CIS 520 – Quiz #2 – Fall 2010 Name: ____________________

Date: Mon., Nov. 15, 2010
Total: 100 pts

1. (25 points) Dynamic Memory Management: A dynamic memory allocator uses a linked list to track free blocks. In the allocator, all blocks are allocated in multiples of 4 bytes in length. Answer the following questions assuming that the free list contains just two blocks, of 28 and 16 bytes each, in that order. Ignore any space required by bookkeeping overhead.

   (a) (5 points) If a process calls malloc( ) with a request for 7 bytes, then the allocator returns a pointer to a block of length 8 because all blocks are allocated in full-word (4 byte) units. The last byte in the block is wasted. Is this an example of internal or external fragmentation? Explain briefly.

   (b) (5 points) If the process calls malloc( ) with a request for 24 bytes and the allocator is using the first-fit algorithm, the request for 24 bytes will be satisfied by breaking the block of size 28 into two parts, and returning the remaining 4 bytes to the free list. This small block of size 4 bytes is sometimes referred to as "sawdust" because it probably will never be used. Is this an example of internal or external fragmentation? Explain briefly.

   (c) (5 points) List a sequence of malloc( ) calls that would succeed using first-fit allocation, but fail with best-fit, or explain why such a sequence cannot exist.

   (d) (5 points) List a sequence of malloc( ) calls that would succeed with best-fit, but fail with first-fit, or explain why such a sequence cannot exist.

   (e) (5 points) List a sequence of malloc( ) calls that would succeed with worst-fit, but fail with best-fit, or explain why such a sequence cannot exist.

2. (25 points) Paging and Page Replacement Algorithms:

   (a) (5 points) Why are page sizes always a power of 2?

   (b) (10 points) Discuss situations in which the most recently used (MRU) page replacement algorithm generates fewer page faults than the least recently used (LRU) page replacement algorithm. Also, discuss under what circumstances the opposite holds.

   (c) (5 points) Discuss the philosophy behind the second chance (clock) algorithm and how it approximates LRU. Drawings are fine here to help with your explanation.

   (d) (5 points) Give a simple example to show how the LRU algorithm and the second chance (clock) algorithm might select different pages for eviction, or explain why they always select the same page.

3. (25 points) System Calls and Swapping (5 points each):

(a) Discuss the advantages of using copy-on-write (COW) when creating a new process by using the Unix fork( ) system call.

(b) Why don't we typically use LRU for selecting pages to swap, but typically use LRU for cache block replacement?

(c) Assume that we have a 2 GHz processor and a hard disk drive with a 10ms average seek time, approximately how many instructions could be executed while waiting for a disk access to complete?

(d) What is thrashing? What can be done to prevent thrashing from occurring?

(e) What is Belady's anomaly? Can the LRU algorithm exhibit Belady's anomaly?

4. (25 points) Virtual Address Mapping: Consider the page table shown below for a system with 32-bit virtual addresses and 28-bit physical addresses, and 4MByte pages; that is, page offsets are represented using 22 bits. Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. If the reference would cause a page fault to be generated, just write "page fault". If you cannot determine the corresponding physical address or if a page fault will occur, based on the page table entries given, just write "unable to determine". Note that "--" indicates that the page is currently not loaded into a frame.

(a) FFFFFFFF → ________________

(b) 3FFFFFFF → ________________

(c) 00006FFF → ________________

(d) 00005432 → ________________

Also, compute the virtual address corresponding to the given physical addresses.

(e) FFFFFFF → ________________

(f) 6EF0FFF → ________________

(g) 3FF0FAD → ________________

(h) 5800000 → ________________

Page	**Page Frame**
000	2C
001	3F
002	--
003	1B
..	..
0FF	2B
100	2D
..	..
3FB	22
3FC	26
3FD	27
3FE	--
3FF	3D