Homework 6 — due on Tuesday 21 November 2000

Total number of points available on this homework is 100. Full credit is equivalent to 100 points.

1. (10 pts) Suppose that a process spends a fraction p of its time waiting on I/O. With n processes in memory at once, the probability that all n processes are waiting for I/O at once is p^n.
   (a) (6 pts) What is the CPU utilization (the percent of time the CPU has work to do)?
   (b) (4 pts) If processes are idle waiting on the CPU 40% of the time, how many processes does it take to achieve a CPU utilization of 97%?
2. (10 pts) What is the difference between external and internal fragmentation?
3. (10 pts) What are the advantages and disadvantages of a large page size?
4. (15 pts) A computer provides each process with 65,536 bytes of address space divided into pages of 4096 bytes. The compiler divides a program into 3 segments: a text segment of size 32,768 bytes, a data segment of size 16,386 bytes, and a stack segment of size 15,870 bytes. Will this program fit in the address space? If the page size were 512 bytes, would it fit? Show your work and explain what problems could arise.
5. (15 pts) Suppose we have demand-paged virtual memory. The page table is in registers. It takes 7 milliseconds to service a page fault if an empty page is available, or the replaced page is not modified. Memory access time is 90 nanoseconds. Assume that replaced page is modified 75% of the time. What is the highest page-fault rate to get an effective memory access time of no more than 200 nanoseconds?
6. (20 pts) Consider a two dimensional array A(1:300,1:300) and the following two loop nests.

```
1. for (i = 1; i <= 200; i++)          2. for (j = 1; j <= 200; j++)
     for (j = 1; j <= 200; j++)             for (i = 1; i <= 200; i++)
       A (i, j) = 0                           A (i, j) = 0
```

   Assume that the variables i and j, and the constants are all in registers, so that the CPU never loads them from memory. The array A is stored in row-major order (i.e., consecutive elements of A are stored in memory as follows: A(1,1), A(1,2) ... A(1,300), A(2,1) ... A(300,300)), and exactly 300 elements of A fit on a single page. Note that the loop nests only access 200×200 elements. Assume that the OS uses an LRU replacement policy, and assigns 3 page frames to the process. Page 0 has the code, such that all instructions references are to page 0; and array A starts on page 1 (i.e., A(1,1) is the first element on page 1).
   (a) How many pages does it take to store A?
   (b) Summarize the page references for loop nest 1 and compute the total number of page faults.
   (c) Summarize the page references for loop nest 2 and compute the total number of page faults.
7. (20 pts) Consider the following page reference string:

   0,1,7,2,3,2,7,1,0,3

   How many page faults would occur for the following replacement algorithms, assuming 2, 3, and 4 page frames? Assume that all frames are initially empty.
   - Optimal
   - FIFO
   - LRU
   - Clock with one bit - assume the "clock" starts out pointing at the first frame.