CS 414	System Programming and Operating Systems	Homework 2
Summer 2001		Due June 28, 2001

Note: The homework is due before the class on the due date.

Answer the following questions:

1. Implement busy-waiting semaphores using critical regions. Define the data structure for semaphore and provide region code for semaphore initialization, *wait*, and *signal*.

2. In a regular paging system, there is one page table associated with each process. If an inverted page table is used, then there is only one page table for the entire system. Each entry in the page table occupies certain bytes. Which of the following will likely need more bytes: an entry in the regular page table or an entry in the inverted page table? Explain why.

3. As defined on page 308 to page 309 of the textbook, a *stack* (page-replacement) algorithm is one with the following property: the set of pages in memory for n frames is always a *subset* of the set of pages that would be in memory with $n + 1$ frames.

   For the *second-chance* (clock), LFU (least frequently used), and MFU (most frequently used) algorithms, answer the following questions:

   (a) Which one(s) of the algorithms is (are) stack algorithm(s)? Explain why.

   (b) Which one(s) of the algorithms could exhibit *Belady's anomaly*? Explain why.

4. To some extent, virtual memory tries to achieve what dynamic loading and overlay do.

   A virtual memory system uses demand paging, which loads in a page whenever requested. This is similar to dynamic loading.

   By using an appropriate page-replacement algorithm, a virtual memory system tries to evict pages that will unlikely be used and use the space for pages that are (will likely be) needed. This is similar to overlay.

   Explain the advantages and disadvantages of using virtual memory instead of dynamic loading and overlay in terms of complexity, the need for OS support or programmer support, and impact on performance.

5. Consider contiguous file allocation.

(a) Contiguous file allocation is superior to linked allocation in terms of file access performance for both sequential access and direct access. Explain why.

(b) Do all these benefits apply to contiguous allocation in memory management? Explain why.

(c) What are the shortcomings of contiguous file allocation? Propose one solution (or a variation) to address at least one of the shortcomings and explain the tradeoff you have made.

(d) (**Optional bonus question**) Describe an application domain in which contiguous file allocation becomes a desirable choice. Explain your answer. (Hint: think of a case where the shortingcomings of contiguous allocation either disappear or are less significant compared to advantages.)

6. Consider a virtual memory system. Each memory access without page fault takes 200 *nsec* (10^{-9} seconds). In case of page faults, 20 *msec* (10^{-3} seconds) is needed for page-fault handling.

(a) Calculate the average memory access performance if the page fault rate is 10^{-6} and 10^{-3}, respectively.

(b) Now suppose LRU is used for page replacement. No hardware support has been provided. For each memory access, LRU requires 20 μsec (10^{-6} seconds) in processing on top of 200 *nsec* for memory access. Assume the page fault rate is 10^{-6} and memory access with page fault takes 20 *msec* (we can ignore the cost of LRU in this case), calculate the average memory access delay.

(c) Now if LRU is being implemented by hardware and does not incur any cost for memory access, redo the calculation (assuming all other parameters stay the same as in (b)).

(d) Redo the calculation in (c) assuming that memory access with page fault now takes 40 *msec* due to the cost of copying the page being replaced onto the disk.

(e) What can you conclude from your calculations?