1. In few line describe what is meant by a **timesharing** environment. Emphasize the task of the Operating System in such an environment.
2. What is the main difference between a **multiprocessing** and **multiprogramming** environment.
3. What is **batch file**? What is the default scheduling algorithm in **batch processing**?
4. What is an **interrupt vector**?
5. Time sharing is a way of sharing out computer facilities between a number of people who want to use the computer at the same time. Each has a separate terminal and gets the impression that they have sole use of the computer with their own 'account'.
6. Multiprocessing : several processors are used on a single computer system to increase the processing power of the machine.

Multiprogramming is a term used to describe how a computer can run more than one application at the same time by dividing up its memory.

(one CPU, multiple processes ready for execution.)

1. A batch file is a text file containing a series of commands intended to be executed by the command interpreter or sequence of commands.

Jobs with similar needs were batched together and were run as a group. The operating system's major task was to automatically transfer control from one job to the following one. A batch system is characterized by the lack of interaction between the user and the job during execution.

1. An "interrupt vector table" (IVT) is a data structure that associates a list of interrupt handlers with a list of interrupt requests in a table of interrupt vectors. Each entry of the interrupt vector table, called an interrupt vector, is the address of an interrupt handler.
2. What do you understand by a **Dual-Mode** processor? What would be the REASON for using a dual-mode processor?
3. Give an example of **synchronous interrupt** and **asynchronous interrupt** (at least 3 in total)
4. What do you understand by a Unix shell external command?
5. Describe the boot process. Mention/explain about POST and the user of MBR
6. Dual-mode is the distinction between execution of user mode and kernel mode (supervisor mode, system mode, and privileged mode). A mode bit is added to the hardware to indicate the current mode: Kernel(0) or user(1).

REASON : Protection of the OS and the user processes.

1. Synchronous : division by 0, underflow or overflow, interrupt, illegal reference, wait(), exit().

Asynchronous : from operating system, come from hardware, scheduler timer.

1. External commands are the commands that are executed by the kernel. These commands will have a process id running for it..

UNIX external commands will start a new shell and then it gets executed.

1. When turn on the computer, a set of instructions (that in the ROM) is executed, power-on self test (POST) check the hardware. BIOS seeks instruction on the master boot record. Bootstrap starts executing, loads in the main memory. the kernel of the OS.
2. Give an example of **TSR** processes.
3. Explain the difference between sequential execution and concurrent execution. Give the advantages of concurrent execution.
4. Give example of Unix shells. What is the command that will change the shell?
5. What type of kernel Android uses?
6. calendars, calculators, spell checkers, notepads
7. C shell, Z shell, TC shell

change shell : chsh

1. Android is based on the Linux kernel.
2. What do you understand by layered structure for an operating system? What will be advantages of having such structure?
3. Give the characteristics of win32 subsystem
4. Are real time systems the same as embedded systems? Why or why not?
5. If we have the hardware support, the OS can be broken into smaller pieces, creating a modular operating system. The OS is broken into a number of layers (levels), each built on top of lower layers. The lowest (0 layer) is the hardware. The highest (Nth layer) is the user interface. Each layer uses functions and services of lower-level layers only.

It is decomposable and therefore effects separation of concerns and different abstraction levels. It allows good maintenance, where you can make changes without affecting layer interfaces

1. interface between user mode and kernel. manages all keyboard, mouse, screen, I/O operation.
2. Maybe, but not all. An embedded system is a computer system designed for specific control functions within a larger system, “often” with real-time computing constraints.