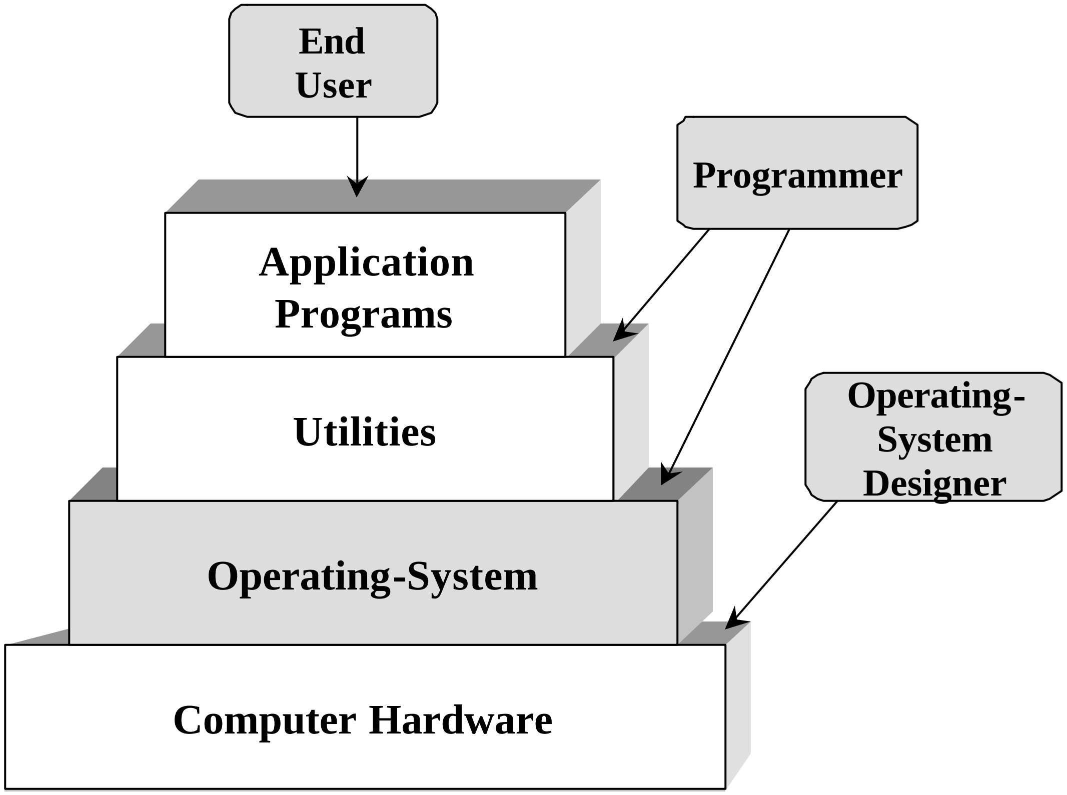
Operating Systems Overview

Chapter 2

Operating System

- A program that controls the execution of application programs
- An interface between the user and hardware
- Masks the details of the hardware

Layers and Views of a Computer System



Operating System Objectives

- Convenience
  - makes a computer more convenient to use
- Efficiency
  - allows the resources to be used efficiently
- Ability to evolve
  - should be constructed in such a way as to permit the effective development of new functions

Operating System Functions

- OS as a User/Computer Interface
- OS as Resource Manager
- Ease of Evolution of an OS

Services Provided by the Operating System

- Program creation
  - editors and debuggers
- Program execution
- Access to I/O devices
- Controlled access to files
- System access

Services Provided by the Operating System

- Error detection and response
  - internal and external hardware errors
    - memory error
    - device failure
  - software errors
    - arithmetic overflow
    - access forbidden memory locations

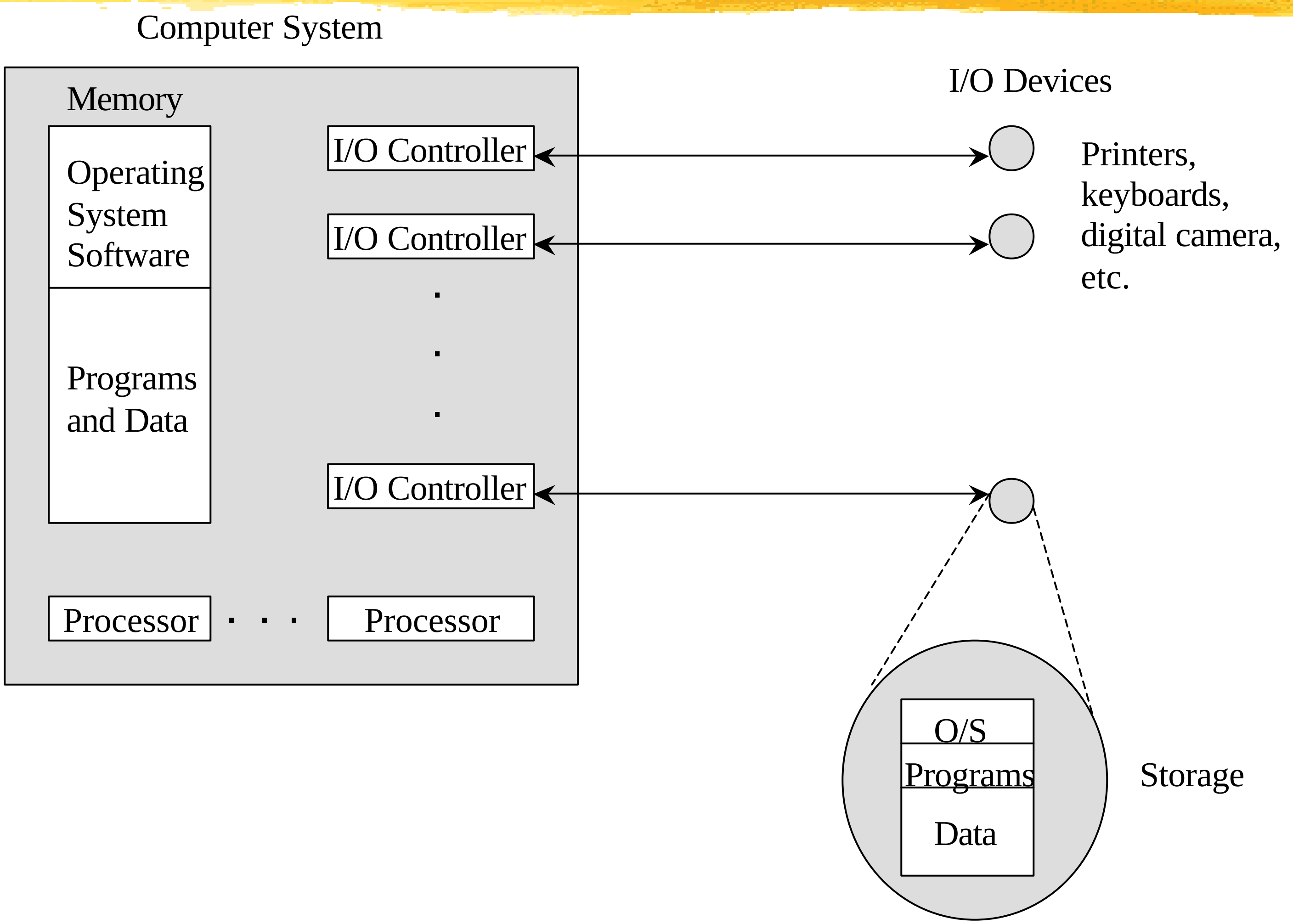
Services Provided by the Operating System

- Accounting
  - collect statistics
  - monitor performance
  - used to anticipate future enhancements
  - used for billing users

Operating System

- It is actually a program
- Directs the processor in the use of system resources
- Directs the processor when to execute other programs

Operating System as a Resource Manager



Ease of Evolution of an Operating System

- Hardware upgrades and new types of hardware
- New services
  - in response to user demand or in response to the needs of system managers
- Fixes
  - any OS has faults

Evolution of Operating Systems

- Serial Processing
- Simple Batch Systems
- Multiprogrammed Batch Systems
- Time-Sharing Systems

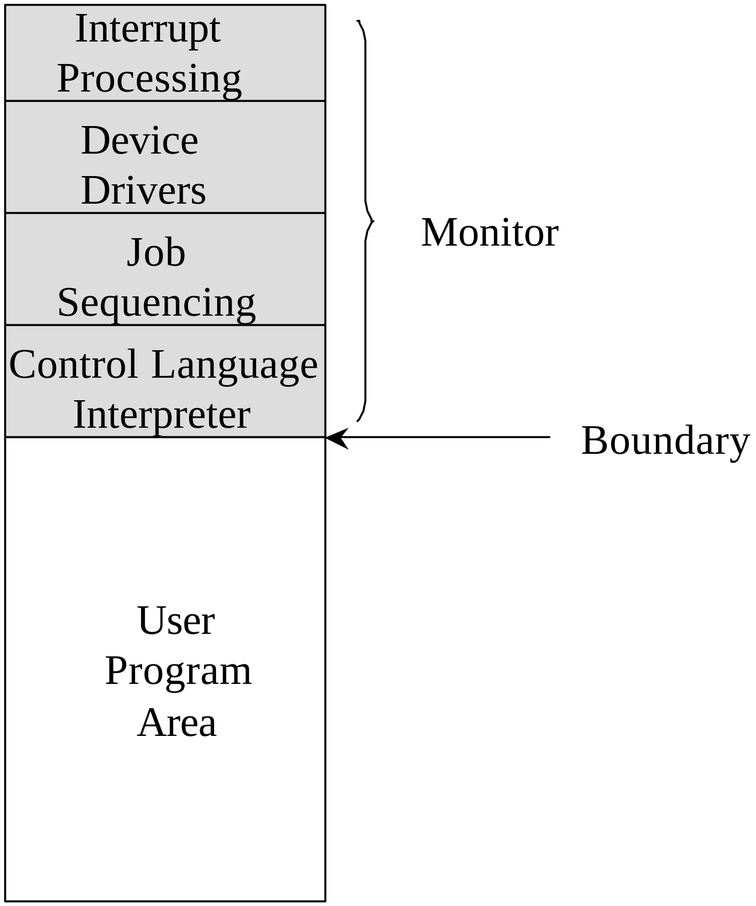
Serial Processing

- programmer interacted directly with the computer hardware - no OS
- problems
  - scheduling, setup time
  - machine is expensive and it is important to maximize machine use
  - wasted time caused by scheduling and setup time was unacceptable

Simple Batch Systems

- Monitor(early 1960s)
  - Batch operating system for IBM computers
    - Software that controls the running programs
  - Jobs are batched together
  - Resident monitor is in main memory and available for execution
  - Other Monitor utilities are loaded when needed

Memory Layout For a Resident Monitor



Simple Batch Systems

- Job Control Language (JCL)
  - Special type of programming language
  - Provides instruction to the monitor
    - what compiler to use
    - what data to use

Desirable Hardware Features

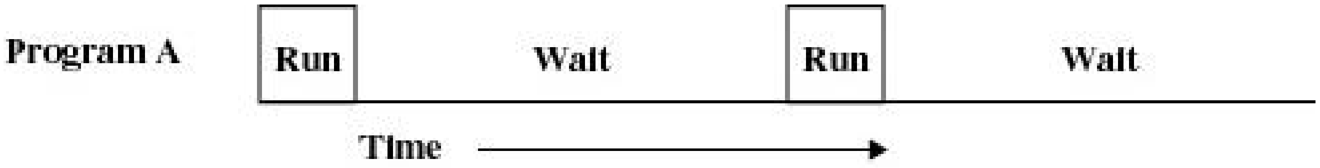
- Memory protection
  - do not allow the memory area containing the monitor to be altered
- Timer
  - prevents a job from monopolizing the system
  - an interrupt occurs when time expires

Desirable Hardware Features

- Privileged instructions
  - executed only by the monitor
  - an interrupt occurs if a user program tries these instructions
- Interrupts
  - provides flexibility for controlling user programs

Uniprogramming

- Processor must wait for I/O instruction to complete before proceeding



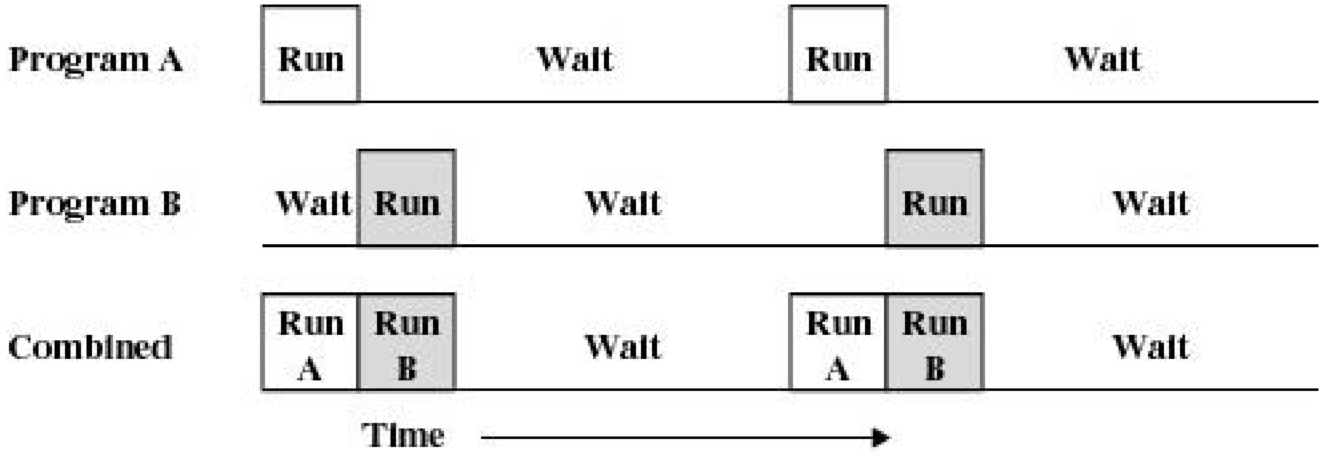
(a) Uniprogramming

Multiprogramming or Multitasking

- Central theme of modern OS
- Multiple programs in main memory at the same time
  - Need enough memory
  - When one program needs to wait for I/O, the processor can switch to the other program
- Needs additional H/W that supports I/O interrupts and DMA (independent I/O processor, I/O channel)

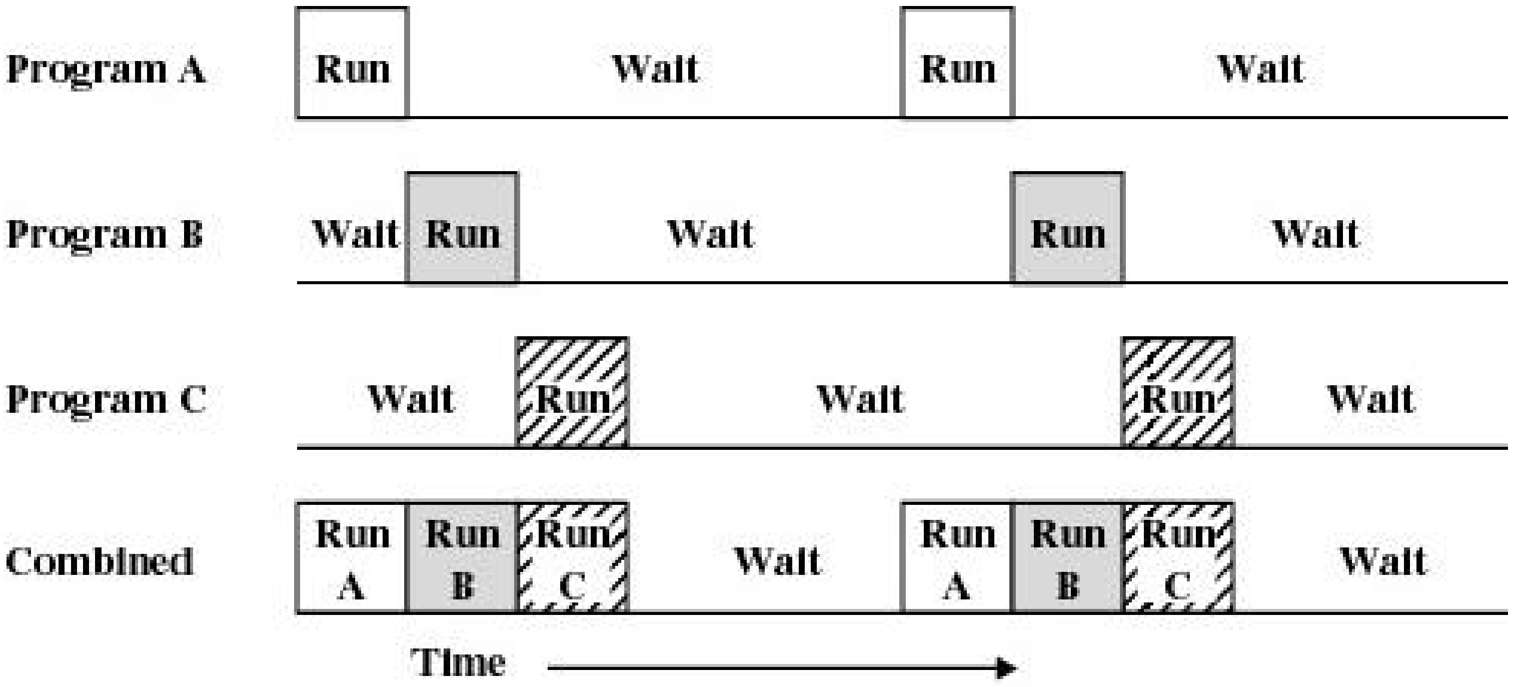
Multiprogramming

- When one job needs to wait for I/O, the processor can switch to the other job



(b) Multiprogramming with two programs

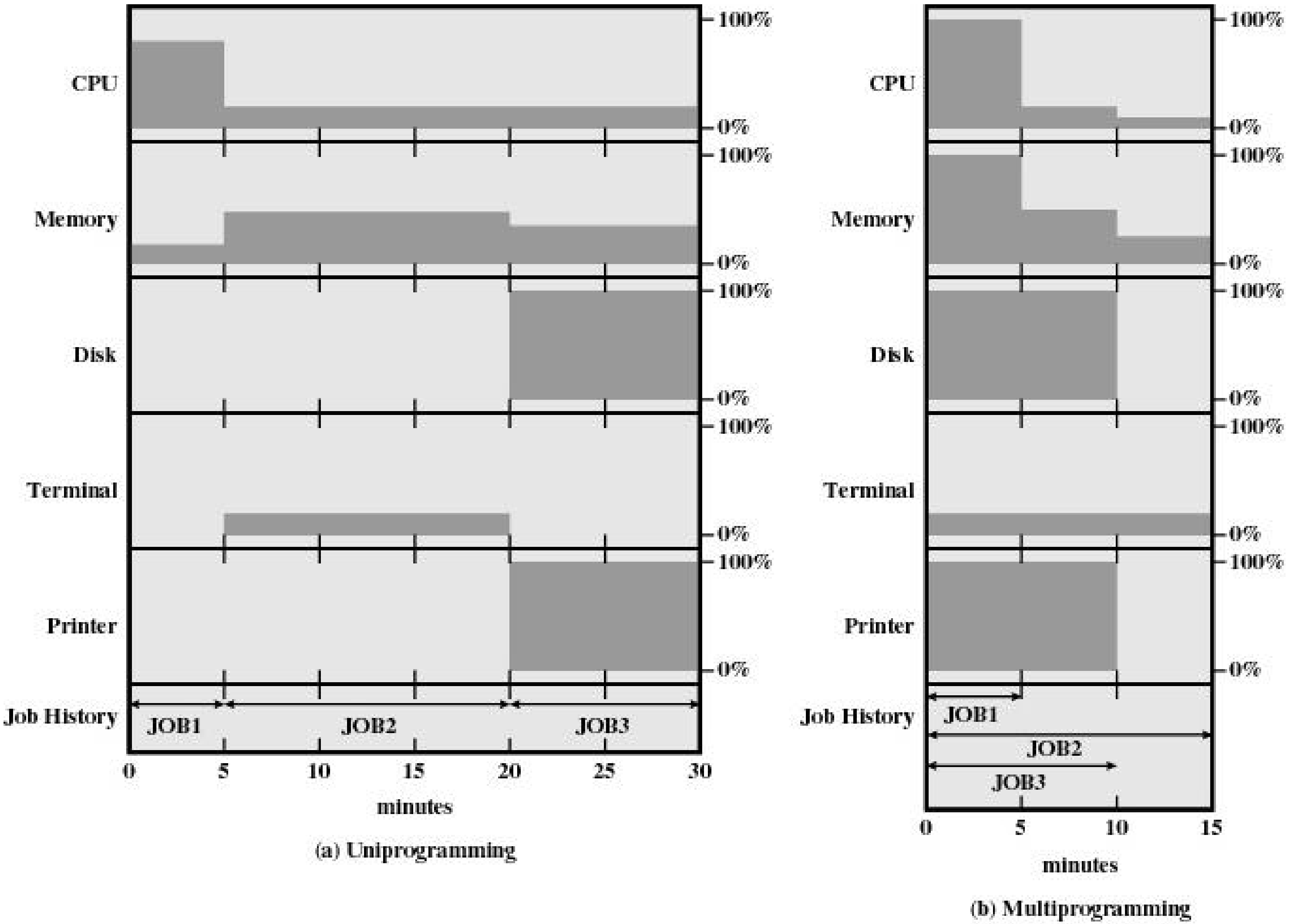
Multiprogramming



(c) Multiprogramming with three programs

Example

	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min.	15 min.	10 min.
Memory required	50K	100 K	80 K
Need disk?	No	No	Yes
Need terminal	No	Yes	No
Need printer?	No	No	Yes



(a) Uniprogramming

(b) Multiprogramming

Figure 2.6 Utilization Histograms

Effects of Multiprogramming

	Uniprogramming	Multiprogramming
Processor use	22%	43%
Memory use	30%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min.	15 min.
Throughput rate	6 jobs/hr	12 jobs/hr
Mean response time	18 min.	10 min.

Time-Sharing Systems

- Using multiprogramming to handle multiple interactive jobs
- Processor’s time is shared among multiple users
- Multiple users simultaneously access the system through terminals

Batch Multiprogramming versus Time Sharing

	Batch Multiprogramming	**Time Sharing**
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

Quiz 1(10 points)

- A process is trying to access a file. Explain how Unix handles this request. Use the following terms in your answer.
  - user ID, group ID
  - effective user ID, effective group ID
  - access permission to owner, group, others

Major Achievements

- Processes
- Memory Management
- Information Protection and Security
- Scheduling and Resource Management
- System Structure

Process

- More general term than a *job*
- Process
  - a program in execution
  - the “animated spirit” of a program
  - the entity that can be assigned to and executed on a processor
- Consists of an executable program, associated data, and execution context

Major Lines of Computer System Development

- Multiprogramming batch operation
  - designed to keep the processor and I/O devices simultaneously busy to achieve maximum efficiency
- Time sharing
  - designed to be responsive to as many users as possible
- Real-Time transaction system
  - users are entering queries or updates against a database

Difficulties with Designing System Software

- The design of the system software to coordinate the above activities turned out to be remarkably difficult
  - with many jobs in progress at any one time, each of which involved numerous steps to be performed in sequence, it became impossible to analyze all of the possible combinations of sequences of events

Main Causes of Errors

- Improper synchronization
  - ensure a process waiting for an I/O device receives the signal
- Failed mutual exclusion
- Nondeterminate program operation
  - when programs share memory, and their execution is interleaved by the processor, they may interfere with each other by overwriting common memory areas in unpredictable ways
- Deadlocks

Process

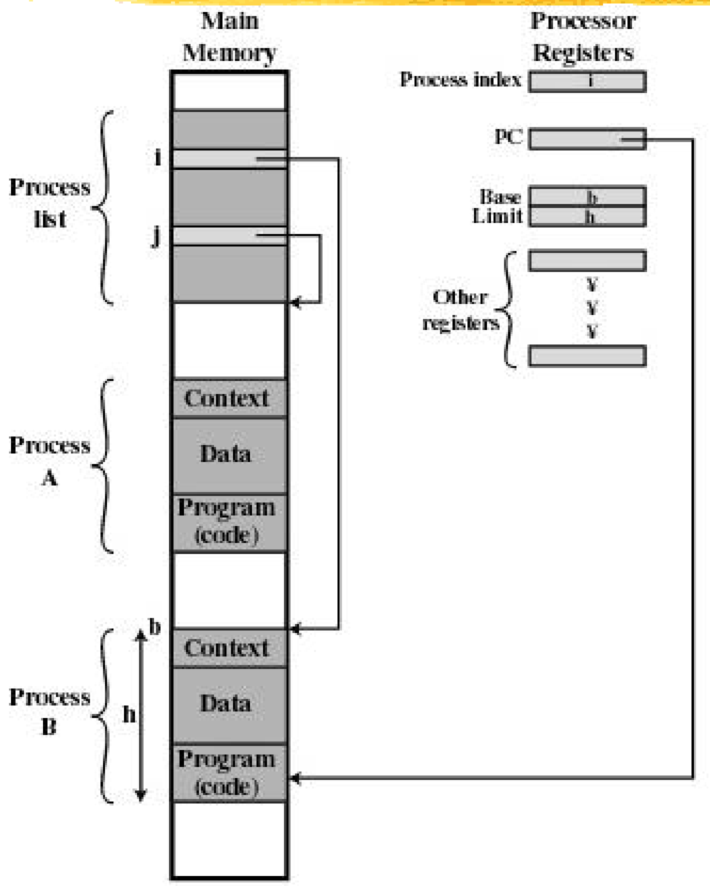


Figure 2.8 Typical Process Implementation

Memory Management

- Process isolation
  - independent processes should not interfere with each other
- Automatic allocation and management
  - allocation should be transparent to the programmer
- Support for modular programming

Memory Management

- Protection and access control
  - sharing of memory creates the potential for one program to address the memory space of another
  - at other times, it threatens the integrity of programs and even of the OS itself
- Long-term storage

File System

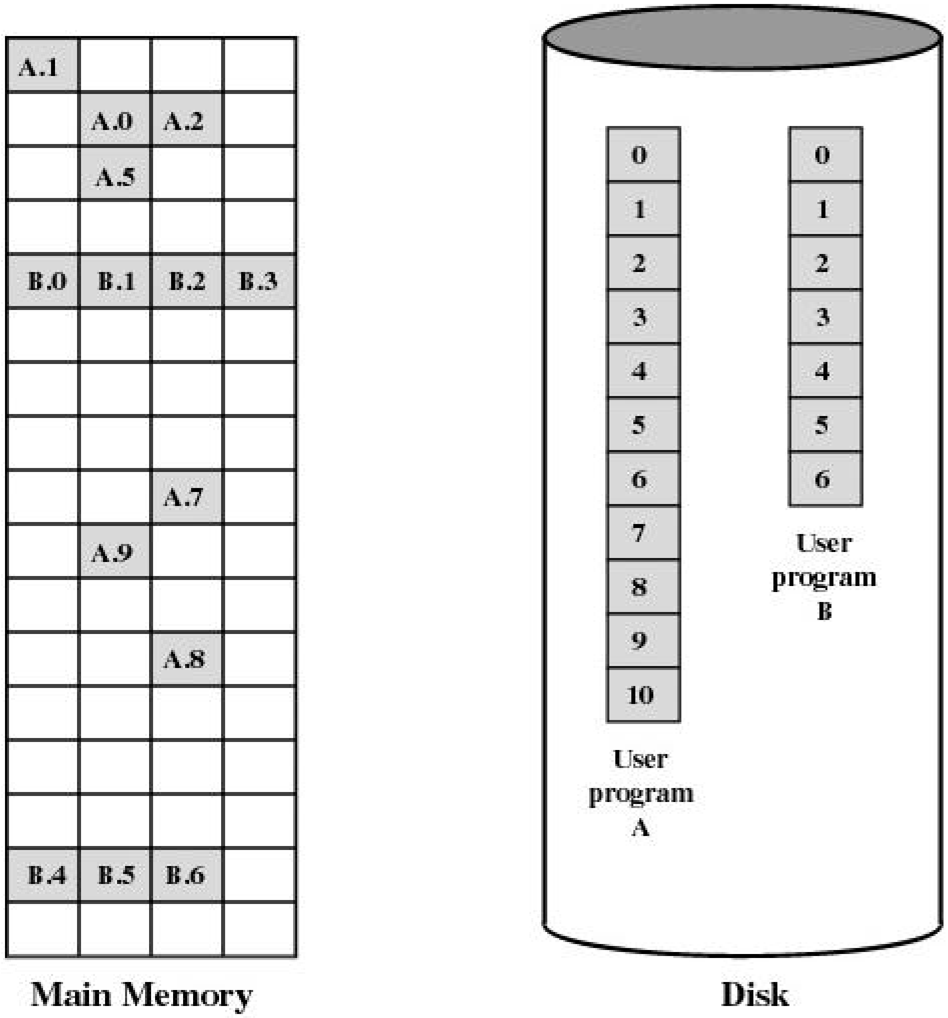
- Implements long-term store
- Information stored in files

Virtual Memory

- Allows programmers to address memory from a logical point of view
  - without regard to the amount of main memory physically available
- While a program is running, portions of the program and data are kept on disk
  - the size of a program can be bigger than that of whole main memory

Paging

- Allows process to be comprised of a number of fixed-size blocks, called pages
- Virtual address is a page number and an offset within the page
- Each page may be located any where in main memory
  - paging system provides for a dynamic mapping between virtual address and real address



Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.

Figure 2.9 Virtual Memory Concepts

Virtual Memory Addressing

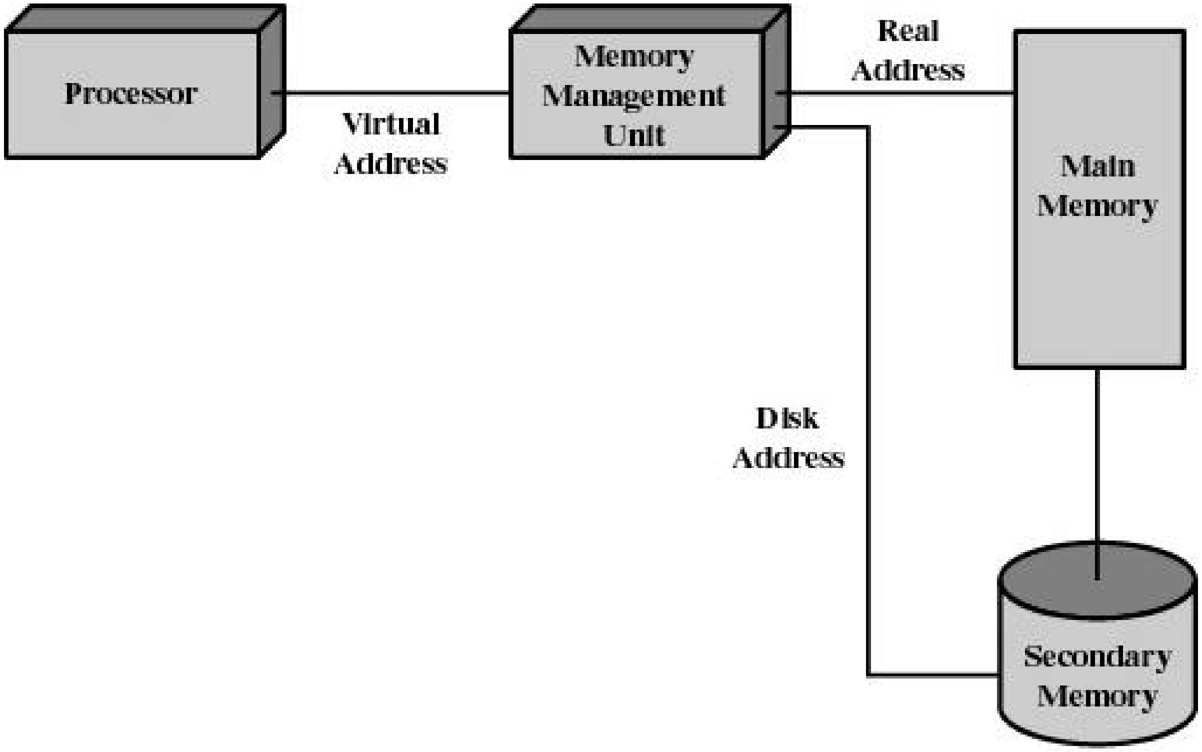


Figure 2.10 Virtual Memory Addressing

Information Protection and Security

- Access control
  - regulate user access to the system
- Information flow control
  - regulate flow of data within the system and its delivery to users
- Certification
  - proving that access and flow control perform according to specifications

Scheduling and Resource Management

- Fairness
  - give equal and fair access to all processes
- Differential responsiveness
  - discriminate between different classes of jobs
- Efficiency
  - maximize throughput, minimize response time, and accommodate as many users as possible

System Structure

- View the system as a series of levels
  - Each level performs a related subset of functions
  - Each level relies on the next lower level to perform more primitive functions
  - This decomposes a problem into a number of more manageable subproblems

Characteristics of Modern Operating Systems

- Microkernel architecture
- Multithreading
- Symmetric multiprocessing
- Distributed operating systems
- Object-oriented design

Characteristics of Modern Operating Systems

- Microkernel architecture
  - assigns only a few essential functions to the kernel
    - address space
    - interprocess communication (IPC)
    - basic scheduling

Characteristics of Modern Operating Systems

- Multithreading
  - process is divided into threads that can run simultaneously
  - Thread
    - dispatchable unit of work
    - executes sequentially and is interruptable
  - Process
    - a collection of one or more threads
    - owner unit of system resources

Characteristics of Modern Operating Systems

- Symmetric multiprocessing
  - there are multiple processors
  - these processors share same main memory and I/O facilities
  - All processors can perform the same functions(hence *symmetric*)

Characteristics of Modern Operating Systems

- Advantages over uniprocessor architecture
  - performance
    - works can be done in parallel
  - availability
    - failure of a processor does not halt the machine
  - incremental growth
    - can enhance performance by adding a processor
  - scaling
    - vendors can offer a range of products

Characteristics of Modern Operating Systems

- Distributed operating systems
  - provide the appearance of a single system for a cluster of separate computers
    - each with its own memory, and I/O modules
    - provides the illusion of a single main memory and a single secondary memory space

Characteristics of Modern Operating Systems

- Object-oriented design
  - used for adding modular extensions to a small kernel
  - enables programmers to customize an operating system without disrupting system integrity

Windows 2000

- Exploits the power of today’s 32-bit microprocessors
- Provides full multitasking in a single-user environment
- Client/Server computing

Windows 2000 Architecture

- Modular structure for flexibility
- Designed to execute on a variety of hardware platforms
- Supports applications written for a variety of other operating system

OS Organization

- Modified microkernel architecture
  - Not a pure microkernel
  - Many system functions outside of the microkernel run in kernel mode
- Any module can be removed, upgraded, or replaced without rewriting the entire system

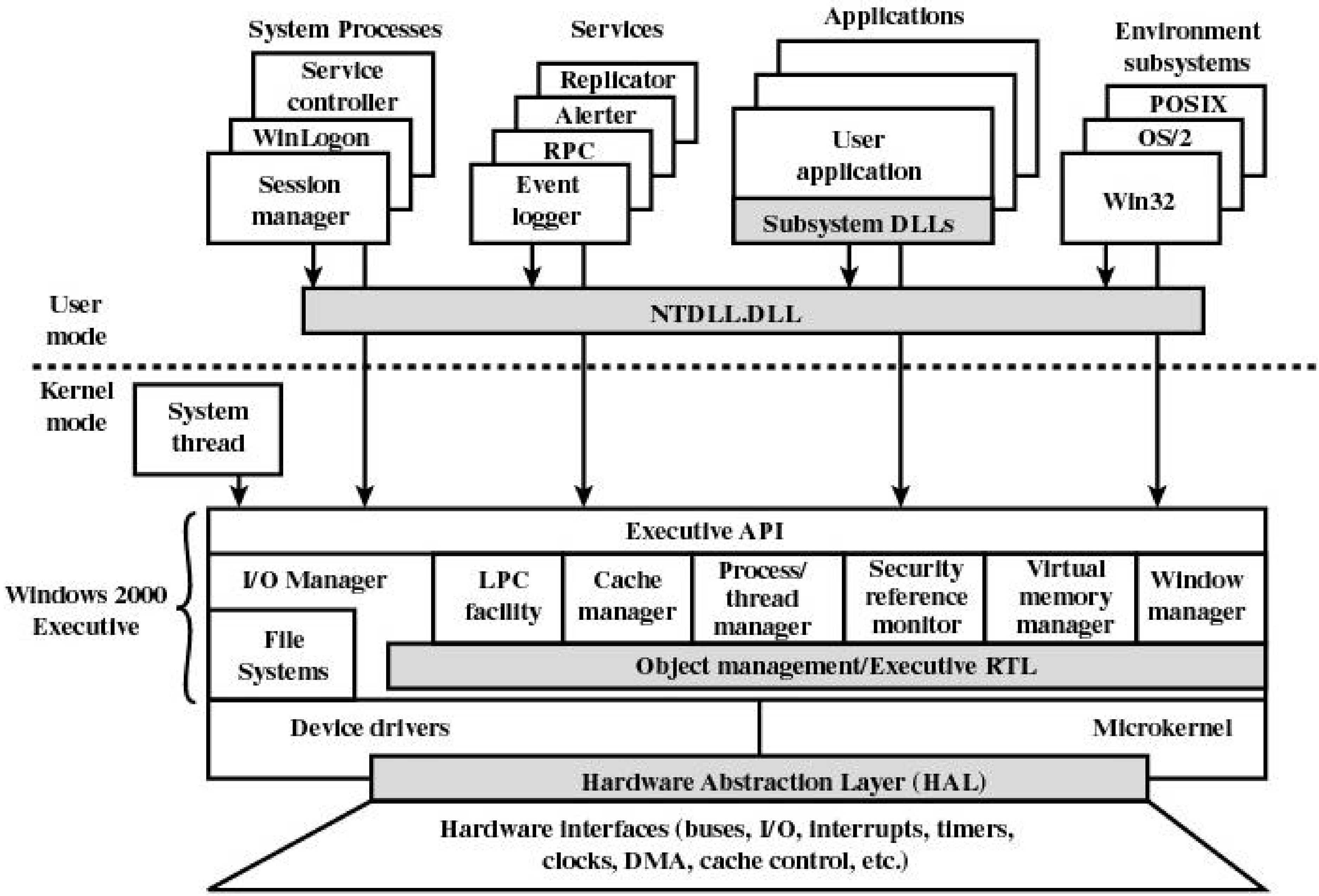


Figure 2.13 Windows 2000 Architecture

Layered Structure

- Hardware abstraction layer (HAL)
  - Isolates the operating system from platform-specific hardware differences
- Microkernel
  - Most-used and most fundamental components of the operating system
- Device drivers
  - Translate user I/O function calls into specific hardware device I/O requests

W2K Executive

- I/O manager
- Object manager
- Security reference monitor
- Process/thread manager
- Local procedure call (LPC) Facility
- Virtual memory manager
- Cache manager
- Windows/graphics modules

User Processes

- Special system support processes
  - Ex: logon process and the session manager
- Server processes
- Environment subsystems
- User applications

Client/Server Model

- Simplifies the Executive
  - possible to construct a variety of APIs
- Improves reliability
  - each service runs as a separate process with its own partition of memory
  - clients cannot not directly access hardware
- Provides a uniform means for applications to communicate via LPC
- Provides base for distributed computing

Threads and SMP

- Different routines can be executed simultaneously on different processors
- Multiple threads of execution within a single process may execute on different processors simultaneously
- Server processes may use multiple threads
- Mechanisms for sharing data and resources between processes

UNIX Architecture

- Hardware is surrounded by the operating-system
- Operating system is called the kernel
- Comes with a number of user services and interfaces
  - shell
  - C compiler

UNIX

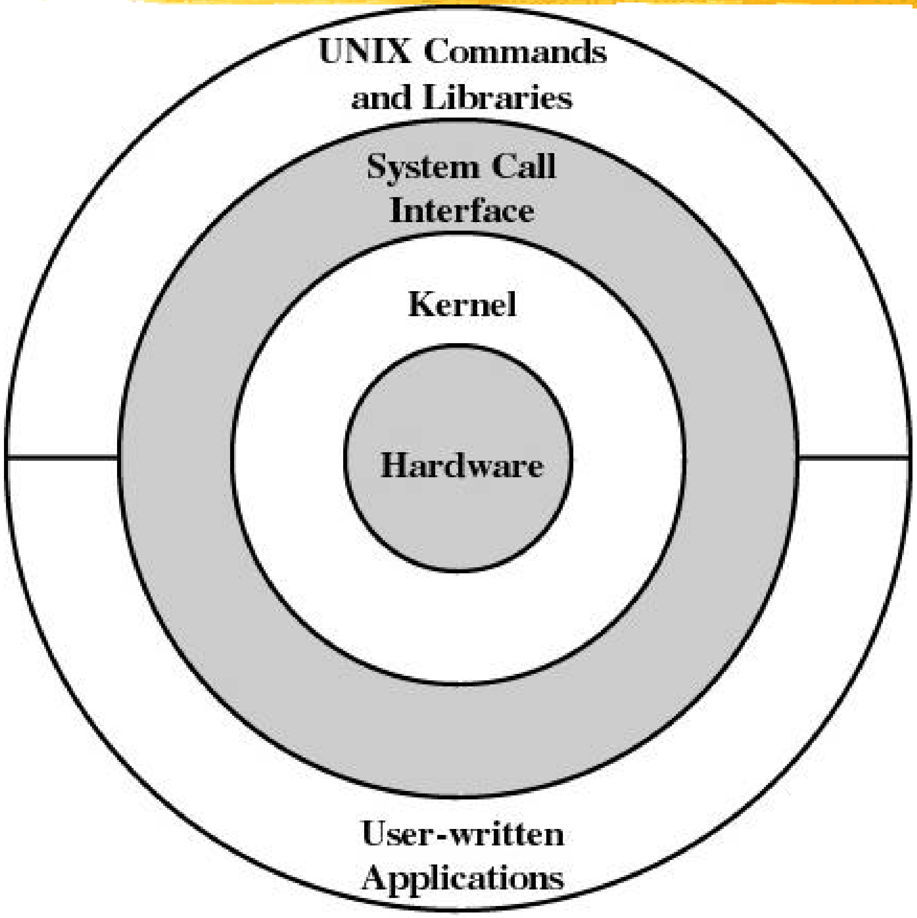


Figure 2.15 General UNIX Architecture

Modern UNIX Systems

- System V Release 4 (SVR4)
- Solaris 2.x
- 4.4BSD
- Linux