

THEME: OPERATING SYSTEMS: ARCHITECTURE, FUNCTIONS, SECURITY, AND MODERN DEVELOPMENT TRENDS

TOSHKENT DAVLAT IQTISODIYOT UNIVERSITETI
INTERNATIONAL BUSINESS AKADEMIK LITSEYI
O'QITUVCHISI MOXITDINOVA SURAYYO AKMALOVNA

moxitdinovasurayyo@gmail.com

+99899 984 27 19

Abstract. Operating systems coordinate computer hardware and software resources while providing a secure and efficient environment for applications. This article reviews operating-system architecture, kernel design, process scheduling, memory management, file systems, security mechanisms, virtualization, cloud-native operating systems, and AI-assisted resource management.

Keywords: operating system, kernel, process, memory management, scheduler, virtualization, cybersecurity, Linux, Windows.

1. Introduction

An operating system is the fundamental software layer that manages hardware resources and provides services for applications. It controls processors, memory, storage, input-output devices, networking and security while creating a consistent execution environment for users and programs.

2. Layered Architecture

Most modern operating systems are organized into layers. User applications interact with system libraries and APIs, which communicate with the kernel. The

kernel directly controls hardware resources and coordinates process execution, memory allocation, device drivers and interrupts.

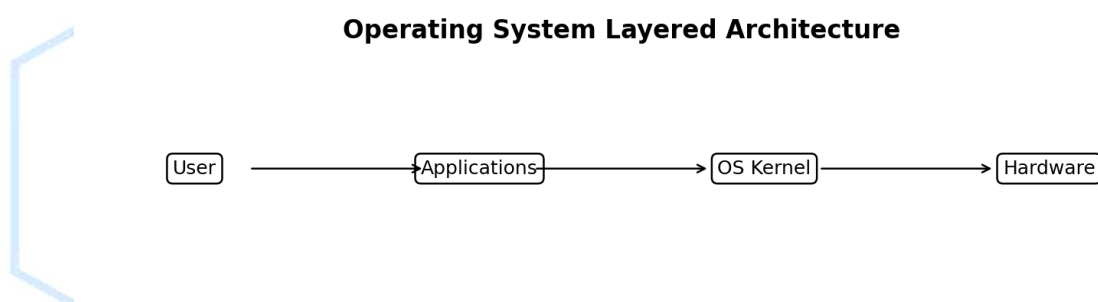


Figure 1. Layered architecture of an operating system.

3. Kernel Types

A monolithic kernel places most system services inside one privileged address space, offering high performance but greater complexity. Microkernels move many services into user space to improve modularity and fault isolation. Hybrid kernels combine elements of both approaches and are widely used in commercial operating systems.

4. Process Management

A process is an executing program with its own address space, registers and execution state. The scheduler selects which process receives CPU time, while context switching preserves and restores process state. Multitasking enables multiple applications to progress concurrently on one or more processors.

Process Life Cycle

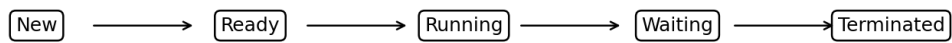


Figure 2. Simplified process life cycle.

5. Memory Management

Memory management allocates RAM efficiently and isolates applications from one another. Virtual memory allows processes to use logical addresses that are translated into physical memory. Paging, caching and memory protection improve both performance and reliability.

Memory Management

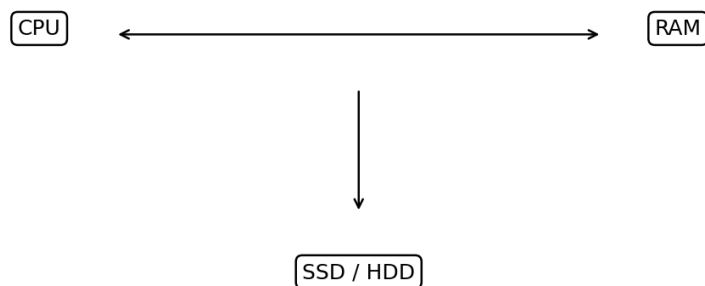


Figure 3. Memory management between CPU, RAM and storage.

6. File Systems and Storage

A file system organizes persistent data into files and directories while managing metadata, permissions and recovery mechanisms. Journaling file systems improve

reliability after unexpected power failures, whereas distributed file systems support cloud-scale storage across multiple servers.

7. CPU Scheduling Algorithms

Scheduling algorithms determine how processor time is distributed. Common strategies include First-Come First-Served, Shortest Job First, Priority Scheduling and Round Robin. Modern schedulers also consider multicore processors, energy efficiency and interactive responsiveness.

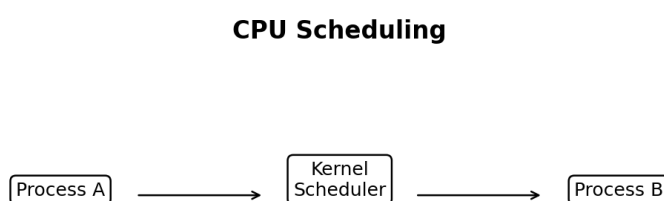


Figure 4. Kernel scheduling between processes.

8. Security and Protection

Operating systems enforce authentication, authorization, encryption, process isolation, secure boot and access control. Sandboxing prevents untrusted applications from accessing protected resources, while kernel updates and vulnerability management reduce security risks.

9. Virtualization and Cloud Operating Systems

Virtualization allows multiple virtual machines to share one physical computer through a hypervisor. Container technologies provide lightweight application isolation, making them essential for cloud-native infrastructure, DevOps and microservices.

10. Modern Trends

Current operating systems increasingly integrate artificial intelligence for workload prediction, adaptive scheduling, energy optimization and automated fault detection. Edge computing, IoT devices and quantum-ready security models are also influencing future operating-system research.

11. Conclusion

Operating systems remain the central software platform of digital computing. Their evolution toward intelligent, secure and cloud-integrated architectures enables efficient resource management across personal computers, mobile devices, servers and embedded systems.

Comparison of Major Operating Systems

OS	Kernel	Main Use	Key Strength
Windows	Hybrid	Desktop/Enterprise	Compatibility
Linux	Monolithic	Servers/Cloud	Open source & scalability
macOS	Hybrid	Personal computing	Integration & stability
Android	Linux-based	Mobile devices	Mobility & ecosystem

References

- [1] A. Silberschatz et al., Operating System Concepts, 10th ed., Wiley, 2018.
- [2] A. Tanenbaum and H. Bos, Modern Operating Systems, 5th ed., Pearson, 2022.
- [3] R. Love, Linux Kernel Development, 3rd ed., Addison-Wesley, 2010.

[4] M. Russinovich et al., Windows Internals, Microsoft Press, 2021.

[5] IEEE Computer Society, Operating Systems and Cloud Computing Standards.