

Computer Science 15-410: Operating Systems

Mid-Term Exam, Fall 2003

1. Please read the entire exam before starting to write. This should help you avoid getting bogged down on one problem.
2. Be sure to put your name and Andrew ID below *and also* put your Andrew ID at the top of *each* following page.
3. This is a closed-book in-class exam. You may not use any reference materials during the exam.
4. You must complete the exam by the end of the class period.
5. Answer all questions. The weight of each question is indicated on the exam. Weights of question *parts* are estimates which may be revised during the grading process and are for your guidance only.
6. Please be concise in your answers. You will receive partial credit for partially correct answers, but truly extraneous remarks may count against your grade.
7. Please don't ask us questions of the form "If I answered like this, would it be ok?" or "Are you looking for ...?"

Andrew Username	
Full Name	

Question	Max	Points	Grader
1.	10		
2.	10		
3.	10		
4.	20		

100

1. 10 points Give a *brief* definition of each of the following terms as they apply to this course. You may add a second sentence providing an example or a clarification.

(a) 2 points Instruction

(b) 2 points Heap

(c)

2 points

(d)

(e)

2 points

 Round-robin

2. 10 points What will happen if the following code is executed by a process in user mode?

```
void bar(void)
{
    int sum = 0, ip;

    ip = &sum;
    while (1) {
        sum += *ip;
        --ip;
    }
}
```

If you feel that multiple scenarios are possible, please pick one and focus on its details. You may wish to break your answer into several parts, such as “At first, ...”, “After some time...” and/or “Eventually...”.

3.

10 points

 ???

4. 20 points Mutex implementation

Imagine you are hired by a small, aggressive start-up soon after you finish 15-410. Your company manufactures arcade-style (coin-operated) video games which run your proprietary operating system kernel, library code, and applications. Your first task is to tune the implementation of mutexes in your company's thread library. In particular, your boss asks you to investigate the issue of whether or not spin-waiting is appropriate.

- (a) 10 points Identify *two* issues which influence the decision of whether to spin-wait in `mutex_lock()` or not, and how long. You should assume that your company's operating system runs on multiple models of game console, and that games may be moved from one model to another without recompilation. Also, your company is investigating the possibility of "MMP" (massively multi-player) games.
- (b) 5 points Identify one or more new system calls which your thread library will call to gather information about whether and how long to spin-wait.
- (c) 5 points Sketch pseudo-code for `mutex_lock()` and show how the results obtained from those system calls are used.