

CS550 “Advanced Operating Systems”
Homework 1

Submission:

This is an INDIVIDUAL written assignment.

Due by 2/12/2024

Total points 100 - Late penalty: 10% penalty for each day late

Please upload your assignment on Blackboard with the following name:

CS550_SectionNumber_LastName_FirstName_HW1.

Please do NOT email your assignment to the instructor and/or TA!



1. **(10 points)** In this task, you are asked to evaluate the efficiency of reading a file with a single-threaded and a multithreaded file server. It takes 25 milliseconds to process a request, dispatch it, and perform the necessary tasks if the required data is stored in the main memory cache. If a disk operation is needed, which occurs 1/3 of the time, an additional 75 milliseconds is required, during which the thread goes to sleep.

a) What is the maximum number of requests per second that the single-threaded server can handle?

b) What is the maximum number of requests per second that the multithreaded server can handle?

2. **(10 points)** Imagine a series of processes P_1, P_2, \dots, P_n that implement a multitiered client-server architecture. Process $P(i)$ acts as a client to process $P(i+1)$, and $P(i)$ only sends a reply to $P(i-1)$ after receiving a response from $P(i+1)$. What are the main challenges with this setup when examining the request-reply performance at process P_1 ?

3. **(10 points)** Not every node in a peer-to-peer network should become superpeer. What are reasonable requirements that a superpeer should meet?

4. **(10 points)** Enumerate all components of the program state that are shared among threads in a multithreaded process.

5. **(10 points)** Would it make sense to limit the number of threads in a server process?

6. **(10 points)** Outline the pros and cons of utilizing multiple processes versus multiple threads.

7. **(10 points)** Outline the pros and cons of preemptive scheduling versus non-preemptive scheduling.

8. **(10 points)** Is a multi-threaded approach always advantageous in terms of performance? Provide an explanation with supporting reasons.

9. **(10 points)** List two distinctions between user-level and kernel-level threads. When is one preferred over the other?

10. **(5 points)** Describe a simple scheme in which there are as many lightweight processes as there are runnable threads.

11. **(5 points)** Explain the difference between a process and a thread, including which uses more resources.