CSC 139 Operating System Principles
Homework 2

Fall 2019

Posted on Nov. 2, due on Nov. 12 (11:59 pm). Write your own answers. Late submission will be penalized (turn in whatever you have).

Exercise 1. (OSC 6.22) (15%) Consider the code example for allocating and releasing processes shown below:

```
#define MAX_PROCESSES 255
int number_of_processes = 0;

/* the implementation of fork() calls this function */
int allocate_process() {
    int new_pid;

    if (number_of_processes == MAX_PROCESSES)
        return -1;
    else {
        /* allocate necessary process resources */
        ++number_of_processes;

        return new_pid;
    }
}

/* the implementation of exit() calls this function */
void release_process() {
    /* release process resources */
    --number_of_processes;
}
```

1. Identify the race condition(s).
2. Assume you have a mutex lock named `mutex` with the operations `acquire()` and `release()`. Indicate where the locking needs to be placed to prevent the race condition(s).

Exercise 2. (OSC 8.28) (20%) Consider the following snapshot of a system:

Answer the following questions using the banker's algorithm:

	Allocation				**Max**			
	A	B	C	D	A	B	C	D
T_0	3	1	4	1	6	4	7	3
T_1	2	1	0	2	4	2	3	2
T_2	2	4	1	3	2	5	3	3
T_3	4	1	1	0	6	3	3	2
T_4	2	2	2	1	5	6	7	5

Available			
A	B	C	D
2	2	2	4

1. Illustrate that the system is in a safe state by demonstrating an order in which the threads may complete.

2. If a request from thread T_4 arrives for (2,2,2,4), can the request be granted immediately?

3. If a request from thread T_2 arrives for (0,1,1,0), can the request be granted immediately?

4. If a request from thread T_3 arrives for (2,2,1,2), can the request be granted immediately?

Exercise 3. (OSC 8.22) (5%) Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Is this system deadlock-free? Why or why not?

Exercise 4. (5%) Can a system be in a state that is neither deadlocked nor safe? If yes, give an example system.

Exercise 5. (OSC 5.23) (5%) Consider a system implementing multilevel queue scheduling. What strategy can a computer user employ to maximize the amount of CPU time allocated to the user's process?

Please complete the following survey questions:

1. How much time did you spend on this homework?

2. Rate the overall difficulty of this homework on a scale of 1 to 5 with 5 being the most difficult.

3. Provide your comments on this homework (e.g., amount of work, difficulty, relevance to the lectures, form of questions, etc.)