

Operating Systems - Chapter 1 Summary

Main Themes of an Operating System

1. Virtualization – OS turns physical resources into easy-to-use virtual resources.
 - CPU → multiple virtual CPUs via scheduling
 - Memory → private address spaces per process
 - Devices → standardized I/O interfaces
2. Concurrency – Managing multiple tasks at once.
 - Issues: race conditions, synchronization, correctness
3. Persistence – Ensuring data survives crashes and shutdowns.
 - File system manages storage (open, write, close)
 - Techniques: journaling, copy-on-write

Design Goals of an Operating System

- Abstraction – hide hardware complexity
- Performance – minimize overhead of virtualization
- Protection & Isolation – prevent interference between processes
- Reliability – OS must not crash, or all apps crash
- Security – protect against malicious software
- Energy Efficiency – important in modern systems
- Mobility – adapt OS for smaller devices like smartphones

Brief History of Operating Systems

- Early OS – just libraries for I/O, batch processing
- System Calls & Protection – separation of user mode and kernel mode
- Multiprogramming – multiple jobs in memory, switching to maximize CPU use
- UNIX Era – simplicity, portability, open-source culture
- PC Era – initially lacked protection (e.g., DOS), later improved
- Modern OS – Linux, macOS, Windows NT, with strong protection and reliability

Summary

An OS virtualizes hardware resources, manages concurrency, and ensures data persistence. Its design goals are to provide abstraction, performance, protection, and reliability. Through historical evolution, OSes have grown into robust, reliable, and essential systems for modern computing.