1. **Differentiate between multitasking, multiprogramming and multiprocessing OS.**

**Ans –**

|  |  |  |  |
| --- | --- | --- | --- |
| S No. | Multiprogramming | Multiprocessing | Multitasking |
| 1 | The concurrent residency of more than one program in the main memory is called as multiprogramming. | The availability of more than one processor per system, which can execute several sets of instructions in parallel is called as multiprocessing. | The execution of more than one task simultaneously is called as multitasking. |
| 2 | One process is executed at a time. | More than one process can be executed at a time | One by one job is being executed at a time. |
| 3 | More time is taken to process the jobs. | Less time is taken for job processing. | Moderate amount of time is taken. |
| 4 | Efficiency is less | Efficiency is maximum | Efficiency is moderate |
| 5 | No further divisions | Symmetric & Asymmetric. | Single User & Multiuser. |

1. **What is the main advantage of time sharing?**

**Ans –**

In time sharing systems all the tasks are given specific time and task switching time is very less, so application do not get interrupted by it. Many applications can run at the same time.

Other advantages of Timesharing operating systems are −

* It provides the advantage of quick response.
* This type of operating system avoids duplication of software.
* It reduces CPU idle time.

1. **List the services provided by an OS.**

**Ans –**

Following are a few common services provided by an operating system −

* Program execution
* I/O operations
* File System manipulation
* Communication
* Error Detection
* Resource Allocation
* Protection

1. **How the OS ensure the non-interference between various processes stored in memory in case of multi-programming and time-sharing systems.**

**Ans –**

* The operating system keeps several jobs in memory at a time.
* This set of jobs is a subset of the jobs kept in the job pool.
* The operating system picks and begins to execute one of the jobs in the memory.
* Multiprogramming operating systems monitor the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle, unless there are no jobs to process.

1. **What is the purpose of system calls and how do system calls relate to OS and to the concept of dual mode (Kernel mode, user mode)?**

**Ans –**

System calls are used when a program wants access to some hardware or software which is controlled by the operating system. It is rare, for example, for an application program to access system hardware directly; instead, it calls interface routines which are part of the operating system. These are system calls. There are various reasons for this. Firstly, even though the hardware varies, the OS can “cover up” the differences, so that application programs don’t have to support a huge range of different hardware for the same purpose. Secondly, when you have multiple applications, all trying to use the same resources, the OS can control access to the resources, and resolve access conflicts.

The OS needs absolute privileges to access the hardware, memory and other resources. Using system calls, application programs request services from the OS, and so do not need the same absolute privileges; it is also a good idea not to give applications unlimited access, for reasons of security, and to ensure that if an application crashes it doesn’t take the OS with it. Hence dual mode operation - privileged mode for the OS itself, limited mode for applications.

1. **Differentiate between symmetric and asymmetric multiprocessing.**

**Ans –**

|  |  |
| --- | --- |
| Asymmetric Multiprocessing | Symmetric Multiprocessing |
| In asymmetric multiprocessing, the processors are not treated equally. | In symmetric multiprocessing, all the processors are treated equally. |
| Tasks of the operating system are done by master processor. | Tasks of the operating system are done individual processor |
| No Communication between Processors as they are controlled by the master processor. | All processors communicate with another processor by a shared memory. |
| In asymmetric multiprocessing, processes are master slave. | In symmetric multiprocessing, the process is taken from the ready queue. |
| Asymmetric multiprocessing systems are cheaper. | Symmetric multiprocessing systems are costlier. |
| Asymmetric multiprocessing systems are easier to design | Symmetric multiprocessing systems are complex to design |

1. **What do you mean by Real time operating system? Give an example of it.**

**Ans –**

Real-time operating system (RTOS) is an operating system intended to serve real time application that process data as it comes in, mostly without buffer delay. The full form of RTOS is Real time operating system.

In a RTOS, Processing time requirement are calculated in tenths of seconds increments of time. It is time-bound system that can be defined as fixed time constraints. In this type of system, processing must be done inside the specified constraints. Otherwise, the system will fail.

Real-time systems are used in:

* Airline reservation system.
* Air traffic control system.
* Systems that provide immediate updating.
* Used in any system that provides up to date and minute information on stock prices.
* Defence application systems like RADAR.
* Networked Multimedia Systems
* Command Control Systems
* Internet Telephony
* Anti-lock Brake Systems
* Heart Pacemaker

1. **Why are operating systems generally implemented using lower-level languages? For instance, Windows, MacOS, and Linux are all implemented in mostly C++, C, and some assembly. Explain why these types of languages are chosen over higher level ones such as Java or Python.**

**Ans –**

Operating systems are written in low level languages like assembly, C, C++.Because low languages are close to machine instructions and can be able to interact easily with the hardware in less time. For example: Unix operating system was totally written in C. Windows OS was written mostly in C++ Assembly.

Assembly language is often required for code that must do very specific hardware initialization where CPU instructions need to be executed in a precise fashion to initialize the system. C is favoured because it is sufficiently high-level to be understandable by humans, but sufficiently low-level to be able to do things like specify hardware addresses, pointers, etc.

1. **Different operating system structures offer both benefits and drawbacks over one another, and it’s important to understand what kind of structure should be used for different use-cases. Consider the following situation. You are tasked with creating a very secure piece of software for an embedded system that should have a relatively small amount of overhead. Given the nature of embedded systems, once you deploy the system you won’t be able to update the software, thus maintainability isn’t a large factor. What structure (simple, layered, microkernel, or modular) should you choose for its kernel? Explain.**

**Ans –**

We should use layered structure for its kernel, because layering makes it easier to enhance the operating system.