15-410, Fall 2016, Homework Assignment 1.

Due Monday, October 17, 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, XyWrite, WordStar, etc.). Submit your answers by placing them in the appropriate hand-in directory, e.g., `/afs/cs.cmu.edu/academic/class/15410-f16-users/$USER/hw1/$USER.ps` or `/afs/cs.cmu.edu/academic/class/15410-f16-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 Chefs (5 pts.)

Jamie, Kelly, and Morgan are chefs in a single kitchen. They cook with knives, bowls, and woks. The maximum number of each item needed by each chef is shown in the following table.

Chef	Knives	Bowls	Woks
Jamie	2	2	3
Kelly	3	2	3
Morgan	2	2	3

Imagine that the kitchen contains 5 knives, 6 bowls, and 6 woks.
Consider the following system state.

Chef	Knives	Bowls	Woks
Jamie	1	2	2
Kelly	1	0	1
Morgan	1	1	2
Available	2	3	1

As you can see, various chefs are still entitled to request various numbers of various tools.

1.1 2 pts

Is the state shown above safe, unsafe, or deadlocked? Explain.

1.2 3 pts

It is the case for at least one chef that, for some tool type, a request for *one* more of that tool, if granted, would leave the system in a safe state, but a request for *two* more of that tool, if granted, would leave the system in an unsafe state. State the two requests that the chef is entitled to make and justify your claim that one of the requests, if granted, would leave the system in a safe state but the other request, if granted, would leave the system in an unsafe state.

(Continued on next page)

2 "Carpe Diem" (5 pts.)

Consider the following critical-section protocol. (This protocol is presented in "standard form," i.e., if thread 1 is running this code, $i == 1$ and $j == 0$.)

```
       volatile int want[2] = {0, 0};
       volatile int turn = 0;

 1.    do {
 2.       ...remainder section...
 3.       want[i] = 1;
 4.       while (turn != i) {
 5.         while (want[j])
 6.           continue;
 7.         turn = i; // carpe diem
 8.       }
 9.       ...begin critical section...
10.       ...end critical section...
11.       want[i] = 0;
12.    } while (1);
```

There is a problem with this critical-section protocol. Identify a required property which this protocol does not have and then present a trace which supports your claim. You may use more or fewer lines in your trace.

Execution Trace

time	Thread 0	Thread 1
0		
1		
2		