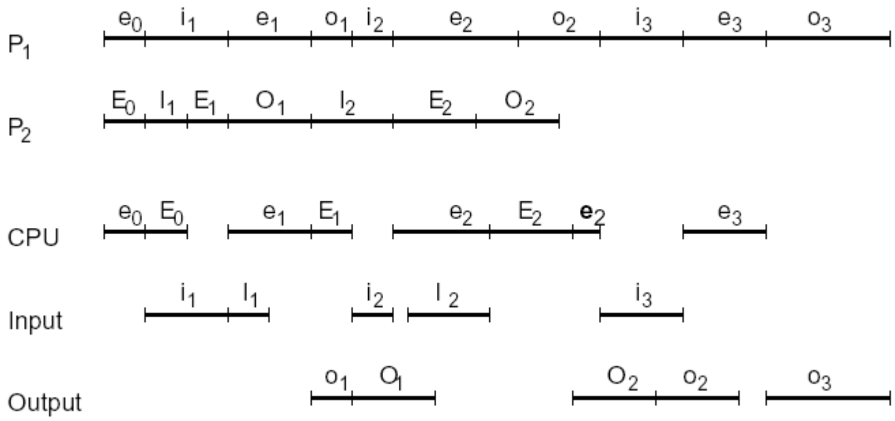
CSCI 3000 (Operating Systems)
Department of Computer Science
East Carolina University
Due February 20, 2023, 11:59PM (No Late Submission)

Instructor: Dr. Kamran Sartipi

NOTE: Refer to the last page for Submission Instructions and Marking Scheme

Question 1 [20 marks]. In the following timing, “ex”, “iy”, and “oz” represent execution, input, and output operations of process “P1”; and “Ex”, “Iy”, “Oz” represent execution, input, and output operations of process “P2”. With respect to this timing explain:

A. How the advantage of multi-tasking in term of CPU utilization is shown?
B. How can we increase the CPU utilization?



Question 2 [20marks]. Using either a UNIX or a Linux system, write a C program that forks a child process that ultimately becomes a zombie process. This zombie process must remain in the system for at least 15 seconds. Process states can be obtained from the command:
ps -l

The process states are shown below the S column; processes with a state of Z are zombies. The process identifier (pid) of the child process is listed in the PID column, and that of the parent is listed in the PPID column.

The easiest way to determine that the child process is indeed a zombie is to run the program that you have written in the background (using the &) and then run the command ps –l to determine whether the child is a zombie process. Because you do not want too many zombie processes existing in the system, you will need to remove the one that you have created. The easiest way to do that is to terminate the parent process using the kill command. For example, if the pid of the parent is 4884, you would enter: kill -9 4884

Question 3 [20 marks]. Consider the following operation:

$$n = \begin{cases} n/2, & \text{if n is even} \\ 3 \times n + 1, & \text{if n is odd} \end{cases}$$

When this operation is continually applied, all positive integers will eventually reach 1. For example, if *n* = 35, the sequence is:

35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1

Write a C program using the fork() system call that generates this sequence in the child process. The starting number will be provided from the command line. For example, if 8 is passed as a parameter on the command line, the child process will output 8, 4, 2, 1. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the wait() call to wait for the child process to complete before exiting the program. Also, perform necessary error checking to ensure that a positive integer is passed on the command line.

Question 4 [20 marks]. An echo server echoes back whatever it receives from a client. For example, if a client sends the server the string Hello there!, the server will respond with Hello there!

Write an echo server using the Java networking API described in Section 3.8.1 of the textbook. This server will wait for a client connection using the accept() method. When a client connection is received, the server will loop, performing the following steps:

- Read data from the socket into a buffer.
- Write the contents of the buffer back to the client.

The server will break out of the loop only when it has determined that the client has closed the connection.

The date server of Figure 3.27 of the textbook uses the java.io.BufferedReader class. BufferedReader extends the java.io.Reader class, which is used for reading character streams. However, the echo server cannot guarantee that it will read characters from clients; it may receive binary data as well. The class java.io.InputStream deals with data at the byte level rather than the character level. Thus, your echo server must use an object that extends java.io.InputStream. The read() method in the java.io.InputStream class returns −1 when the client has closed its end of the socket connection.

Question 5 [20 marks]. Design a program using ordinary pipes in which one process sends a string message to a second process, and the second process reverses the case of each character in the message and sends it back to the first process. For example, if the first process sends the message “Hi There”, the second process will return “hI tHERE”. This will require using two pipes, one for sending the original message from the first to the second process and the other for sending the modified message from the second to the first process. You can write this program using either UNIX pipes.

END

Instructions for Submission of Assignments

- For the descriptive questions submit a PDF file.
- For programming questions add "comments" to the source code, including:
    - Description for "classes" and "methods" (functions in C)
    - Comments for "variables" and most of the statements.
    - A "ReadMe.txt" file that clearly states the steps for compiling and running the source code, including the results for sample input.
    - Put each programming question in a separate folder with name "Ax-Qy" where "x" is the assignment number and "y" is the question number.
- Put all PDF files and programming folders in a folder with the name: "Gx-Ay-ddmm2023" where "x" is group number, "y" is assignment number, "dd" is date, and "mm" is month. Submit the "zipped" file of this folder to the Canvas system by the due date.
- Each group will submit only one assignment for the group.
- One of the students in the group will submit the assignment.
- (If indicated in the assignment) There will be 20% penalty for a one-day late submission and after that the dropbox will be closed. Otherwise, there is no late submission and the dropbox will be closed at the due date.
- **<u>Do Not</u>** send your assignment by email to the Instructor or the TA. It will not be accepted.
- Put the following statement in a separate word document and submit it with your assignment:

This assignment (Assignment X) has been discussed in our group practice meetings and the assignment solutions (including the source code) have been developed by our group (Group Y).

- Name: PirateID:
- Name: PirateID:
- Name: PirateID:

Marking scheme:

Q1: A: 10 marks; B: 10 marks

Q2 to Q5:

- Code compiles and runs: 4 marks
- Correctness: 10 marks
- Comments: 4 marks
- Read me text for each program: 2 marks