Theory Assignment 1

By

Keyur Patel

40154883

A report submitted in partial fulfilment of the requirements of COMP346

Concordia University

February, 2022

**Answer 1:**

1. An operating system in simple words is a system software and is a program that manages a computer’s hardware. The purpose of an operating system is to provide a basis for application programs and to act intermediary between the computer user and the computer hardware.
2. Batch: needed for efficiency for multiprogramming

Timesharing: creates interactive computing so users can interact with multiple jobs  
Dedicated: used for a single purpose program without the need of switching to another program

Real-time: processes data in real-time without the use of a buffer in the computer memory  
Multiprogramming: organizes jobs so CPU only has one to execute

1. Time sharing would make sense when there is more than one user who use the machine, but it is a huge task. Since the number of users would be limited, the power of the one task can be shared by the hardware, and it would not be harmful to the overall performance. For a small task, a personal computer is best since it would suit to a user’s own needs, and it does not require multiple users to work on such a task.

**Answer 2:**

|  |  |  |  |
| --- | --- | --- | --- |
| Process/ Time | CPU | I/O | CPU |
| P1 | 15 | 10 | 10 |
| P2 | 10 | 5 | 15 |

For single programmed system, we must traverse through all processes one by one so it would look like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 15 | 10 | 10 | 10 | 5 | 15 |

The minimal time required to complete execution of two processes is 65 units.

1. For the multi-programmed, one process would execute while the other awaited I/O, implying that we would fully ignore the I/O time in the calculation.

|  |  |  |  |
| --- | --- | --- | --- |
| 15 | 10 | 10 | 15 |

The total execution time is 50 units.

1. Single programmed throughput: Time/number of processes = 65/2 = 32.5

Multi-programmed throughput: Time/number of processes = 50/2 = 25

Multiprogramming reduces throughput by performing multiple tasks and reducing CPU time.

**Answer 3:**

1. The advantage of interrupt over polling is that if the CPU wants attention, it will interrupt the current process and give it that attention. This ensures that the device does not waste any CPU cycles. Polling, on the other hand, will check the state of the register on a regular basis to see if the event has happened. This would mean that polling would waste a lot of CPU cycles. Polling, on the other hand, would be more suited if the jobs on the I/O were short and frequent since it would not waste CPU cycles.
2. With polling, it is possible to use the DMA controller. With this we can check the change of status in a particular device since it would access every memory bit one by one.
3. a) If the process is not atomic, we would have inconsistencies in the results overall. For example, the registration for any membership. If context switching happens in the middle of the process, we might be registered with the wrong phone number and would then have inconsistencies.

b) To achieve atomicity, we could synchronise the methods. This would ensure that no inconsistencies would happen one process will only work if the other is produced.

**Answer 4:**

1. As system codes are stored in the memory area in kernel mode, I/O needs to be executed by the use of these system codes. We can switch to user mode to get access to these codes, however, we will then render them vulnerable to an attack, hence I/O can only be done in kernel mode.
2. a) We would have to account for the memory space used by malicious code in kernel mode when adding it. The code might be in kernel or user mode when we enter the interrupt. If it switches to kernel mode, it will be read-only, which means it won't be as dangerous because it won't be able to modify the memory directly. When in user mode, it can refer directly to kernel memory, allowing him to adjust the position of the interrupt and so allowing the code to run.

b) One solution would be to give the interrupt's placement priority and prevent the user from changing it. This would prevent the malicious code from causing harm to the CPU.

**Answer 5 :**

1. a)  If we let N be the number of processes. If we execute the process one less than the maximum (which in this case would be N – 1 times), we could achieve maximum time if it doesn’t increase the waiting time.
2. b)  If the waiting time of one process overlaps with another, this would make it that the total execution time would increase. As in a single programmed, only one process can run at a time, and since one process would need to end before the other would need to run, this increases the waiting time, which would hence increase the execution time. x

**Answer 6:**

1. Since you have no control over the time that you read, this would not need to be

privileged.

1. As you don’t want memory to be just cleared randomly, this would need to be privileged.
2. Since it is carried in user mode, this would not need to be privileged.
3. Once again, since this is also carried out in user mode, this would not need to be

privileged.

1. As you copy one register to another, there might be a chance that you enter in kernel mode, which would mean that there is a possibility for this instruction to be critically fatal, hence this would need to be privileged.
2. Switching off an interrupt would mean that the process would reach kernel mode, and this could use the whole CPU, hence this would need to be privileged.

vii. As this is a normal system call, this would not need to be privileged.

Hence, ii, v, and vi need to privileged.

**Answer 7:**

Network Operating System: - Offers communication to machines to a network

- Allow messages to be exchanged between these machines, this allows the whole network of machines to act as one single unit.

Distributed Operating System: -

Allows for different communication paths, which allow for connection to other different websites.

-Mostly used by computer users due to it’s ability to multitask

- Allow resources stored in a single hardware to be shared amongst participating users by shared memory.

Aside from the fact that they both use communication path to share data, there is nothing else common between these two.