15-410, Fall 2007, Homework Assignment 1.
Due Monday, October 8, 20:59:59 p.m.

Please observe the non-standard submission time... As we intend to make solutions available on the web site immediately thereafter for exam-study purposes, please turn your solutions in on time.

Homework must be submitted in either PostScript or PDF format (not: Microsoft Word, Word Perfect, Apple Works, LaTeX, XyWriter, WordStar, etc.). Submit your answers by placing them in the appropriate hand-in directory, e.g., `/afs/cs.cmu.edu/academic/class/15410-f07-users/$USER/hw1/$USER.ps` or `/afs/cs.cmu.edu/academic/class/15410-f07-users/$USER/hw1/$USER.pdf`. A plain text file (.text or .txt) is also acceptable, though it must conform to Unix expectations, meaning lines of no more than 120 characters separated by newline characters (note that this is *not* the Windows convention or the MacOS convention). Please avoid creative filenames such as `hw1/my_15-410_homework.PdF`.

1 Interruption (10 pts.)

Consider the following assembly wrapper code for a timer handler used in Project 1.

```
.globl handle_timer

handle_timer:
    PUSHA
    CALL c_timer_handler
    POPA
    RET
```

1.1 2pts

What mistake was made in this code?

1.2 5pts

What incorrect execution pattern will result when the wrapper above is invoked by timer interrupts? Draw a small picture or two to indicate the pattern. You may wish to consider the case in which the main program is executing the following code.

```
    unsigned int big1 = 0xffffffff, big2 = 0xffffffff, big3 = 0xffffffff;

    handler_install();

    enable_interrupts();

    while (big1-- > 0)
        while (big2-- > 0)
            while (big3-- > 0)
                continue;

    printf("All done.\n"); /* Yep, *all* done. */

    reboot();
```

1.3 3pts

What will be the symptoms or externally observable effects of running this code? If something is displayed on the screen, what is it likely to be? It isn't necessary to research the exact Project 1 hardware configuration to be 100% sure of your answer; a sensible detailed answer will be accepted as long as it is plausible.

2 Threads (10 pts.)

Suppose you are writing code designed to be part of a "cooperative" (non-preemptive) user-space thread library expected to run on a uniprocessor. Assume that the API for the thread library is the "Thread Management API" found in Section 4.1 of the Project 2 handout.

2.1 7pts

Define a type `mutex_t` and write *the simplest implementations you can* for:

```
int mutex_init(mutext_t *mp)
int mutex_lock(mutex_t *mp)
int mutex_unlock(mutex_t *mp)
```

2.2 3pts

What problems, if any, will arise if this code is run on a multi-processor machine?

3 Plumbers (5 pts.)

Amy, Bob, and Casey are chefs in a single kitchen. They cook with spoons, pots, and burners. They each have special methods and dishes, and consequently they need different numbers of these items in the worst case. The maximum numbers they need are shown in the following table.

Chef	Spoons	Pots	Burners
Amy	5	2	2
Bob	4	6	2
Casey	7	1	1

3.1 2 pts

Is the following state safe? Why or why not?

Chef	Spoons	Pots	Burners
Amy	1	2	2
Bob	3	6	0
Casey	4	0	0
Available	3	0	1

3.2 2 pts

Is the following state safe? Why or why not?

Chef	Spoons	Pots	Burners
Amy	0	2	2
Bob	3	3	0
Casey	6	0	0
Available	1	1	1