Comprehensive Notes on System Calls

1. Introduction to System Calls

1.1 Definition and Purpose

System calls are the programming interface to the services provided by the operating system (OS). They form the bridge between user programs and the OS kernel. System calls allow user-level programs to request services from the OS.

1.2 Implementation Languages

System calls are typically written in high-level languages like C or C++. For efficiency, lower-level parts may be written in assembly language.

1.3 Access Methods

Programs access system calls indirectly via Application Programming Interfaces (APIs), which provide a higher-level abstraction to system services.

1.4 Common APIs

1. Win32 API (Windows)
2. POSIX API (UNIX, Linux, macOS)
3. Java API (JVM)

2. System Call Interface

2.1 Structure

The system call interface acts as a bridge between user programs and the OS kernel, managing transitions between user and kernel mode.

2.2 Numbering System

System calls are associated with unique numbers, with the interface maintaining a table indexed by these numbers.

2.3 Invocation Process

1. User program invokes a system call (direct or API).

2. Interface identifies kernel routine.

3. Control passes to the kernel.

4. Kernel executes the request.

5. Control returns to user program.

2.4 Return Values

System calls typically return a status value: Negative for errors, positive or zero for success.

3. Types of System Calls

3.1 Process Control

System calls include process creation, termination, memory management, and debugging support.

3.2 File Management

Calls for file creation, deletion, reading, writing, and attributes management.

3.3 Device Management

Managing device I/O with calls for device requests, reads, writes, and detaching devices.

3.4 Information Maintenance

Time, system data, and attributes management calls.

3.5 Communications

Calls for message sending, receiving, and remote device management.

3.6 Protection

Access control system calls for resources and permissions management.

4. System Call Parameter Passing

Methods include passing parameters via registers, memory blocks, and stacks.