



DEPARTMENT OF AIML

23CST202- OPERATING SYSTEMS II YEAR IV SEM

AIML-B



UNIT 5- I/O SYSTEM

TOPIC – KERNEL I/O SUBSYSTEM

The kernel provides many services related to I/O. Several services such as scheduling, caching, spooling, device reservation, and error handling - are provided by the kernel's I/O subsystem built on the hardware and device-driver infrastructure. The I/O subsystem is also responsible for protecting itself from errant processes and malicious users.

1. I/O Scheduling

To schedule a set of I/O requests means to determine a good order in which to execute them. The order in which the application issues the system call is the best choice. Scheduling can improve the overall performance of the system, can share device access permission fairly to all the processes, and reduce the average waiting time, response time, and turnaround time for I/O to complete.

OS developers implement schedules by maintaining a wait queue of the request for each device. When an application issues a blocking I/O system call, The request is placed in the queue for that device. The I/O scheduler rearranges the order to improve the efficiency of the system.

2. Buffering

A buffer is a memory area that stores data being transferred between two devices or between a device and an application. Buffering is done for three reasons.

1. The first is to cope with a speed mismatch between the producer and consumer of a data stream.
2. The second use of buffering is to provide adaptation for data that have different data-transfer sizes.
3. The third use of buffering is to support copy semantics for the application I/O, "copy semantic " means, suppose that an application wants to write data on a disk that is stored in its buffer. it calls the write() system's call, providing a pointer to the buffer and the integer specifying the number of bytes to write.

3. Caching

A cache is a region of fast memory that holds a copy of data. Access to the cached copy is much easier than the original file. For instance, the instruction of the currently running process is stored on the disk, cached in physical memory, and copied again in the CPU's secondary and primary cache.

The main difference between a buffer and a cache is that a buffer may hold only the existing copy of a data item, while a cache, by definition, holds a copy on faster storage of an item that resides elsewhere.

4. Spooling and Device Reservation

A spool is a buffer that holds the output of a device, such as a printer that cannot accept interleaved data streams. Although a printer can serve only one job at a time, several applications may wish to print their output concurrently, without having their output mixes together.

The OS solves this problem by preventing all output from continuing to the printer. The output of all applications is spooled in a separate disk file. When an application finishes printing then the spooling system queues the corresponding spool file for output to the printer.

5. Error Handling

An OS that uses protected memory can guard against many kinds of hardware and application errors so that a complete system failure is not the usual result of each minor mechanical glitch. Devices, and I/O transfers can fail in many ways, either for transient reasons, as when a network becomes overloaded or for permanent reasons, as when a disk controller becomes defective.

Error Handling Strategies : Ensuring robust error handling is a critical aspect of the Kernel I/O Subsystem to maintain the stability and reliability of the operating system. The strategies employed for error handling involve mechanisms for detecting, reporting, and recovering from I/O errors. Below are key components of error handling strategies within the Kernel I/O Subsystem:

1. **Error Detection Mechanisms:** The Kernel I/O Subsystem incorporates various mechanisms to detect I/O errors promptly.
2. **Error Reporting:** Once an error is detected, the Kernel I/O Subsystem employs mechanisms to report the error to higher levels of the operating system or user applications.
3. **Error Recovery Mechanisms:** Recovering from I/O errors is crucial to maintaining system stability.
4. **User Notification:** Informing users or administrators about I/O errors is essential for timely intervention and system maintenance:
 - User Alerts:** Providing alerts to users, either through the user interface or system notifications, can prompt immediate attention to potential issues.
 - Automated Notifications:** Implementing automated notification systems, such as emails or messages, to inform system administrators about critical errors for proactive system management.

6. I/O Protection

Errors and the issue of protection are closely related. A user process may attempt to issue illegal I/O instructions to disrupt the normal function of a system. We can use the various mechanisms to ensure that such disruption cannot take place in the system.

To prevent illegal I/O access, we define all I/O instructions to be privileged instructions. The user cannot issue I/O instruction directly.

The Kernel I/O Subsystem in Operating System

An Operating System (OS) is a complex software program that manages the hardware and software resources of a computer system. One of the critical components of an OS is the Kernel I/O Subsystem, which provides an interface between the operating system and input/output (I/O) devices. The Kernel I/O Subsystem manages the I/O requests made by the user applications and translates them into hardware commands that the devices can understand. In this article, we will discuss the importance of the Kernel I/O Subsystem and its advantages and disadvantages.

Importance of Kernel I/O Subsystem

The Kernel I/O Subsystem is an essential part of any modern Operating System. It provides a unified and consistent interface to the I/O devices, which enables the user applications to access them without knowing the details of the underlying hardware. The Kernel I/O Subsystem also manages the concurrency and synchronization issues that arise when multiple applications try to access the same device simultaneously.

Advantages of Kernel I/O Subsystem

- **Device Independence:** The Kernel I/O Subsystem provides device independence to the user applications. It abstracts the hardware details and provides a unified interface to the devices. This means that the application developers can write code that is independent of the hardware platform, and the Kernel I/O Subsystem takes care of the hardware-specific details.
- **Efficient Resource Management:** The Kernel I/O Subsystem provides efficient resource management for the I/O devices. It manages the I/O requests and schedules them in a way that optimizes the usage of the available resources. This ensures that the I/O devices are not overutilized, and the system remains responsive.
- **Concurrency Management:** The Kernel I/O Subsystem manages the concurrency issues that arise when multiple applications try to access the same device simultaneously. It ensures that the applications get exclusive access to the device when needed and allows multiple applications to share the device when appropriate.

Disadvantages of Kernel I/O Subsystem

- **Complex Implementation:** The Kernel I/O Subsystem is a complex software component that requires a lot of resources to implement and maintain. Any issues with the Kernel I/O Subsystem can affect the performance and stability of the entire system.
- **Security Risks:** The Kernel I/O Subsystem can pose security risks to the system if not implemented correctly. Attackers can exploit vulnerabilities in the Kernel I/O Subsystem to gain unauthorized access to the system or cause a denial-of-service attack.

Typical functions and services offered by the Kernel:

1. **Process management:** Save context of the interrupted program, dispatch a process, manipulate scheduling lists.
2. **Process communication:** Send and receive inter process messages.
3. **Memory management:** Set memory protection information, swap-in/ swap-out, handle page fault.
4. **I/O management:** Initiate I/O, process I/O completion interrupt, recover from I/O errors.
5. **File management:** Open a file, read/ write data.
6. **Security and protection:** Add authentication information for a new user, maintain information for file protection.
7. **Network management:** Send/ receive data through a message.