**Concept**

1. In order to provide application compatibility, system calls are rarely changed. Linux particularly enforces this (as opposed to in kernel APIs that can change as needed).
2. System call VS IPC
3. Kernel modular approach
4. inline functions

**System Programming** can be defined as the act of building Systems Software using System Programming Languages. According to Computer Hierarchy, one which comes at last is Hardware. Then it is Operating System, System Programs, and finally Application Programs. Program Development and Execution can be done conveniently in System Programs. Some of the System Programs are simply user interfaces, others are complex. It traditionally lays between the user interface and system calls.

| **BASIS FOR COMPARISON** | **KERNEL** | **OPERATING SYSTEM** |
| --- | --- | --- |
| Basic | Kernel is an important part of the operating system. | Operating System is a system program. |
| Interface | Kernel is an interface between software and hardware of the computer. | Operating System is an interface between user and hardware of the computer. |
| Type | Monolithic kernels and Microkernels. | Single and Multiprogramming batch system, Distributed operating system, Realtime operating system. |
| Purpose | Kernel memory management, process management, task management, disk management. | In addition to the responsibilities of Kernel, Operating System is responsible for protection and security of the computer. |

**Questions**

**Dictionary**

Advocate: - Push for something

Merit: - The quality that renders something desirable, valuable or useful