CSC 139 Operating System Principles
Homework 2

Spring 2020

Posted on April 15, due on April 26 (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:

```c
#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. (5%) Two processes, A and B, each need three records, 1, 2, and 3, in a database. If A asks for them in the order 1, 2, 3, and B asks for them in the same order, deadlock is not possible. However, if B asks for them in the order 3, 2, 1, then deadlock is possible. With three resources, there are 3! or six possible combinations in which each process can request them. What fraction of all the combinations is guaranteed to be deadlock-free?

Exercise 6. (5%) A machine has 48-bit virtual addresses, 32-bit physical addresses, and the page size is 8 KB. How many entries are needed for one process's page table?

Exercise 7. (5%) A computer with a 32-bit address space uses a two-level page table. Virtual addresses are split into a 9-bit top-level page table field (the directory), an 11-bit second-level page table field, and an offset. How large are the pages and how many are there in the address space?

Exercise 8. (OSC 10.1) (10%) Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

Exercise 9. (OSC 9.6) (15%) Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

Exercise 10. (OSC 10.9) (15%) Consider the following page reference string: 7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1. Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?

- LRU replacement
- FIFO replacement
- Optimal replacement

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.)