資訊系統原理

郭大維教授
臺灣大學資訊工程系

Contents

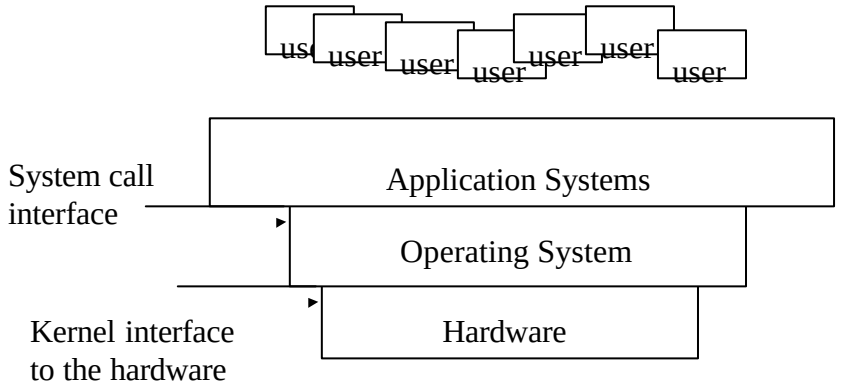
- Computer Systems Overview
  - Operating Systems Concept
  - UNIX
  - Other System Services
- Unified Modeling Language
  - UML Introduction
  - System Development Process
  - Use Cases, Class Diagrams, etc.

Operating Systems Concept

- What is an operating system?
- Operating system architecture
- Process concept
- CPU scheduling
- Memory management
- File and I/O systems
- Networking

What is an operating system?

- What is an operating system?
  - A package of software called OS!



* "Operating system concepts", Silberschatz and Galvin, Addison Wesley, pp. 3-5

What is an operating system?

- A control program
  - Control the execution of user programs
  - Prevent errors/misuse
- An environment for efficient/ convenient usage of a computer system.
- A resource allocator
  - CPU, memory space, file storage, I/O devices, shared code, data structures, etc.

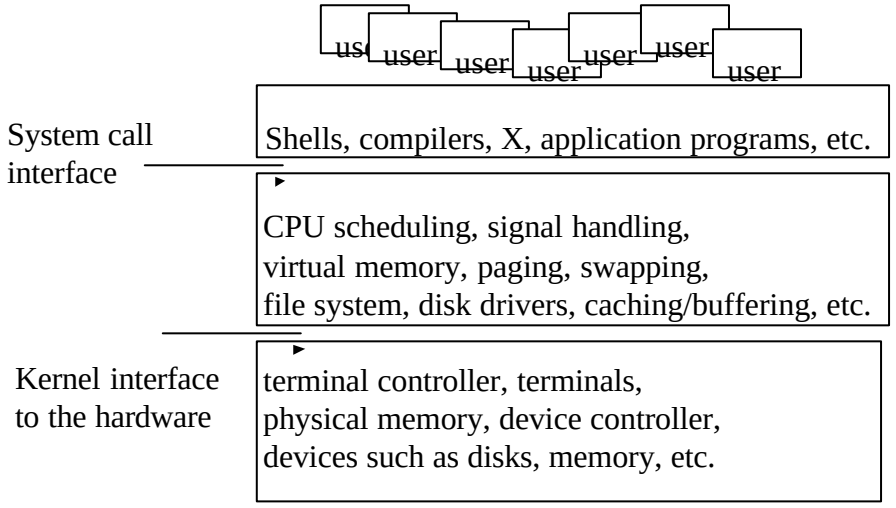
What is an operating system?

- Terminology
  - Multiprogramming
    - CPU/job scheduling – short/mid/long-term
  - Time-sharing
    - Multiprogramming + CPU switching ~ interactivities
  - Batch processing
    - Job pool – with/without multiprogramming
  - Spooling (Simultaneous Peripheral Operation On-Line)
    - Printf -> buffer (memory/disks) -> printout at a printer.
    - Card readers -> disks -> run process

Operating Systems Concept

- What is an operating system?
- Operating system architecture
- Process concept
- CPU scheduling
- Memory management
- File and I/O systems

OS Architecture



UNIX

OS Architecture

- What components/functionality it has?
  - Process Management
    - Creation/deletion/suspension/resumption of user/system processes
      - A process is a program in execution.
    - Process scheduling
    - Mechanisms for process synchronization
    - Interprocess communication mechanisms
  - Memory Management
    - Memory allocation/deallocation
    - Paging/segmentation memory management

* "Operating system concept", Silberschatz and Galvin, Addison Wesley, pp. 49-54

What is an operating system?

- File Management
  - Creation/deletion of files/directories
  - Mapping of files to secondary storage
- I/O Systems & Storage Management
  - Hide the peculiarity of specific H/W devices from users
  - Storage allocation and management
  - Disk scheduling
- Networking
  - Various networking service such as naming resolution
- Protection System
  - CPU, Memory, I/O devices

OS's on Parallel/Distributed Systems

- Parallel Systems
  - More than one processor in close communication, sharing of bus, clock, sometimes memory and peripheral devices – tightly coupled systems!
  - Symmetric/asymmetric operating systems.
- Distributed Systems
  - More than one processor without sharing of memory or any clock – loosely coupled systems!
  - Heterogeneous vs homogeneous systems!

* "Operating system concept", Silberschatz and Galvin, Addison Wesley, pp. 14-17

Real-Time OS

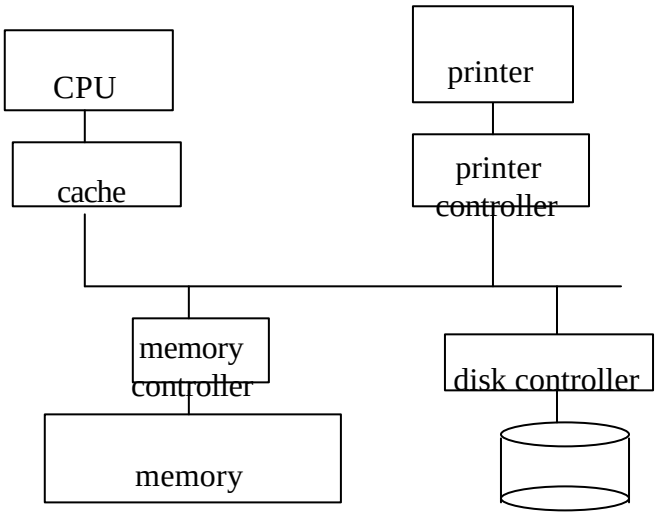
- Why RTOS
  - A convenient and reliable environment to develop time/safety-critical applications.
- Requirements – depending on applications!
  - Predictability – Verifiability & interrupt latency
  - Reliability - Strictness of Deadline Violations
  - Reconfigurability - System Size and Functionality
  - Efficiency of System Components - Time Granularity, Threads, and Resource Management
  - Variable Models of Task Communication - Characteristics of Applications

What is an operating system?

- Roadmap
  - Booting
  - I/O Structure
  - Storage Hierarchy
  - etc

* "Operating system concept", Silberschatz and Galvin, Addison Wesley, pp. 24-30, 35-37

Computer System Architecture

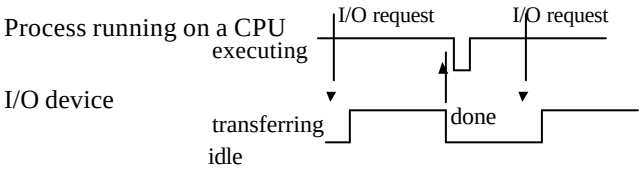


Booting

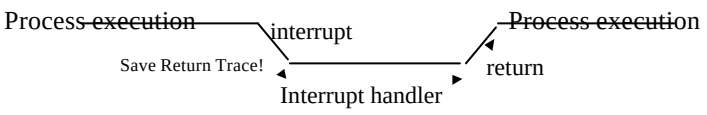
- Bootstrap program
  - Initialize all aspect of the systems
    - E.g., CPU registers, device controllers, memory, etc.
  - Load OS, and Run it!
    - Run init to initialize system services
      - Start virtual memory, various daemons, login processes, etc.

I/O Structure

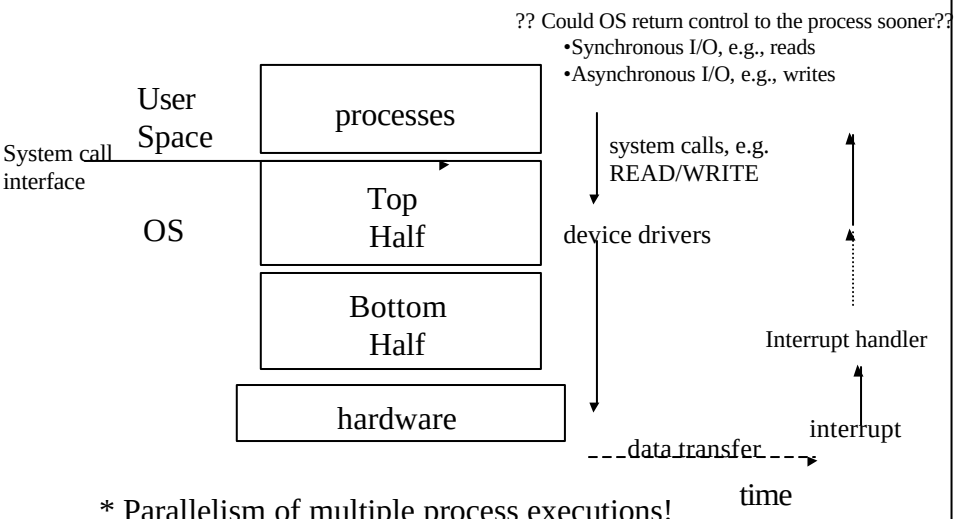
- Parallelism of CPU and I/O activities



- Interrupts

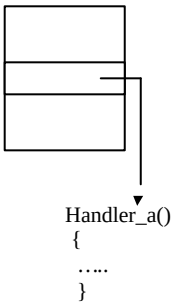


I/O Structure



* Parallelism of multiple process executions!

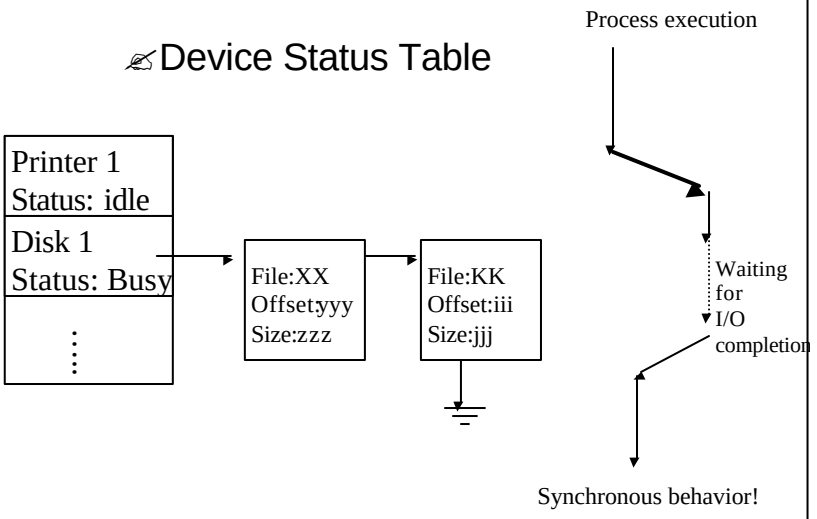
I/O Structure



- Interrupt types:
  - Software interrupts (traps)
    - Signals, division-by-zero, etc.
  - Hardware interrupts
    - Service requests of I/O devices, etc.
- Servicing of interrupts
  - Generic handler
  - Interrupt vector (UNIX)
- Masks, Disabling, Enabling
  - bits to mask interrupts

I/O Structure

- Device Status Table

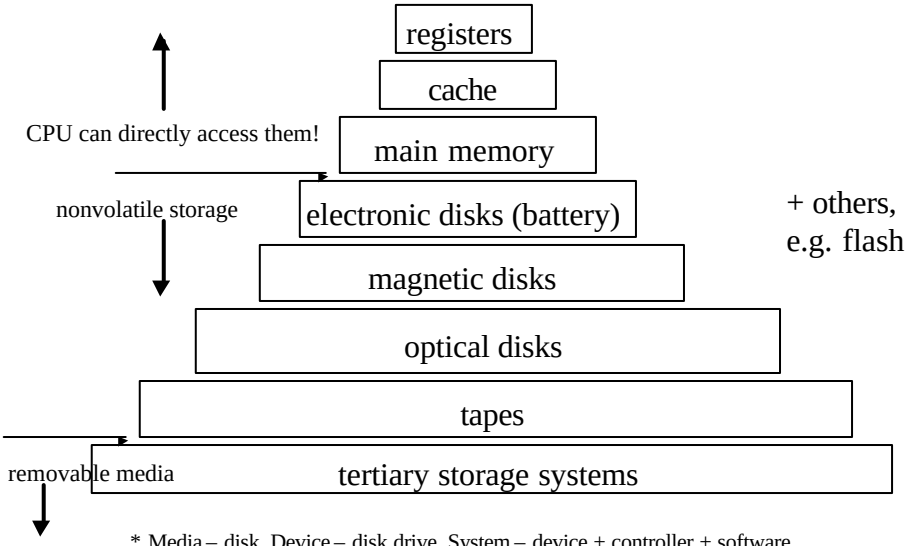


I/O Structure

- DMA (Direct Memory Access)
  - Release CPU from handling excessive interrupts
    - e.g., a high-speed device:
      - 2us service / 4us
  - Procedure, e.g., WRITE
    1. Use device driver to set up the registers of the DMA controller.
    2. DMA moves blocks of data between memory and its own buffers.
    3. Transfer from its buffers to its devices
    4. Interrupt the CPU

Until done

Storage Hierarchy



* Media – disk, Device – disk drive, System – device + controller + software
* "Operating system concept", Silberschatz and Galvin, Addison Wesley, pp. 24-30, 35-37

I/O and Storage Systems

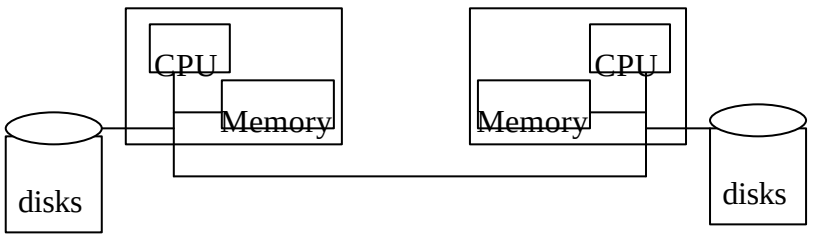
- Cost/bit vs Capacity/dollar vs volatility
- Device I/O
  - Memory-Mapped I/O
    - For devices with fast response time
  - Program I/O (PIO)
    - polling
  - Interrupt-Driven I/O

Storage Hierarchy

- Caching
  - When an information is used and might be used again, it is cached at a faster storage system.
  - Strategies at different levels?
- Buffering
  - When an information is “pushed” out to a slower storage system, it is buffered at a faster system for later actions.
  - Relationship with caching?

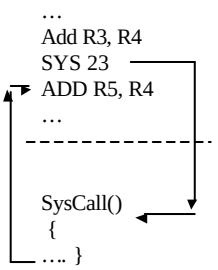
Storage Hierarchy

- Coherency and Consistency
  - Vertical information flow
  - “Horizontal” information flow



Dual Mode Protection

- What is “dual mode”?
  - User mode
  - Kernel mode
    - Privileged instructions, such as I/O, memory-setting related instructions
- Rationale
  - Only execute privileged instructions at the kernel mode to protect errors and misuse!
- Requirement
  - Hardware support

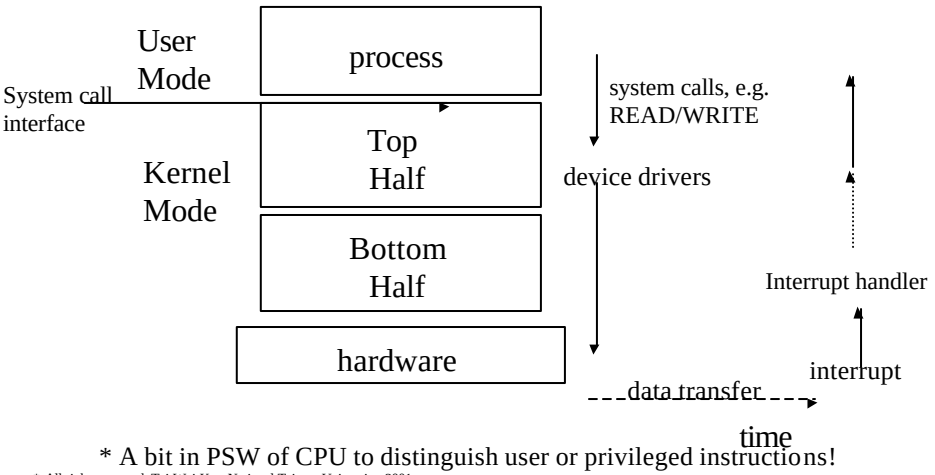


* “Operating system concept”, Silberschatz and Galvin, Addison Wesley, pp. 39

Booting - Revisiting

- Bootstrap program – Kernel and Single User Mode
  - Initialize all aspect of the systems
    - E.g., CPU registers, device controllers, memory, etc.
  - Load OS, and Run it! – Kernel and Multi-User Mode
    - Run init to initialize system services
      - Start virtual memory, various daemons, login processes, etc.
  - Shell Processes – User and Multi-User Mode

I/O Structure Revisiting



* A bit in PSW of CPU to distinguish user or privileged instructions!