15-410, Fall 2008, Homework Assignment 1.
Due Monday, October 6, 20:59:59 p.m.

You need complete only Question 3 and one other question.

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-f08-users/$USER/hw1/$USER.ps` or `/afs/cs.cmu.edu/academic/class/15410-f08-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 Nuts & Bolts (10 pts.)

Uh-oh...somebody's skeleton game kernel just got into trouble.

```
simics> c
Hello from a brand new kernel!
[cpu0] (@ cycle 15054660) Exception 13: General_Protection_Exception
Psuedo-exception 1035 caught!
[cpu0] cs:0x001021a8 p:0x001021a8  cmp byte ptr [eax],0x0
_doprnt (fmt=0x1041cf "foo says %d\n", args=0x106aac "", radix=0, putc=0x101f90,
    putc_arg=0x106958 "\230i\020") at /foo/p1/410kern/stdio/doprnt.c:208
208             while (*fmt != '\0') {
simics> bt
#0 0x1021a8 in _doprnt (fmt=0x1041cf "foo says %d\n", args=0x106aac "", radix=0, putc=0x101f90,
    putc_arg=0x106958 "\230i\020") at /foo/p1/410kern/stdio/doprnt.c:208
#1 0x10205c in vsnprintf (s=0x106998 "Hello from a brand new kernel!", size=255,
    fmt=0x1041cf "foo says %d\n", args=0x106aac "")
    at /foo/p1/410kern/stdio/sprintf.c:73
#2 0x101f7d in SIM_printf (fmt=0x1041cf "foo says %d\n")
    at /foo/p1/410kern/simics/simics_c.c:11
#3 0x100b9c in kernel_main (mbinfo=0x2a020, argc=1, argv=0x106ba4, envp=0x67eac)
    at /foo/p1/kern/game.c:64
#4 0x100ac2 in mb_entry (info=0x2a020, istack=0x106b00) at /foo/p1/410kern/entry.c:32
#5 0x100a0f in _start () in kernel
#6 0xfe30 in ?? ()
simics>
```

Here's the code:

```
int kernel_main(mbinfo_t *mbinfo, int argc, char **argv, char **envp)
{

    lmm_remove_free( &malloc_lmm, (void*)USER_MEM_START, -8 - USER_MEM_START );
    lmm_remove_free( &malloc_lmm, (void*)0, 0x100000 );

    handler_install(tick);
    interrupt_setup();

    lprintf( "Hello from a brand new kernel!" );

    extern int foo(void);
    lprintf("foo says %d\n", foo());

    while (1) {
        continue;
    }

    return -1;
}
```

Here is a register dump (excerpted from "pregs -all", in the same oddball order which Simics uses).

Register	Value
EAX	0x001041cf
ECX	0x00106b3c
EDX	0x001069b5
EBX	0x0002a020
ESP	0x001067e8
EBP	0x00106930
ESI	0x0002a204
EDI	0x0002a205
EIP	0x001021a8
EFLAGS	0x00200006
ES	0x0018
CS	0x0010
SS	0x0010
DS	0x0000
FS	0x0018
GS	0x0018

What went wrong?

2 Dining Philosophers (10 pts.)

Consider the following modification of the standard dining-philosophers scenario. Some philosophers are "right-handed" and run the following code:

```
while (1) {
  acquire_chopstick(right); /* "may" block */
  acquire_chopstick(left);  /* "may" block */

  eat();

  release_chopstick(right);
  release_chopstick(left);

  think();
}
```

However, other philosophers are "left-handed" and run this code instead:

```
while (1) {
  acquire_chopstick(left);  /* "may" block */
  acquire_chopstick(right); /* "may" block */

  eat();

  release_chopstick(left);
  release_chopstick(right);

  think();
}
```

Explain why, at any table with a mixture of left-handed and right-handed philosophers, deadlock cannot occur.

3 Zero Terror in the Bakery (20 pts.)

Consider the Bakery Algorithm as presented in class. As you know from 15-213, "an `int` is not an integer." One unfortunate feature of real-world `int`s is that they overflow. In order to stave this off as long as possible, assume that the code presented in class has been adjusted so that the `number[]` array is declared as `unsigned int`.

Using the tabular format presented in class, show how overflow wreaks havoc with the Bakery Algorithm. You may use more or fewer lines or columns.

Execution Trace

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