# CSCI-UA.0202-002: Operating Systems
# Midterm Exam (October 13th, 2022)

**Important Notes- READ BEFORE SOLVING THE EXAM**

- **If you perceive any ambiguity in any of the questions, state your assumptions clearly and solve the problem based on your assumptions. We will grade both your solutions and your assumptions.**
- **This exam is take-home.**
- **The exam is posted on Brightspace, at the beginning of the Oct 13th lecture (11am EST).**
- **You have up to 23 hours and 55 minutes to submit on Brightspace, in the same way as you submit an assignment.**
- **Your answers must be very focused. You may be penalized for wrong answers and for putting irrelevant information in your answers.**
- **Your answer sheet must be organized as follows:**
  - **The very first page of your answer must contain only:**
    - **You Last Name**
    - **Your First Name**
    - **Your NetID**
    - **Copy and paste the honor code showed in the rectangle at the bottom of this page.**
  - **In your answer sheet, answer one problem per page. The exam has four main problems, each one must be answered in a separate page.**
- **This exam consists of 4 problems, with a total of 100 points.**
- **Your answers can be typed or written by hand (but with clear handwriting). It is up to you. But you must upload one pdf file containing all your answers.**

**Honor code (copy and paste to the first page of your exam)**

- You may use the textbook, slides, and any notes you have. But you may not use the internet.
- You may NOT use communication tools to collaborate with other humans. This includes but is not limited to Google-Chat, Messenger, E-mail, etc.
- Do not try to search for answers on the internet it will show in your answer, and you will earn an immediate grade of 0.
- Anyone found sharing answers or communicating with another student during the exam period will earn an immediate grade of 0.
- **"I understand the ground rules and agree to abide by them. I will not share answers or assist another student during this exam, nor will I seek assistance from another student or attempt to view their answers."**

**Problem 1**

a. [4 points] State two advantages of static linking (in 1-2 sentences each).

b. [4 points] State two disadvantages of static linking (in 1-2 sentences each).

c. [4 points] State two advantages of dynamic linking (in 1-2 sentences each).

d. [4 points] State two disadvantages of dynamic linking (in 1-2 sentences each).

e. [4 points] When you start a process, by clicking an icon or typing a command for example, is there any possibility that a *linking* operation may take place? Explain briefly.

f. [5 points] Suppose the source code of a program has the following two lines (the rest of the code does not contains any fork() or any other execve() and does not contain bugs):

```
…
execve(progname1);
execve(progname2);
…
printf(“End of program\n”);
exit(0);
```

How many times the statement “End of program” will be printed? Justify in 1-2 sentences.

**Problem 2**

The following C program forks several processes and prints some numbers on the screen.
**a.** The table below has a list of possible outputs as well as some outputs that cannot be generated by that program. Write “yes” next to the output that can be generated and “no” otherwise. For the ones that cannot be generated (i.e. you wrote a “no” next to it) give in 1-2 sentences one reason why this output cannot be generated by the program. (e.g. a reason can be: “x cannot be printed before y” or “y cannot be printed more than once”, etc.)

```
main(){
        if( fork() == 0){
          if( fork() == 0)
                printf(“3”);
          else{
                wait();
                printf(“4”);
          }
        }
        else{
          if( fork() == 0 ){
                printf(“1”);
                exit(0);
          }
          printf(“2”);

        }
        printf(“0”);
        return 0;
}
```

[20 points, 1 point each entry]

| **Output** | **Can be printed (Yes/No)** | **Reason (1-2 sentences)**<br>**Put a reason only for the entries that you said cannot be printed.** |
|---|---|---|
| 2030401 | | |
| 1234000 | | |
| 2300140 | | |
| 2034012 | | |
| 3200410 | | |
| 4030201 | | |
| 1230004 | | |
| 1233040 | | |
| 3210004 | | |
| 1320440 | | |

**b.** [2 points] How many processes have been created in total?
**c.** [3 points] How many processes executed the line “return 0”? Explain.

**Problem 3**

Suppose there is an OS that uses the three-state model (running, ready, and blocked) as we discussed in class. There is one CPU available. Four processes exist in the system. For each one of the situations mentioned in the following table, state whether this situation can happen or cannot happen. And next to each one, write 1-2 sentences explanation of your choice. You can neglect the time taken by the OS scheduler.
[5 points for each row: 2 for yes/no and 3 for justification].

| **Situation** | **Can happen? (yes/no)** | **Justification (1-2 sentences)** |
|---|---|---|
| All processes are in the *blocked* state. | | |
| All processes are in the *ready* state. | | |
| All processes are in the *running* state. | | |
| Two processes are in the *ready* state and two processes are in the *blocked* state. | | |
| One process in the *running* state and three processes in the *blocked* state. | | |

**Problem 4**

We have studied the scheduler and looked at some scheduling algorithms in class. Answer the following questions.

**a.** [5 points] In the shortest remaining time next (also known as Shortest-Time-To-Completion First) scheduling, assume we have one CPU and four processes: A, B, C, and D. Suppose all four processes are only CPU-intensive with no I/O. Assume also that no other processes will enter the system. That is, we have only these four processes. Currently process A is running on the CPU and processes B, C, and D are in the ready state. This scheduling algorithm is preemptive. Is there a scenario where A will be removed, before it completes its whole execution, from CPU and another process (B, C, or D) be put instead? Explain in 1-2 sentences.

**b.** [10 points] In round-robin scheduling, state five scenarios (in 1-2 sentences each) where a running process is removed from the CPU. Assume we have only one CPU.

**c.** [10 points] We have seen Multi-level Feedback Queue (MLFQ) scheduling in class. Answer the following questions. Do *not* give justifications to your answers.

- Is MLFQ preemptive or non-preemptive?
- The main goal of MLFQ is to increase the average response time of processes. True or false?
- The main goal of MLFQ is to increase the average turn-around time of processes. True or false?
- If all processes happen to be in the same queue, MLFQ acts as if it is FCFS scheduling. True or false?
- There may be cases where all processes happen to be in the same queue, and that queue is not the highest priority (i.e. the highest priority queue is empty). True or false?