Computer Operating Systems

Problem #1 Describe the need for synchronization in OS design. Define semaphores and two operations that are available on each semaphore. Define the bounded buffer (producer-consumer) problem. Given the following code for consumer:

```
while (true){
   // produce item for the buffer
   wait(full);
   wait(mutex)
   // consumer item from the buffer
   signal(mutex)
   signal(empty);
}
```

Reconstruct the data structures (semaphores and their initialization) and the code for producer. Explain the operation of your code.

Problem #2 Explain the concept of *page fault* and *victim page* in virtual memory organization. Explain what page replacement policy is and what it tries to optimize. Describe FIFO, Optimal (OPT) and Lest Recently Used (LRU) policies.

Assume that main memory contains 4 frames (numbered 0 through 3) and the paging system contains 10 pages (numbered A through J). Assume that the frames are initially empty. For each of the above page replacement policies: FIFO, OPT, LRU, give an example of a page reference string (a list of page references) that incurs exactly 10 page faults. Explain your answer. Note that filling an empty frame counts as a single page fault.

Problem #3 Define block, sector, track and cylinder. Explain why application programs usually access the disk through the operating system rather than directly. Motivate why it is necessary for the operating system to keep track of free (unallocated) disk blocks. Describe two allocation methods: free block list and bit vector. Compare their relative advantages and disadvantages.