CS 4323 Design and Implementation of Operating Systems I

Assignment 01: Full Marks 100

(Due Date: 02/21/2021, 11:59 PM CDT)

This assignment is to be completed individually in C programming.

This assignment will be checked on CSX machine. Please make sure, they run without any compilation or runtime issue on CSX machine. Logging on to the CSX account and other relevant information can be easily accessed in: http://www.cs.okstate.edu/loggingon.html

Objectives:

- Familiarize the use of fork() system call to run multiple processes simultaneously.
- Familiarize the mechanism to communicate among processes, using POSIX IPC mechanism in UNIX

Programming Task:

You will have a main process, let us name it a Server process for ease of explanation. The Server process is going to read the input file items.txt and it is going to create a shared memory, where its keeps all of the information about the gift items reads from the file, using structure. The items.txt file contains 100 lines, each line has following 3 columns description about the gift item:

- first column represents the serial number, used to indicate the item number,
- second column represents the gift item,
- third column represents the price of the gift and the store where it is available.

You need to store this information about each product in 4 fields within the structure:

- serial number
- gift name
- price
- store

The Server process then asks the user to enter the value for *N* and then creates *N* number of Customer processes and 1 Helper process. The Server process also asks the user to enter an order for these *N* processes and passes this information to the Helper process. Each of these processes (both Customer and Helper) will be running in parallel. Each of these

- Customer processes will ask user to enter one integer number (let's say *M*), which will represent the number of gifts to be selected from the collection of gift items available in the shared memory. Then, each of these *N* number of Customer processes will read the items list from the shared memory (created by the Server process) and randomly picks the *M* number of items.

- Helper process uses the order to get the number of gift items from each of these N processes. For this communication, it is required to use message queue.

For example

if *N* = 3, then each there will be 3 Customer processes:

- Customer process A: if *M* = 2, then it will select 2 gift items
- Customer process B: if *M* = 5, then it will select 5 gift items
- Customer process C: if *M* = 4, then it will select 4 gift items

If the order selected is B, C and A, then the Helper process will get the collected item list from process B first, followed by process C and finally process A.

The Helper process then computes the cost of all gift items for each process and displays the summary result to the console. The summary result should include:

- Customer process ID,
- all list of gift items selected by that process (each line should be for one gift), and
- finally the total price for that process.

Once the Helper process displays the summary result and saves all the result in the output files: one for each process, it then signals the Server process that all the task is done and the program exits by Server process printing “Thank you”.

Programming requirements:

Your program should extensively use the following concepts:

1) Files are used for both reading and writing. Use appropriate output name for each process.

2) Two types of IPC mechanisms are to be used:
    a. Memory sharing between the Server process and the Customer processes as well as to Helper process.
    b. Message passing between the Customer process and the Helper process.

3) Structure arrays need to be used for gift items.

4) The Server process should wait till the Helper process signals it that all task is done.

5) There should be no zombie or orphan processes

The grading will be as follows:

- Correct use of memory sharing among Server process and the Customer processes and Helper process . **[30 Points]**
- Correct use of message queue among the Customer processes and Helper process **[30 Points]**
- Correct use of fork() system call. **[15 Points]**
- Correct functioning of Server process **[7 Points]**
- Correct functioning of Customer processes **[10 Points]**
- Correct functioning of Helper process **[8 Points]**

Submission Guidelines:

- You need to submit all the required C file with an extension .c. For example:
  - assignment00_lastName_firstName_<File name>.c
  - assignment00_lastName_firstName_<File name>.h

- All the C files should be submitted in the pdf format as well. Please make sure that you do not screen shot any code or save the program in the image form. All the codes need to be copied and pasted in the text form. For example:
  - assignment00_lastName_firstName_<File name>.pdf

- You need to include readMe.txt file which should include how to run your program.
  - readMe.txt

- In the assignment00_lastName_firstName_mainFile.c file, use the following header information:
  - Author Name:
  - Email: <Use only official email address>
  - Date:
  - Program Description:

- Use comments throughout your program.

- Each function should be properly commented:
  - Mention each argument type, purpose
  - Function description
  - Return type of the function

- Failure to follow standard programming practices will lead to points deduction, even if the program is running correctly. Some of the common places where you could lose points are:
    - Program not compiling successfully: -20 points
        - The TA will run the program using the input file items.txt in CSX machine.
    - No comments on code (excluding function): -5 points
    - No comments on function: -5 points
    - Not writing meaningful identifiers: -5 points
    - Failure to submit files as specified: -10 points