CS 423, Spring 2010
Midterm (total = 25 points)

Name: _____

NetID: _____

This is an open-book/notes take-home midterm. Please work on this midterm independently. Please print this sheet and answer on the hard copy in the space provided. The exam is due Wednesday, March 3rd by 10am Central time (in class).

Q1: If your operating system supports only *blocking* I/O calls, which type of threads would you use to implement a Web server (circle one)? **(1 point)**

User-level threads Kernel-level threads

Why (one short sentence, please)? **(1 point)**

Answer: _____

Q2: A test-and-set instruction, `test_and_set(x)`, returns the current value of Boolean variable `x` and sets `x` to true in one atomic action. Implement mutex lock() and unlock() operations using the test-and-set instruction. Please do not exceed the space provided below.

```
lock (x) {                          /* (1 point) */


}

Unlock (x) {                        /* (1 point) */


}
```

Q3: A 32-bit machine has a main memory of 1 GB and a page size of 2KB. If 4 bytes were needed for each entry in the page table, how long is the (single-level) page table? **(2 points)**

Q4: In Q3, how long would the page table be if an inverted page table was used? **(2 points)**

Q5: Consider a system with five processes P1, …, P5, and four resources R1, …, R4 protected by respective mutexes. All mutexes are initially unlocked. The following sequence of resource requests then occurs: (i) P5 requests R4, (ii) P1 requests R1, (iii) P4 requests R1, (iv) P3 requests R3, (v) P5 requests R3, (vi) P1 requests R2, (vii) P3 requests R2, (viii) P1 requests R4. You can assume that each request is either granted or blocked (according to regular mutex rules) before the next request is made. No locks are released in the above sequence. Draw the wait-for graph. **(2 points)**

Does a deadlock exist (circle one answer)? **(1 point)**

YES NO

Which processes are deadlocked? (Note that, the answer could be *all*, *none*, or a specific subset in between) **(1 point)**

Answer: _____

Q6: In a 64-bit architecture which type of page table would you recommend? (Circle one) **(1 point)**

Single level page table Multi-level page table Inverted page table

Q7: In the linux kernel, which scheduling policy does not support time slices? **(1 point)**

Answer: _____

Q8: In the linux kernel two processes are scheduled by SCHED_NORMAL. One is very interactive and the other is very compute intensive. After a while, their dynamic priorities drift apart. Approximately how many levels apart can their dynamic priorities drift, at most? (Please round to the nearest 5) **(1 point)**

Answer: _____

Q9: Consider the four mutex locks L1, L2, L3, and L4.Two processes A and B request these four locks. Each process requests them in some (possibly different) pre-programmed sequence. There are 24 possible sequences for requesting the locks making for 576 programming possibilities. How many of these can *possibly* result in a deadlock? Do not make assumptions on when the locks are released. A process may or may not release some locks before acquiring others.
(2 points)

Q10: A system has 6 tasks and 10 locks. Under the priority ceiling protocol, how many lower-priority tasks can a single higher-priority task wait for, in the worst case? **(1 point)**

Answer: _____

Q11: In some system, let P1, P2 and P3 be three processes where P1 is the highest priority process and P3 is the lowest. P3 arrives first and is executed immediately. It requests mutex L1. Before it ends, P2 arrives and requests mutex L2. Before P2 ends, P1 arrives and requests L1 followed by L2. Assume that a process does not relinquish its locks until it finishes execution. Use the timeline below to depict the execution sequence of the three processes if they run the priority inheritance protocol. **(2 points)**

P1 _____

P2 _____

P3 _____

Q12: Answer Q11 if the priority ceiling protocol is used instead. **(2 points)**

P1 _____

P2 _____

P3 _____

Q13: Two threads adopt the software implementation of mutexes, mentioned below:

```
while (lock == 1) {};
lock = 1;
/* perform critical section */
lock = 0;
```

Which of the following properties are *guaranteed* by the implementation? (Circle all that apply) **(1 point)**

Mutual exclusion Bounded wait Progress

Q14: In a virtual memory system the time it takes to access the TLB is y, whereas the time it takes to access physical memory is 10y. The TLB hit ratio is 60%. If a two level page table is used, what is the average virtual memory access time? **(2 points)**

_____ Good Luck _____