15-410, Spring 2006, Homework Assignment 1.

Due Monday, October 9, 6:59:59 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-f06-users/$USER/hw1/$USER.ps` or `/afs/cs.cmu.edu/academic/class/15410-f06-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 Memory Madness (10 pts.)

Consider a system with 600 megabytes of temporary disk space.

Processes A and B share an execution pattern:

```
void A_or_B(void) {
  temp_allocate(100 * 1024 * 1024); /*1*/
  ...other code...
  temp_allocate(100 * 1024 * 1024); /*2*/
  ...other code...
  exit(code); /* invokes temp_free() as side effect */
}
```

Process C's execution pattern looks like this:

```
void C(void) {
  temp_allocate(100 * 1024 * 1024); /*1*/
  ...other code...
  temp_allocate(100 * 1024 * 1024); /*2*/
  ...other code...
  temp_allocate(100 * 1024 * 1024); /*3*/
  ...other code...
  temp_allocate(100 * 1024 * 1024); /*4*/
  ...other code...
  temp_allocate(100 * 1024 * 1024); /*5*/
  ...other code...
  exit(code); /* invokes temp_free() as side effect */
}
```

1.1 4 pts

Illustrate how this system can deadlock.

Use the table below and the execution-trace format presented in the lecture slides. You may not need to fill in all lines of the table, and you may use more lines than we provide if you wish.

Execution Trace

time	Process A	Process B	Process C
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

1.2 2 pts

Sketch a process/resource graph showing the deadlock.

1.3 4 pts

Is the following state safe? Why or why not?

Who	Max	Has	Room
A	200	100	100
B	200	200	0
C	500	300	200
System	600	0	-

2 Paradise Lost (5 pts.)

Slide 12 of the "Paradise Lost" lecture contains a BSD-style "'XXX comment"' indicating that something is wrong with the code. Briefly state the problem and provide code which fixes it.

3 IRET (10 pts.)

Slide 27 ("IA32 Single-Task Mode Example") of the "Hardware" lecture shows that a "surprise" (meaning a system call, hardware interrupt, or exception) which does not involve a stack switch pushes, in order, `%EFLAGS`, `%CS`, and `%EIP`. Meanwhile, slide 18 of the "Yield" lecture explains that in the case of a stack switch due to a privilege-level change, a "surprise" pushes, in order, `%SS`, `%ESP`, `%EFLAGS`, `%CS`, and `%EIP`. Consider for the purposes of this question a "surprise" which does not push an error code. One instruction, `IRET`, is used to return to the previous execution stream, whether or not a stack switch was required to handle the surprise. Briefly explain how IRET determines whether it needs to pop three or five register values from the stack.

We understand this question requires command of a level and quantity of detailed x86 hardware knowledge which we would not require in an exam setting.