A

Quiz1: CS3.301: Operating Systems and Networks, IIIT Hyderabad Time: 45 minutes; Max. Marks: 20			
Student's Roll No.	Date of Exam	Student's Signature	Invigilator's Signature

Note: Answer all questions. Make appropriate assumptions. Give a brief answer. There are 10 questions. Each question is for 2 marks.

1	**What is the objective of "time-sharing"? Why the same objective can not be realized through multiprogramming?** Ans: The objective of timesharing is to realize interactive computing. It can not be realized through multiprogramming as in multiprogramming, scheduler releases CPU whenever it encounters I/O operation. Sometimes, it may take long time to encounter I/O. As a result, other processes may have to wait for CPU for unpredictable duration. So, it is difficult to realize interactive computing through multiprogramming. In timesharing, CPU is interrupted after a fixed time period.
2	**In earlier operating systems, there was no mode bit, How the mode bit has enhanced the protection in the modern operating systems?** Ans: Sensitive tasks, such as setting of timer, should only be operated by operating system program. As there was no mode bit, it was difficult to have a control to restrict the execution of sensitive programs by the user in an efficient manner. The mode bit allowed to develop a mechanism so that OS can have a control over sensitive programs. By classifying the tasks into two groups: privileged and non-privileged, the mode bit allowed restriction of user programs in executing privileged tasks. The development of OS has become relatively simple after the emergence of mode bit.
3	**Explain the advantage of "DMA" data transfer over "interrupt initiated data transfer".** Ans: If we want to transfer a large data from DISK or any peripheral to main memory, we have to interrupt for every by byte with the interrupt initiated data transfer. The DMA mechanism allows the transfer of large chunk of data without frequent interrupts. As a result, the data transfer is fast.
4	**Explain how the multi-threading framework can exploit the power of multi-processing architectures as compared to the process framework?** Ans: The process framework means process with single thread. It can only use single CPU as it can only has a single stream of execution. Multithreading exploits multi-processor architectures. As multi-threading process has several streams of execution, each thread may be running parallel on a different processor. A single threaded process can only run on single CPU, no matter how many are available. [2]
	Give one example of "mechanism" and one example of "policy". What will happen if you design operating systems based on only policy decisions? Solution:

	The specification and design of OS is highly creative task. One important principle is separation of policy and mechanism. Mechanisms describe how things are done. Policies describe what will be done. Mechanisms do not change often; Policies may change. Examples of mechanism: Timer is used for CPU protection. Priority algorithm for scheduling. Example of policy: How long the timer set. Which process gets the priority. [1] If you implement operating system using only policies, the operating system has to be updated frequently as policies change often. So, it is better to leave the policy specifications to users [1]
6	**Explain the terms "waiting queue" and "Ready queue" in the Operating System. When the process enters and exits from waiting queue.** Ans: Ready queue: Contains the list of processes which are ready to run, If CUP is allocated. Waiting queue: Contains the list of processes which are waiting for some event to happen. They can not be executed, even CPU is available. Process enters waiting queue: When process interrupts during its execution, if Interrupt occurs for disk I/O or any other event which requires waiting, the process is shifted to waiting queue. Process exits waiting queue: When the event for which the process is waiting is over, the process moves from waiting to ready queue.
7	**Justify the requirement of medium-term scheduler** Ans: In computer system, if all the processes are I/O bound, the ready queue will be empty and CPU will be idle. If all the processes are CPU bound, the I/O queue will be empty, the devices will go unutilized and the system will be imbalanced. In this situation, medium-term scheduler is included by introducing swapping. The basic idea is that it can be advantageous, to forcibly remove the CPU bound process from the memory and introduce I/O bound process or remove I/O bound process and introduce CPU bound. Swapping means removal of process from main memory to disk to improve the performance. At some later time, the process can be reintroduced into main memory and its execution can be continued when it left off. Swapping improves the process mix (I/O and CPU), when main memory is unavailable.
8	**Explain how the notion of layering has helped in developing the operating system?** Ans: Earlier operating systems were monollthic operating systems. As OS constitutes a very bog program, there used be issues. Debugging and modification according to hardware change was a huge task. A small change used to result into modifications at multiple routines. Layering is a general philosophy that builds on the above approach. Decompose functionality into layers such that. Hardware is level 0, and layer t accesses functionality at layer (t-1) or less. Access via appropriately defined system calls. Information hiding was the main concept. The advantages of layered design: Resulted into a systematic design methodology called modular design with well defined interfaces between layers. It help in the prototyping/development: Association between function and layer eases overall OS design. As OS development and debugging can be done by layer by layer, it simplified debugging and system verification.

9 **Normally, OS has two privilege levels: user mode and system mode. The virtual hine OS like Xen has four privilege levels. Discuss the reasons.**

Solution:

Virtual machine like Xen has four levels. The reasons are as follows. Normally, two modes are implemented to protect OS from unintended/intended misuse by user programs. In case of virtual machine, we need four modes as we have to implement two layers of protection: One layer, we have to protect guest OS from the user programs of the guest OS. Second layer, we have to protect, virtual machine OS from multiple guest operating systems and corresponding application programs. [2].

Or

The detailed explanation is as follows.

(i) Virtual machine itself is an operating system on the top of hardware. On the top of virtual machine, multiple guest OS are supported. So, in principle, each guest OS is like an application of another OS called "virtual machine". So guest OS has both modes Guest OS mode, Virtual Machine mode.

(ii) Each guest OS has an application, which runs in a user mode of guest OS. So, the Application runs in four modes.
   (i) Guest user mode: When application runs in a user mode of guest OS
   (ii) Guest OS mode: When application executes system call of guest OS.
   (iii) Virual machine user mode: When guest OS is running the application
   (iv) Virtual machine system mode: When guest OS runs a system call of virtual machine OS for running an application.

10 **In the paper "My recollections of OS design", What are the problems with the probe instructions. How interrupts solved the problem and introduced new problems ?**

(i) Problem created by probe instructions: Processor time is wasted. Since execution of probe instruction takes time, if the processor probes the peripheral with high frequency, it slows down the processor/computation which probing with low frequency increases the reaction time. [1]

(ii) Solution with interrupts: Interrupts helped in saving processor time and using it for executing user programs. When the computer calculated at full speed, a piece of dedicated hardware monitors the outside world for completion signals from communication devices. When the completion (for example data transfer) is detected, the device interrupts the processor. [1]

(iii) New problems: As the timings of interrupts is unpredictable, they raised several synchronization and consistency issues which were resolved in due course.