Evolution of Operating Systems

The provided file is a lecture on the evolution of operating systems, prepared by Dr. Akshi Kumar from Goldsmiths, University of London. It traces the development of operating systems from early, simple systems to modern, complex ones.

The evolution of operating systems has been driven by the need to keep pace with improvements in hardware and to offer new services, such as internet support. This requires the design of operating systems to be modular, with clean interfaces and an object-oriented methodology.

Timeline of Operating System Evolution:

1. Early Systems (1950s): - Single-user systems where a programmer acted as the operator, running machines from a console using punched cards or paper tape. - Known as Serial Processing, inefficient due to slow I/O, long setup times, and very low CPU utilization. - Only one program executed at a time, with no user-computer interaction.

2. Simple Batch Systems (1960s): - Jobs were batched together by type, and managed by an operator. - The monitor managed job execution. - Job Control Language (JCL) introduced to specify compilers and input data. - Resident monitor reduced idle time between programs.

3. Multiprogrammed Batch Systems (1970s): - Multiple processes resided in memory simultaneously, CPU allocated to another program while one waited for I/O. - Increased CPU utilization. - Multitasking OS: Executed many programs by swapping in and out (e.g., Windows XP, Vista, 7). - Multiuser OS: Allowed multiple users on one system (e.g., Windows 2000, Ubuntu, Mac OS).

4. Time-Sharing and Real-Time Systems (1970s): - Extension of multiprogramming with several jobs in memory simultaneously, sharing CPU time. - Improved response time and CPU utilization. - Real-time systems: Hard real-time (failure if late), Soft real-time (reduced accuracy if late).

5. Personal/Desktop Computers (1980s): - Portable, single-user systems with keyboard, mouse, and screen. - Focused on user convenience and responsiveness.

6. Multiprocessor Systems (1980s): - Featured multiple CPUs in close communication. - Improved throughput, reliability, and cost-effectiveness. - Jobs divided into smaller tasks executed in parallel.

7. Networked/Distributed Systems (1980s): - Computation distributed across loosely coupled processors, no shared memory, communication via lines. - Benefits: resource sharing, computation speed, reliability. - Types: Client-Server (centralized server handling requests) and Peer-to-Peer (local memory, direct communication).