

SYSTEM CALLS

System calls serve as the fundamental interface between user applications and the operating system kernel. When a program needs to perform operations that require privileged access to hardware or system resources, it must make a request through these controlled gateways.

HOW THE PROCESS WORKS IN DETAIL:

1. User program prepares arguments for the system call
2. Program executes a special processor instruction (int 0x80 or syscall)
3. CPU switches from user mode to kernel mode
4. Kernel verifies the request is valid and safe
5. Kernel performs the operation using its privileged access
6. Results are packaged and returned to user space
7. CPU switches back to user mode
8. Program continues execution with the results

ADVANTAGES OF THIS DESIGN:

- Security: Prevents user programs from directly accessing hardware
- Stability: Kernel can validate all requests
- Consistency: Provides standardized interface across hardware
- Multitasking: Kernel can manage resource sharing.

GETDENTS()

The `getdents()` system call provides direct access to directory entries in the filesystem. It is more efficient than higher-level alternatives because it reads multiple directory entries in a single operation.

WHY USE GETDENTS DIRECTLY:

- Performance: Reduces context switches between user/kernel space
- Control: Provides raw access to directory data structures
- Learning: Demonstrates how filesystems actually work
- Flexibility: Allows custom directory processing

C Program Code to Implement `getdents()`

Here are Screenshots of the implementation of the code on the terminal

