. FIFO
   ii. LRU
   iii. OPTIMAL

8. Simulate with a program to schedule disk in seek optimization.
9. Implement any file allocation technique (Linked, Indexed or Contiguous)