CIS 415 * Spring 2010 * Quiz 2

1. **Timer and I/O Interrupts** (10 pts)
Consider a multicore or multiprocessor computer with 4 cores (processors) that share memory, I/O devices, and OS code. Assume processes are eligible to run on any of the four cores at all times.

a. For the timer interrupt system, how many timer clocks are needed for best performance? Circle your answer and explain in one sentence or less.

   One clock  Four clocks  Doesn’t matter how many clocks

   Any answer accepted. Bad question – too hardware dependent. Multicore systems default to one clock that interrupts core 0 which on some systems can then cause interrupts on the other cores as needed by decrementing a counter for each CPU to enforce timeslicing on the other CPUs.

b. For the I/O interrupt system, on which CPU(s) should the interrupt occur? Circle your answer and explain in one sentence or less.

   On one specific CPU  On any single CPU  On all four CPUs

   Any CPU can handle the interrupt and move the blocked process to the ready queue.

2. **Processes and Threads** (10 pts)
For each statement below, circle P if the statement is true for processes and circle T if it is true for threads. Some statements may be true for both; some may not be true for either.

a. PC keeps track of next instruction to be executed   both
b. Has high context switch overhead   P only
c. Can be CPU-bound or I/O bound   both
d. Can only be scheduled by the OS kernel   P only
e. Its ID is only known by the kernel   neither
f. Can fork a child   P or both
g. Has a unique CPU state associated with it
h. Useful for embarrassingly parallel programs
i. Also known as lightweight process
j. Costs 25 cents at Walmart

3. **fork and exec** (5 pts)
What code is running after this code is executed? What can you eliminate and still have the same effect? Circle the answer and explain.

```c
main() {
    pid_t pid1, pid2;
    pid1 = exec("/bin/mycode");
    pid2 = fork();
    if (pid2 == 0) exec("/bin/childcode");
    else exec("/bin/parentcode");
}
```

/bin/mycode  /bin/childcode  /bin/parentcode  none

Remove the last three instructions. They will never be executed.