International Institute of Information Technology, Hyderabad

(Deemed to be University)

CS3.301 Operating Systems and Networks – Monsoon 2024

End Semester Examination

Max. Time: 3 Hr **Max. Marks: 70**

Instructions to the students:

1. Answers written with pencils won't be considered for evaluation.
2. Please **read the descriptions** of the questions (scenarios) **carefully**.
3. There are a total of ten questions with five MCQs and carries total of 70 marks. For MCQs you can select one or more options (if necessary). Please refrain from writing long explanations for MCQs. **Keep the explanations short and to the point. Note that explanations are mandatory for MCQs**
4. Please **state any assumptions** made clearly.
5. Use pseudocode when necessary (you are not required to provide the exact C implementation).

Good Luck

Welcome to the EduOS Design Challenge!

EduOS is a Unix-based Operating System that is being designed and developed by a team of open-source enthusiasts to create an OS for enabling students and enthusiasts to learn and master the concepts of Operating Systems and Networks. The team is following classic textbooks in operating systems and networks to build this system to ensure that the system is representative of the concepts that the students will be learning during OSN course. The OS part is being handled by three teams: One team focussing on process and memory virtualization, The second team focussing on concurrency and the third team focussing on Persistence. Apart from this, there is also a network team that is responsible for building the network stack as per the OSI model. The goal is to create a simple, educational operating system that is modular, efficient, and easy to understand, while also ensuring compatibility with essential concepts of Operating Systems and networks.

Recognizing that poor design practices are a common issue resulting in long-term maintenance challenges, the team of EduOS needs your help (since you have just finished your OSN course) to review their existing design and provide guidance to the teams regarding the design of the overall EduOS in general as well as certain specific aspects.

As the expert, you are given 180 minutes to go through the issues and provide solutions. You must complete these tasks within the allocated timeframe, as the release date of EduOS is approaching and they don't have more time to gather feedbacks. Each task is assigned a specific number of contributor points, and the cumulative score of all the tasks will determine your final contributor points with a maximum of 70 points. Highest contributor will be listed in the website of EduOS as "Star OSN contributor".

1. During your review of EduOS, you observe that the processes fall into three categories: regular programs that typically run for 20 ms and utilize only the primary memory, an email checker that periodically executes for an average of 8 ms to check for new emails, and background processes that run for about 7 ms on average, primarily performing disk operations. Given this setup, the team seeks your guidance on several aspects:

a. Considering the different types of processes that are executing, what mechanisms do you think the OS can use to ensure that processes can share the CPU (considering it is a single core one). Further, what are the different states that the OS needs to consider in order to appropriately schedule the process **(4 points)**

b. The Virtualization team requires your help in deciding whether to use a pre-emptive scheduler or a non-preemptive scheduler. What would you suggest given the above scenario? Explain by providing an example of one pre-emptive and non-preemptive scheduler for the above scenario and compare them based on appropriate metrics **(6 points)**

c. Further, the Virtualization team has decided to use paging for memory management. The team is considering a 32-bit address space with 2 KB pages. The team estimates each entry in the page table to consume 4 bytes. What according to you will be the approximate memory overhead of using paging if the team estimates the OS to support around 100 processes? How will this change if the team plans to use a two-level page table? Are there any advantage of using two-level page table in this context? **(5 points)**

2. In EduOS, all processes rely on system calls to read from or write to the disk. For a **read operation**, an OS thread retrieves data from the disk and writes it to a shared buffer, from which the process can read. Conversely, for a **write operation**, the process writes data to the shared buffer, which is then written to the disk by an OS thread. Given the limited memory, all processes share a **single common buffer**, which creates contention when multiple processes attempt to access it simultaneously. To ensure consistency and prevent race conditions, the concurrency team requires a solution to synchronize access to the shared buffer. How do you think a synchronization mechanism for the shared buffer can be designed? Provide a pseudocode solution that ensures safe access to the buffer during concurrent reads and writes. Clearly state the assumptions made and explain how your solution guarantees synchronization without causing deadlock or starvation. **(10 points)**

3. The Persistence team is considering to integrate a simple file system (SFS) with 64 blocks assuming an underlying disk of size 256KB. In the SFS each block is of size 4KB. Consider that for each i-node, SFS stores 256 bytes worth of Metadata. The team wants to dedicate space of 224KB to store the data of the files. The team is planning to dedicate 5 blocks for i-node. Given this structure, the team needs your help in understanding the following:
   a. How many files can your SFS store at any point of time? **(2 points)**
   b. Considering that each i-node has 48 bytes reserved for direct pointers, where each pointer takes 4 bytes of size. The team is using a file system scheme where each pointer directly references a single data block to enable file access. Considering that SFS only supports direct pointers, What is the limit on the file size that SFS can store and access? Considering that SFS can support not just direct pointers, how can this be enhanced to support access/storage of a file up to a size 4 GB (Assume that underlying disk has total storage size of 8 GB). **(4 points)**
   c. The persistence team wants to increase the storage of the underlying disk from 8 GB to possibly 32 GB. At the same time, the team wants to ensure that the disk can tolerate failure with support for hot swap. Performance is not a major concern but the system

should provide good sequential reads, writes and random reads and writes. What mechanism do you suggest? Elaborate on the approach and explain how it will help navigate the constraints **(4 points)**

4. The Network team has identified that at a broad level the communication between a source and destination needs to be supported in two scenarios. To this end, the team wants to know from you how this can be achieved in each of the following scenarios. They have a machine with an IP address 172.168.10.46, MAC 1B:2C:3D:4E:5E:6F, with a subnet mask of /24. The public IP address of the network is 21.53.19.90, and the IP address of the router is 172.18.12.92; MAC address 1D: 7G:98:1C:1A:2D. The team wants to understand how the data transfer can be achieved in the following cases:
   a. The destination machine has an IP address of 172.168.10.95 with a subnet mask 255.255.255.0, MAC 8A:9B:1C:2F:9F:6D. Please note that the source machine does not have the MAC address of the destination machine as well. can you help the team understand how this communication can work with clear reasoning **(4 points)**

   b. The destination machine has an IP address of 192.168.21.18 with a subnet mask of 255.255.255.0, MAC: 9F:1B:3C:4F:7A:5A. The public IP of the destination machine network is 221.18.21.62. The destination machine is part of a larger organization which is in a different autonomous domain compared to the source network. The team requires your help in understanding how the overall flow of data happens from the source machine to the destination machine. **(7 points)**

   c. As a next step, the team wanted to transfer a file to a machine within the same network using an application layer protocol running over TCP. Assuming that the file contains data: 1110011001100110110101010101010101. Can you explain the team the process that happens to enable the successful transfer of data from the transport layer perspective? **(4 points)**

5. As decided, the virtualization team has used paging for memory management where they have devised a technique using swap space to manage even processes with large virtual address space. However, the exact policy for page replacement has yet to be decided. The team has two types of traces of page access in mind, and they are looking at the main memory as a cache with a size of 3 pages:

   Trace 1: 0, 1, 2, 0, 1, 3, 0, 3, 1, 2, 1

   Trace 2: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

   The team is deliberating between different page replacement policies. They would like to know from you which policy to use. Which policy do you recommend and why? Demonstrate using the necessary metrics and the trace table for the policy you will suggest for the team given the scenario. Compare it against the Optimal Page Replacement Policy (the baseline). What if main memory can hold up to 4 pages? **(5 points)**

6. The Network team has implemented the UDP protocol in the stack and now they wanted to try something out of curiosity. The team wants to implement a TCP on the top of the UDP protocol. According to you, which among the following are essential to consider for making this happen and how? **(3 points)**
   a. Sequence and acknowledgement number
   b. Flow control

c. Congestion Control
d. Checksum
e. All of the above

7. The concurrency team is considering the usage of a Spin lock to address concurrency problems between two threads. However, few members of the team feel that there is no single use case where spin lock can actually work better compared to using semaphores. What is your opinion and why? **(3 points)**
   a. Spin locks can work better than semaphores when critical sections are short
   b. Spin locks can work better than semaphores when there are large number of threads
   c. Spin locks are always better than semaphores
   d. Semaphores are always better than spin locks
   e. None of the above

8. As the network team continued building the network stack, they wanted to try the effectiveness of the stack implemented. They observed that, whenever they were connecting a node for the first time to a network, there is a broadcast request that happened from port number 68 and the node was getting a response from few servers listening on port 67. The team is trying to understand what is happening in this situation. What do you suggest and why? **(3 points)**
   a. Its due to the DNS request that is being sent
   b. The client machine is using NAT to manage the public IP
   c. The client machine is making a request using DHCP
   d. ARP request is being broadcasted to all machines for updating the ARP table
   e. None of the above

9. The Synchronization team has implemented the notion of semaphores in their UNIX based OS. However, they further wanted to test by putting it to use to solve some classical problem. They have come up with the following snippet to solve the producer consumer problem. The idea is to use a buffer of size 10 integers. They want your opinion on if the written snippet contains some synchronization issues or if it is good to be deployed. What would be your opinion and why? **(3 points)**

Producer
```
void *producer(void *arg)
{
    ... //for loop
    sem_wait (&mutex);
    sem_wait (&empty);
    put(i);
    sem_post (&full);
    sem_post (&mutex);
}
```

Consumer
```
void *consumer(void *arg)
{
    .. //for loop
    sem_wait (&mutex);
    sem_wait (&full);
    int tmp = get();
    sem_post (&empty);
    sem_post (&mutex);
}
```

   a. This program does not have any synchronization issues
   b. This code can result in a deadlock
   c. This program will work only if there is one producer and one consumer
   d. The program will work for a buffer size of 1
   e. None of the above

10. The Network team is looking to build application layer support in their EduOS stack. The team is considering using HTTP/1.1 and thereby built an HTTP server, but since HTTP is stateless, they are not sure how to implement the mechanism to store some state information. What do you recommend? Please mark your option followed by explanation on how it can help address the concern **(3 points)**
    a. The team can make use of cookies
    b. Web caches can be leveraged
    c. The team should use HTTP/1.0 instead
    d. Make use of CDNs
    e. None of the above

***************Now it's time to wait for final contributor points*****************