15-410, Spring 2011, Homework Assignment 1.

Due Tuesday, February 22, 20:59:59

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-s11-users/$USER/hw1/$USER.ps` or `/afs/cs.cmu.edu/academic/class/15410-s11-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 Tape drives (4 pts.)

Consider a system with three processes and five tape drives. The maximal needs of each process are declared below:

Resource Declarations

| Process A | Process B | Process C |
|---|---|---|
| 3 tape drives | 2 tape drives | 4 tape drives |

Imagine the system is in the state depicted below. List one request which the system should grant right away, and one request which the system should react to by blocking the process making the request. Briefly justify each of your answers.

| Who | Max | Has | Room |
|---|---|---|---|
| A | 3 | 1 | 2 |
| B | 2 | 1 | 1 |
| C | 4 | 2 | 2 |
| System | 5 | 1 | - |

**(Continued on next page)**

# 2 Race conditions (6 pts.)

Consider the two threads whose body functions are shown below (imagine each is invoked via `thr_create()`):

```
int n = 1;

void *thread0(void *ignored) {
  while (n >= 0)
    --n;
  return (0);
}

void *thread1(void *ignored) {
  while (n < 1)
    ++n;
  return (0);
}
```

## 2.1 3 pts

Show an execution trace in which thread 1's loop *body* executes exactly three times.

## 2.2 3 pts

Show an execution trace in which both thread 0 and thread 1 run forever.

When showing an execution trace, use the tabular trace format found in the lecture slides. Please be sure that your trace is clear and conclusive (otherwise your answer will not receive full credit). You may use more or fewer lines or columns than are provided in this sample table. You may change the column headings if you wish.

Execution Trace

| time | Thread 0 | Thread 1 |
|---|---|---|
| 0 | | |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |