

CS703 – Advanced Operating Systems

Assignment No. 2

Instructions to Solve Assignments

The purpose of the assignments is to give you hands on practice. It is expected that students will solve the assignments themselves. Following rules will apply during the evaluation of assignment.

- Cheating from any source will result in zero marks in the assignment.
- Any student found cheating in any two of the assignments submitted will be awarded "F" grade in the course.
- No assignment after due date will be accepted.

Question No. 1 (10 marks)

Give two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?

Question No. 2 (10 marks)

Which resources are used when a thread is created? How do they differ from those used when a process is created?

Question No. 3 (10 marks)

In the “Too Much Milk” problem discussed in the lecture, what is the flaw in solution#2?

Question No. 4 (10 marks)

Revive the Readers/Writers problem discussed in lecture 12, write the code for Reader() and Writer() functions, when readers are given priority over writers, keeping the problem constraints in mind?

Question No. 5 (8 + 8 + 8 + 1 = 25 marks)

Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5 , all at time 0.

- Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.
- What is the turnaround time of each process for each of the scheduling algorithms in part a)?
- What is the waiting time of each process for each of the scheduling algorithms in part a)?
- Which of the schedules in part a) results in the minimal average waiting time (over all processes)?

Question No. 6 (35 marks)

Programming Question

Write a multithreaded Pthread program that generates the Fibonacci series. This program should work as follows: The user will run the program and will enter on the command line the number of Fibonacci numbers that the program is to generate. The program will then create a separate thread that will generate and display the Fibonacci numbers. Attach all the source code files along with the assignment solution file. Also, tell the command to compile the program.