

AUCSC 380 – Operating Systems Concepts – Winter 2021  
Assignment 1

**Due date: Feb 22<sup>nd</sup>, 2021, 12:00 am**

**Goal**

Gain some familiarity with the process management

1. Suppose we have a multiprogrammed computer in which each job has identical characteristics. In one computation period,  $T$ , for a job, half the time is spent in I/O, and the other half in processor activity. Each job runs for a total of  $N$  periods. Assume a simple round-robin scheduling is used, and I/O operations can overlap with processor operation. Define the following quantities:

- Turnaround time = actualtime = actual time to complete a job
- Throughput=averageThroughput = average number of jobs completed per time period  $T$
- Processor utilization=percentage utilization= percentage of time that the processor is active (not waiting)

Compute these quantities for one, two, and four simultaneous jobs, assuming that the period  $T$  is distributed in each of the following ways:

- a. I/O first half, processor second half [5 points]
- b. I/O first and fourth quarters, processor second and third quarter [5 points]

2. In your own words, explain what system calls are, and how do system calls relate to the OS and to the concept of dual-mode (kernel-mode and user-mode) operation? [10 points]

3. Create a text file named my input.txt that contains the following four lines:

Child 1 reads this line

Child 2 reads this line

Child 3 reads this line

Child 4 reads this line

Write a C program, named question3.c, that forks four other processes. After forking the parent process goes into wait state and waits for the children to finish their execution. Each child process reads a line from the file input.txt. Child 1 reads line 1, child 2 reads line 2, child 3 reads line 3 and child 4 reads line 4). Each child prints its process ID followed by the respective line. The lines can be printed in any order [10 points].

4. Write a C program that calls `fork()` and then calls some form of `exec()` to run the program `/bin/ls`. See if you can try all of the variants of `exec()`, including (on Linux) `execl()`, `execle()`, `execlp()`, `execv()`, `execvp()`, and `execvpe()`. Why do you think there are so many variants of the same basic call [10 points]?

5. Write a C program that opens a file (with the `open()` system call) and then calls `fork()` to create a new process. Can both the child and parent access the file descriptor returned by `open()`? What happens when they are writing to the file concurrently, i.e., at the same time? [10 points]

## Instructions

- You must hand in a separate pdf file for each coding questions (Q3, Q4, Q5), and one pdf file for questions 1 and 2.
- Each file must include your answer to the question, a fully documented source code of the program, and a testing of your program.
- The instructor will run some programs each time. These will be chosen randomly or if it seems the code does not match the testing submitted. If the code does not match the testing that is submitted, then a mark of 0 (zero) will be awarded for the assignment.
- A program that does not compile will get a mark of 50% at best, depending on documentation, style, and how much of the program appears finished. Please make sure your program compiles, even if incomplete

## Hand in on eClass:

- 1) One zipped folder, named as your first name that, contains
  - a. four pdfs. File names must be A12, A3, A4, and A5 (All files must be pdf)
  - b. source code files (your .c) for questions C3.c, C4.c, and C5.c.