

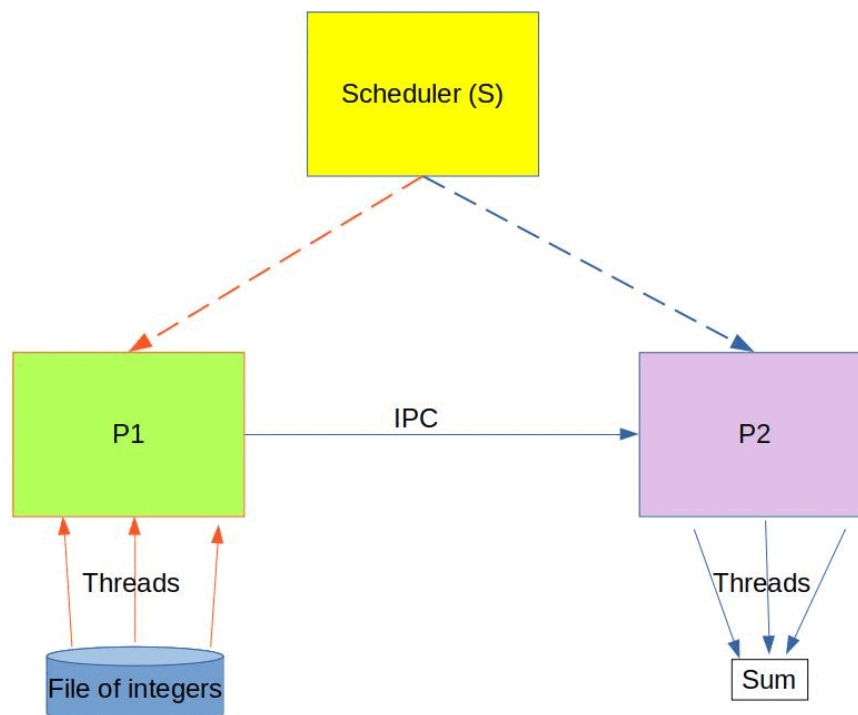
CS F372 Operating Systems
Second Semester 2021-22
Assignment 2
Total Marks: 60
Time allowed: 15 days
Deadline: 15th April 23:59 hours

- All programs are to be written in C
- You may use the same groups as in Assignment 1. If you are changing your groups, the individual late days for assignment 1 will carry over for you. That is, if you are moving from a group which has used 2 late days, to a group which has not used any late days, 2 late days will still apply to you individually.
- All members of the group must contribute to the assignment. A demo-cum-viva session will be scheduled during which it will be ascertained that all members have contributed. If any member does not contribute, the member will be awarded 0 in the assignment.
- All submissions will be passed through a code and text similarity checker. If the codes of two or more groups match then all the group members will be summarily awarded 0 in the assignment. This is irrespective of if only one member of a group is the offender. There is no partial penalty for dishonesty. Lifting code from the Internet also constitutes cheating.
- Honest but incorrect submissions will be awarded partial credit through the demo-cum-viva session
- Discussion is encouraged between groups, copying is strictly prohibited. What is the difference? Check the last part of this page:
<https://www.cse.iitd.ac.in/~mausam/courses/col772/spring2019/>. Use Piazza for discussions.

Problem Statements:

- a) Write a multi-threaded C program (P1) that reads a file containing space-separated positive integers in ASCII format. Some example input files are [here](#). Please generate more input files in order to generate the plots as required below. The number of integers can be arbitrarily large (upto 1 million integers). The name of the file and the number of integers in the file will be supplied in the command line.
 - i) Different threads should read different parts of the file. Vary the number of threads from 1... to arbitrarily large
 - ii) Record the time that it takes to read the entire file into memory with different number of threads (1, 2, ... n)
 - iii) Plot time against the number of threads for different numbers of integers in the input file. Analyse the findings from the plots.
- b) Write a C program (P2) which uses IPC mechanisms to receive the numbers read in the program in part (a). This program spawns multiple threads to sum the numbers passed by the program P1 in part (a). The program prints the sum on the terminal.
 - i) Vary the number of threads from 1... to arbitrarily large. Different threads can calculate the partial sums of the series

- ii) Record the time it takes to sum the integers with different number of threads
 - iii) Plot the time against the number of threads for different numbers of integers.
Analyse the findings from the plots.
- c) Write a scheduler program S. S spawns 2 children processes which exec to become the processes P1 and P2 in part (a) and part (b). S uses some mechanism (e.g. sleep) to simulate a uniprocessor scheduler. That is, it puts Process 1 to sleep and lets Process 2 execute, and vice versa. Simulate the following scheduling algorithms in S:
 - i) Round Robin with time quantum 1 ms
 - ii) Pre-emptive priority scheduling, where the priority is a function of the amount of processing time that a process has already received (lesser the processing time received, higher the priority). Priority is updated every 1 ms.
 - iii) Plot the total turnaround time vs workload size and waiting time vs workload size for the different scheduling algorithms. How do the two algorithms compare for the same workload size? Analyse your findings.
 - iv) What is the switching overhead in the different cases?
- d) You might need to take care of race conditions which might arise at different parts of the assignment



What to submit:

A single zip file for each group containing:

1. A text file with the details of the group members, including the names and BITS email addresses
2. C program(s) for the three parts above
3. A PDF file containing the analysis as required above. Credit will be given for plots and analysis which clearly elucidate what is happening. You are free to innovate on the plots and the analysis to earn credit. You can use any tool to generate the plot, including spreadsheets
4. Submit on CMS. It is sufficient for one group member to submit.

Evaluation:

Evaluation will be done through a demo-cum viva session where your code will be checked for correctness using different test cases. It will be checked that you have satisfied the different requirements of the assignment. Your report will be checked and questions will be asked regarding the code and analysis. General questions related to concepts in the assignment might also be asked. All group members must know all aspects of the assignment.