

Final Test  
Changpan Zheng NUID:002107943

Q1 : Consider the following snapshot of a system:

	Allocation	Max	Available
	ABCD	ABCD	ABCD
P0	0012	0012	1520
P1	1000	1750	
P2	1354	2356	
P3	0632	0652	
P4	0014	0656	

Answer the following questions using the banker's algorithm:

a. What is the content of the matrix Need?

The Need matrix is calculated as the difference between the Max matrix and the Allocation matrix.

	Allocation	Max	Need	Available
P0	0012	0012	0000	1520
P1	1000	1750	0750	
P2	1354	2356	1002	
P3	0632	0652	0020	
P4	0014	0656	0642	

b. Is the system in a safe state?

To determine if it is in a safe state, we can use the banker's algorithm.

Initially, the available resources are ABCD = 1520.

Firstly, P0 only need 0000, so we can complete P0, and then the available will be 1520+12=1532.

Then, P2 can be completed, available will be 1532 with 1354 become 2886

Then, P1 can be completed, available will be 2886 with 1000 become 3886

Then, P3 can be completed, available will be 3886 with 0632 become 3,14,11,8

Then, P4 can be completed, available will be 3,14,11,8 with 0014 become 3,14,12,9

Therefore, the system is in a safe state since the sequence <P0, P2, P1, P3, P4> can complete their execution without entering a deadlock state.

Q2. Create a nodejs code which implements a get method where you send "Hello {Name}" as the body and the response body is "Welcome {Name}"

2. Seen as the attachment.

Q3: Provide two programming examples in which multithreading provides better performance than a single-threaded solution.

1. Image processing: When you want to change many pictures at the same time, you can use different groups to work on them at the same time. This makes it faster because the computer can use all of its parts at once and finish the work more quickly.
2. Web server: When many people want to look at a website at the same time, a website can use different groups to handle each person's request. Each group can work on one person's request, and the website can respond to many people at once. This makes the website work better and faster when many people want to see it.

Q4. Provide two examples where multithreading does not provide better performance than single threaded.

1. Simple tasks like Calculator: For tasks that are simple and don't require much computational effort, having multiple threads may actually slow things down due to the extra work of creating and managing threads. In these cases, a single-threaded solution might be better because it's simpler and has less overhead.
2. Serial dependencies: If tasks have strict dependencies on each other and need to be synchronized, like tasks that must be executed in a specific order or tasks that use shared resources, multithreading may not be more efficient. Managing synchronization between threads can add overhead and reduce performance gains, making a single-threaded solution with proper sequential execution potentially more efficient.

Q5: Describe the differences among short-term, medium-term, and long-term scheduling.

Short-term scheduling manages the execution of processes on the CPU.

Medium-term scheduling manages the changing of processes between main memory and secondary storage.

Long-term scheduling manages the admission of processes into the system.

Each type of scheduling plays an important role in optimizing system performance and managing system resources in an OS.

Q6. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use nonpreemptive scheduling and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

What is the average turnaround time for these processes with the FCFS scheduling algorithm?

process	Time of Process	Arrival Time	Complete Time	Turn Around Time	Wait Time	Response Time
P1	8	0.0	8	8	0	0

P2	4	0.4	12	11.6	7.6	7.6
P3	1	1.0	13	12	11	11

Avg. Turn Around Time =  $(8 + 11.6 + 12) / 3 = 10.53$

Q7: Write code for following system:

Person has a bank account.

There are 4 threads adding money in the bank account.

There are 5 threads taking money out of the bank account.

Make sure that threads cannot take out money which is not there i.e. synchronization of the bank account

Make sure that the account tally is always correct.

This system will run forever.

Seen as the attachment.