INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLGY, SHIBPUR

MID-TERM EXAMINAYTIONS, 2022

Operating system (CS 3201)

F.M. 50 TIME: 45 MIN

Question no. 1 is mandatory – answer any three from the rest

Name: Abhiroop Mukherjee

Enrolment Number: 510519109

Gsuite id: 510519109.abhirup@students.iiests.ac.in

1. i) What is the difference between timesharing and multiprogramming systems?

Answer:

Time Sharing	Multiprogramming
In Time Sharing System, processes/users are allocated with resources in a respective time slots.	Multiprogramming Systems allows multiple processes simultaneously by monitoring their states and switching in-between processes.
Two or more users can use a processor in their terminal.	A process can be executed by a single processor.
Time Sharing Systems has fixed time slice.	Multiprogramming Systems has no fixed time slice.
In Time Sharing System, execution power is taken off before finishing of execution.	In Multiprogramming System, before finishing a task the execution power is not taken off.

ii) On early computers there were no DMA. What could be the implications of multiprogramming in such a system?

Answer:

The main reason for providing DMA is to give the CPU flexibility to perform any other task while the I/O is being performed by CPU.

If there is no DMA in the multiprogramming system then the CPU will be fully occupied by the I/O devices, eventually leading no advantage of having a multiprogramming system benefit.

iii) I/O devices are handled with privileged instructions --- why?

Answer:

Modern OS provides minimum two modes of operation: user mode and kernel mode.

Privileged instructions (that can possibly harm the system when executed by malicious process) can be executed only in kernel mode.

I/o instruction falls in this category and are entrusted to OS. If I/O instructions are entrusted to users, they may misuse them and cause havoc to the computer.

iv) Many early processors could run only in one mode and there were no privileged instructions – what is the implication to offer a good OS for machine with those CPUs

Answer:

Lack of Kernel and User mode distinction means that there is no protection for computer by the user's code. This gives full access of computer to the user which might cause a lot of problem. User may mess up some important part of OS and could brick the computer.

OS made in this type of system need to have a check (like a static address checking) to make sure this does not happen. One way they can do this is by scanning the whole user program first for any danger before executing

v) Modern operating systems decouple a process address space from the machine's physical memory. List two advantages of this design. [4 x 5]

Answer

a) There's a protection system built with this approach as process generating VA can by checked if it is going to other processes segment and can by blocked be OS.

b) OS can change data location to anywhere for memory management while process is not hampered with it.

2. A system has 8-GB RAM of which 256 MB is used by the OS and the page table size is 256 MB. If the processes are all taking 512 MB and has similar operational characteristics then to achieve 85% of CPU utilisation what could be the maximum I/O wait time in percentage that can be accepted. [10]

3. Consider the following solution to the mutual-exclusion problem involving two processes P0 and P1. Assume that the variable **turn** is initialized to 0. Process P0's code is presented below.

```
/* Other code */
while (turn != 0){  /* Do nothing and wait. */
}
 Critical Section /* . . . */
 turn = 0;
/* Other code */
```

For process P1, replace 0 by 1 in above code. Determine if the solution meets all the required conditions for a correct mutual-exclusion solution. [10]

Answer:

For a technique to solve the mutual exclusion problem, the solution should satisfy the following three conditions

I) Mutual Exclusion
   - No two processes concurrently be in their critical sections

II) Progress
   - If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder sections can participate in deciding which will enter its critical section next, and this selection cannot be postponed indefinitely

III) Bounded waiting
   - There exists a bound, or limit, on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

Given the solution to solve the mutual-exclusion problem involving two processes P0 and P1, if there exist is a speed difference between P0 and P1, a process will have to wait for the slower process to complete before going into its critical section.

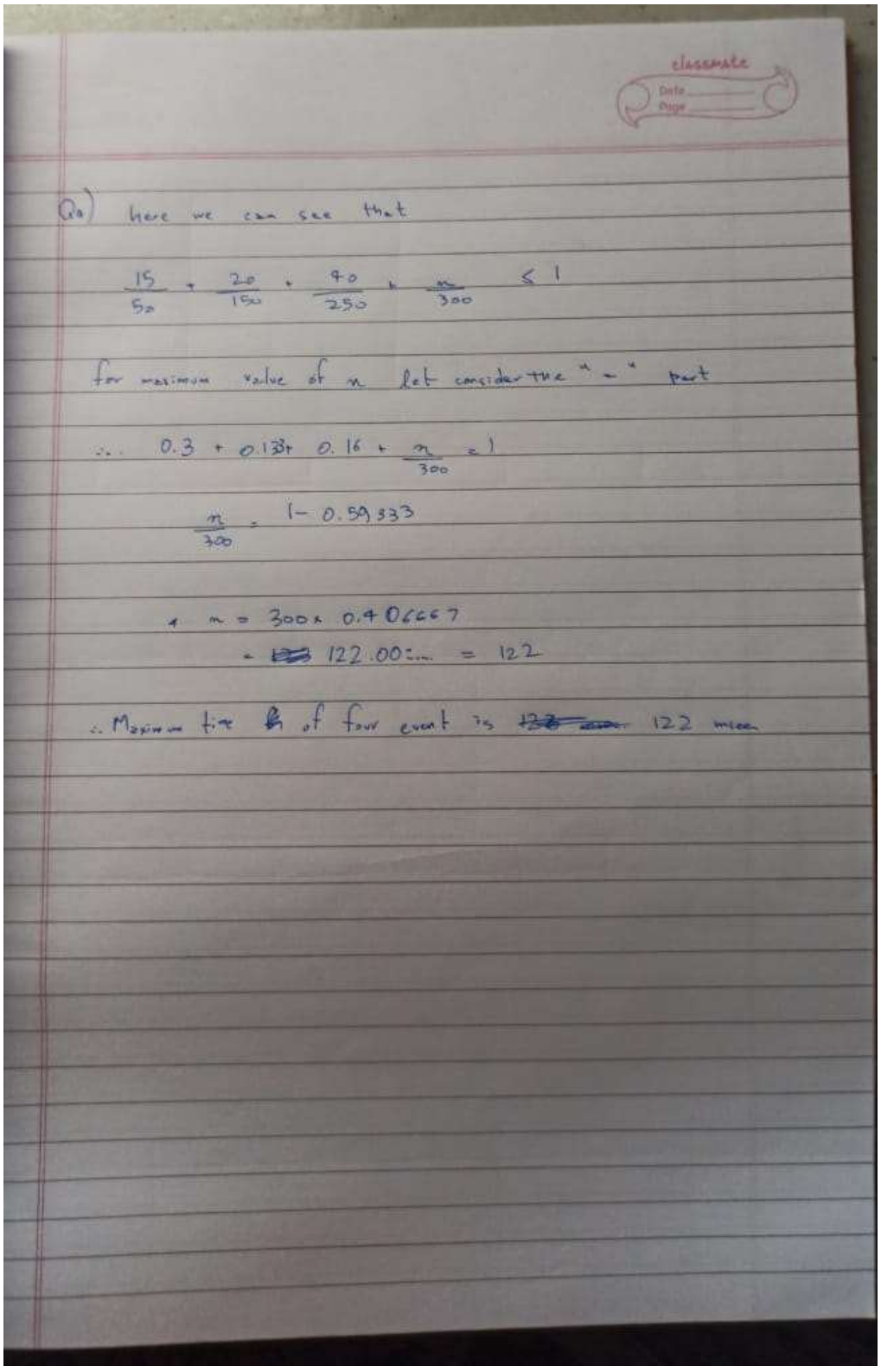
Consider the following example:

- P0 is in its CS and while leaving sets it to 0.
- P1 enters its CS and quickly completes it and made turn = 1. Now both the processes are in their non-CS portion and turn = 1.
- Now, P0 moves to its CS and immediately completes it and sets turn = 0. now both the processes are in their non-CS section.
- Now, P0 tries to enter to its CS and will wait for the turn to become 0, while P1 is not in a critical section.
- This breaks the rule II above as P1 executing remainder section is deciding that a P0 should wait before going to a critical section.

This method follows rule I as P0 and P1 will not be executing concurrently.

This method also follows Rule III as the bound here is 1. For example, the P0 will go to its critical section after P1 is done with its turn.

4. A soft real-time system has four periodic events with periods of 50, 150, 250, and 300 msec each. Suppose that the four events require 15, 20, 40, and X msec of CPU time, respectively. What is the largest value of X for which the system is schedulable? [10]



Ans) Here we can see that

$$\frac{15}{50} + \frac{20}{150} + \frac{40}{250} + \frac{n}{300} \leq 1$$

for maximum value of n let consider the "=" part

$$\therefore 0.3 + 0.133 + 0.16 + \frac{n}{300} = 1$$

$$\frac{n}{300} = 1 - 0.59333$$

$$\therefore n = 300 \times 0.406667$$

$$= 122.00\ldots = 122$$

∴ Maximum time of four event is 122 msec

5. What are internal and external fragmentations. Mention the advantages and disadvantages of segmentation. [10]

Answer:

Internal Fragmentation

Internal Fragmentation is a phenomenon experienced in an OS using Process Virtualization to provide private address space to multiple process.

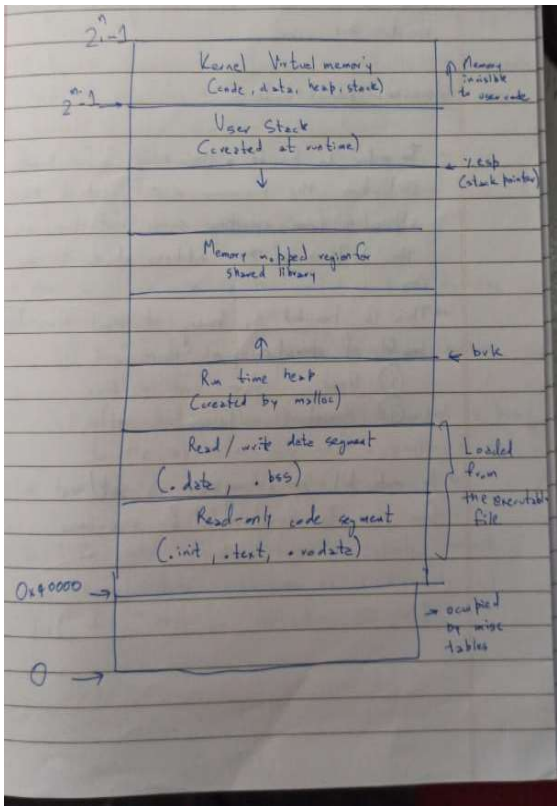
Virtualization is done by OS to give a process an illusion that the process has entire access of the memory using hardware base and bound registers.

OS gives a section of memory to process to by setting base and bound registers in the Process Control Block (PCB) of the respective process. In this part of memory, the process views the entire part of the memory has its own and uses address as if that allocated part of memory starts from address 0. Any address generated by the process (called Virtual Address (VA)) is converted to actual Physical Address by some offsetting the VA by the base register value (if the address request passes all the preliminary checks)

Only one base bound register pair are required here and, in the memory location provided the process has all of its data which includes the code, stack (growing downwards), heap (growing upwards) and more. (Figure added)

Here with many processes we can see that the space between stack and memory are not being used but are being allocated to processes.

This is called Internal Fragmentation.



External Fragmentation

In segmentation multiple Base bound pair are present per process to segment the parts of program. This removes internal fragmentation, but this introduces small gaps in segments itself in the momory.

This is called external fragmentation.

Advantages of Segmentation

- Better sparse space management
- Faster translation using hardware
- Sharing of code (and r/o data; may be a table)

Disadvantages of Fragmentation

- Segmentation cannot fully manage the sparce space.

- For example, say, a sparsely used heap is on the memory as a single logical segment it would remain in memory to be accessed every now and then.

-

6. For a 12-bit VA show the translation mechanism if the page size is 512 bytes. Assume that the PT is a linear array with only one entry. What are the standard bits added in a PTE for different administrative action by the OS? [10]