COSC 6360 **QUIZ #1** **JULY 13, 2009**

Name: _____ (First name first) **Score:** _____

Closed book. You can have with you one single-sided 8½ by 11 sheet of notes.

1. Why is it impossible to evaluate the memory requirements of a UNIX process ***at creation time***? When does it become possible? (2×10 pts)

At creation time, the new process is a clone of its parent. To evaluate the true memory requirements of the process, we need to wait until the child issues an exec() system call.

2. Consider an old BSD virtual memory system with a single-handed clock? Assuming that the clock scans the whole main memory once every two seconds, what are the minimum and maximum times that the page will stay in main memory after the last reference to it? (2×10 pts)

Answer: The page will remain in main memory at least _two_ seconds and at most _four_ seconds plus or minus a small fraction of a second.

Explanation: A page that was referenced for the last time just after it was marked invalid by the hand of the clock will remain in memory for the duration of almost two complete memory scans while a page that was referenced for the last time just before it was marked invalid will remain in memory for the duration of a single scan.

3. How does Mach ensure that all processes accessing a mapped file share the same view of the file? (20 pts)

The range of virtual memory addresses allocated to the mapped files will be shared by all processes accessing the file.

4. Consider a 64-bit system using a clustered page table. What would be the size of page table entry if the page table uses (a) ***partial subblocking*** with a subblocking factor of ***two*** and (b) ***complete subblocking*** with the ***same*** sublocking factor? (2×10 pts)

Answer: Page table entries will occupy ___24___ bytes if the page table uses partial subblocking and ____32___ bytes if it uses complete subblocking.

5. The ***Midway*** distributed shared memory system uses a request/release scheme similar to that of Munin but requires users to specify which shared variables are associated with each request() call. In a few words, explain how this additional requirement affects (a) the run-time performance of user programs and (b) their portability. (2×10 pts) (*Hint: If needed, you can continue your answer on the back of the sheet.)*

a. With Munin, a workstation must performing a request() must wait until it has received the most recent values of all shared variables. With, Midway, the workstation will only request the most recent values of the shared variables associated with the request() call. As a result, Midway will be faster than Munin.

b. Porting a program to Midway is a much harder task because the programmer porting the program will have to find out first which shared variables are accessed within each request()/release() pair. This is a daunting task for any large complex program.