Mid Sem Exam 2021

Operating System Lab (CS342)

Department of CSE, IIT Patna

Date:- 16-March-2021 Deadline:- 17-March, 10 AM

Instructions:

1. All the questions should be completed and uploaded before the deadline. Marks will be deducted for the submissions made after the deadline.
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. Proper indentation & appropriate comments (if necessary) are mandatory. [2+2 marks]
4. You should zip all the required files and name the zip file as roll_no.zip, eg. 1501cs11.zip.
5. Provide a **readme** file with all the execution details (commands to execute) of the codes and outputs/observations (if necessary).
6. Upload your assignment (the zip file) in the following link:
https://www.dropbox.com/request/4Ug0uFnUWS2ViaSDPZRZ

Q1. Write a program (c\c++) to execute the zombie and orphan process in a single program.

Q2. Fork system call

An important alternating series is the Alternating Harmonic Series defined as .

$$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \cdots$$

Write a c program to do the following task using fork system calls. Take the value of n from the user using the command line argument. Create three child processes to calculate the output of the given series.

- Child 1 should compute the sum of negative numbers in the series.
- Child 2 should compute the sum of positive prime numbers in the series.
- Child 3 should compute the sum of positive non-prime numbers in the series.

Parent process should print the final result of the series.

Threads

We have following properties of matrix addition:

1. Commutative Law

If A = [aij], B = [bij] are matrices of the same order, say m × n, then A + B = B + A.

2. Associative Law

For any three matrices A = [aij], B = [bij], C = [cij] of the same order, say m × n, (A + B) + C = A + (B + C).

3. Existence of Additive Identity

Let A = [aij] be an m × n matrix and O be an m × n zero matrix, then A + O = O + A = A. In other words, O is the additive identity for matrix addition.

4. Existence of Additive Inverse

Let A = [aij]m×n be any matrix, then we have another matrix as – A = [–aij]m×n such that A + (–A) = (–A) + A= O. So – A is the additive inverse of A or negative of A.

Take any three matrices, A, B, and C as input from the user and Create four threads to check the correctness of all the four properties of matrix addition.

- Thread 1 will check Commutative Law and prints the output as “Commutative”.
- Thread 2 will check Associative Law and prints the output as “Associative”.
- Thread 3 will check Additive Identity and prints the Additive Identity Matrix as output.
- Thread 4 will check Additive Inverse and prints the Additive Inverse Matrix as output.

Q3. Priority Scheduling is a method of scheduling processes that are based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on an FCFS basis. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

Consider n processes, P1, P2 .. Pn. Write a program to find out the average waiting time (WT), the turn-around time(TAT), and the completion order of the processes using the Priority preemptive scheduling algorithm (in case of conflict, the process with smaller process id will execute first).

The number of processes and for each process, the arrival_time, priority, and burst_time are to be considered as input.

Q4. Write a shell script program that will take a filename(.text : F) and a number (K) as arguments and do as follows :

i. It selects all the words which occur with frequency greater than K from the file F and replaces the all occurrences of such words from all files within the directory with the word “MASKED“.

ii. Find a list of ‘K’ file names of the directory (in descending order of no. of “MASKED” word), which contains at least K/2 “MASKED” words.