1. Introduction

1.1 The Role of Operating Systems
- Bridging the Hardware/Application Gap
- Three Views of Operating Systems

1.2 Organization of Operating Systems
- Structural Organization
- The Hardware Interface
- The Programming Interface
- The User Interface
- Runtime Organization

1.3 Operating System Evolution and Concepts

The role of OSs

- bridge hardware/application gap
  - machine instruction vs high level operation
    - compiler bridges gap
  - linear memory vs data structures
    - compiler bridges gap
  - secondary memory devices vs files
    - OS bridges gap
  - I/O devices vs high level I/O commands
    - OS bridges gap

Three views of OSs

- OS is an extended machine
  - principle of **abstraction** hides complexity
  - OS provides high level operations using lower level operations
- OS is a virtual machine
  - principle of **virtualization** support sharing
  - OS provides virtual CPU, memory, devices
- OS is a resource manager
  - balance overall performance with individual needs (response time, deadlines)

Organization of OSs

- structural organization
  - monolithic structure
  - layered structure

Figure 1-8

Organization of OSs

- hardware interface
  - applications and OS compiled into machine instructions
  - interrupts and traps allow OS to seize control
    - process management (time-sharing)
    - device management (I/O completion)

Figure 1-9

Organization of OSs

- hardware interface (cont)
  - modes of CPU execution
    - privileged/nonprivileged
    - SVC causes trap
    - control transferred to OS in privileged mode
    - OS turns off privileged mode when returning to user

Organization of OSs

- programming interface (Fig. 1-8)
- invoking system services
  - library call (nonprivileged)
  - kernel call (privileged)

Figure 1-10

Organization of OSs

- user interface (Fig. 1-8)
  - text-based shell (e.g. Unix)
    - command interpreter
    - shell scripts
  - graphics-based GUI (e.g. Windows)
    - icons
    - menus

Organization of OSs

- runtime organization
  - service is a subroutine
  - service is an autonomous process

Figure 1-13

OS Evolution and Concepts

- early systems
  - bootstrapping
- batch OSs
  - I/O processors
  - interrupts
  - relocatable code

OS Evolution and Concepts

- multiprogramming systems
  - overlap CPU and I/O
  - protection
  - synchronization and communication
  - dynamic memory management (swapping)
- interactive OSs
  - guarantee response time
  - time-sharing (quantum)

OS Evolution and Concepts

- PC and workstation OSs
  - GUI
- real-time OSs
  - deadlines (scheduling)
- distributed OSs
  - loosely coupled/tightly coupled
  - consistent timeline (logical clocks, time stamps)