

# CPSC 331 - Operating Systems

Instructor Marc Corliss

Homework 4

Due: 10/24/07 (beginning of class)

1. Explain the difference between logical and physical addresses. Which memory management techniques use logical addresses? Why is this separation beneficial?

2. Consider a logical address space of 32 pages of 1024 words each, mapped onto a physical memory of 8 frames.

(a) How many bits are there in the logical address?

(b) How many bits are there in the physical address?

3. Consider a paging system with the page table stored in memory.
  - (a) If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
  
  
  
  
  
  
  
  
  
  
  - (b) If we add a TLB, and 75 percent of all page-table references are found in the TLB, what is the effective memory reference time? (Assume that finding a page-table entry in the TLB take zero time, if the entry is there.)
  
  
  
  
  
  
  
  
  
  
4. When do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

5. Explain an inverted page table. What is its advantage over a conventional page table? Naive search of an inverted page table is inefficient. How can this search be made efficient?

6. Given the memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of the First-fit, Best-fit, and Worst-fit algorithms place the processes of P1 (212K), P2 (417K), P3 (112K), and P4 (426K) (in order)? Which algorithm makes the most efficient use of memory?

7. Assume a virtual address space of  $2^{32}$  bytes, a physical address space of  $2^{20}$  bytes, and a page size of 256 bytes.

(a) Sketch out each address, denoting the bits for the page and the offset parts.

(b) How many pages are in this system?

(c) How many frames are available?

(d) What is the size of the page table needed to support this system? Can it be kept in memory at one time?

(e) Suggest some alternative designs which may combat this problem.

8. Consider the following page reference string

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

How many page faults would occur for each of the following page replacement algorithms, assuming three frames? Show your work, i.e., the sequence of pages brought into each frame. You should assume that all frames are initially empty.

(a) LRU

(b) FIFO

(c) Optimal