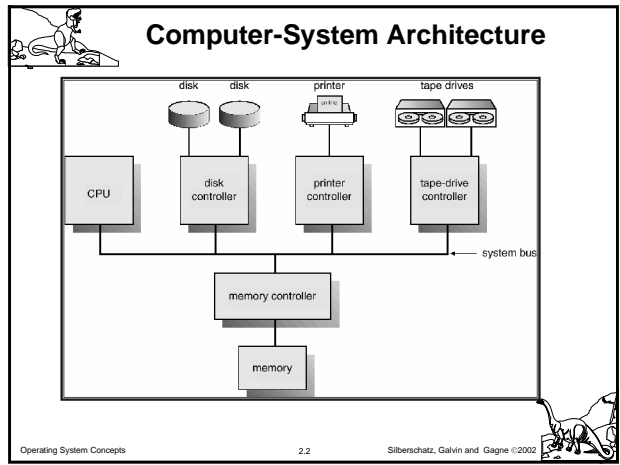


## Chapter 2: Computer-System Structures

- Computer System Operation
- I/O Structure
- Storage Structure
- Storage Hierarchy
- Hardware Protection
- General System Architecture

Operating System Concepts 2.1 Silberschatz, Galvin and Gagne ©2002



## Computer-System Operation

- I/O devices and the CPU can execute concurrently. Each device controller is in charge of a particular device type. Each device controller has a local buffer. CPU moves data from/to main memory to/from local buffers. I/O is from the device to local buffer of controller. Device controller informs CPU that it has finished its operation by causing an *interrupt*.

Operating System Concepts 2.3 Silberschatz, Galvin and Gagne ©2002

## Common Functions of Interrupts

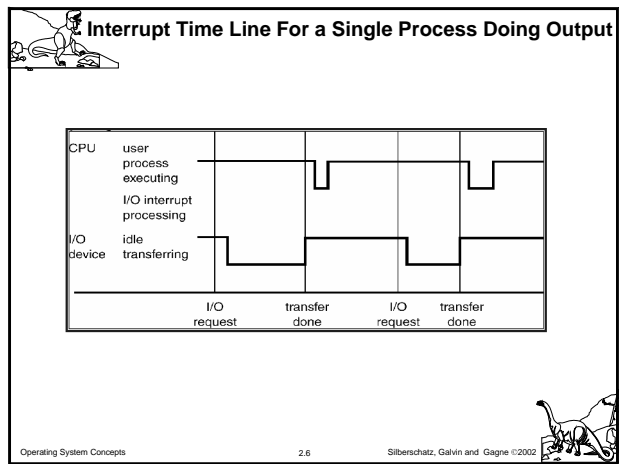
- Interrupt transfers control to the interrupt service routine generally, through the *interrupt vector*, which contains the addresses of all the service routines. Interrupt architecture must save the address of the interrupted instruction. Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*. A *trap* is a software-generated interrupt caused either by an error or a user request. An operating system is *interrupt driven*.

Operating System Concepts 2.4 Silberschatz, Galvin and Gagne ©2002

## Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter. Determines which type of interrupt has occurred:
  - polling*
  - vectored* interrupt system
 Separate segments of code determine what action should be taken for each type of interrupt

Operating System Concepts 2.5 Silberschatz, Galvin and Gagne ©2002



## I/O Structure

- After I/O starts, control returns to user program only upon I/O completion.
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access).
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing.
- After I/O starts, control returns to user program without waiting for I/O completion.
  - System call* request to the operating system to allow user to wait for I/O completion.
  - Device-status table* contains entry for each I/O device indicating its type, address, and state.
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt.

Operating System Concepts 2.7 Silberschatz, Galvin and Gagne ©2002

## Two I/O Methods

Synchronous

(a)

Asynchronous

(b)

Operating System Concepts 2.8 Silberschatz, Galvin and Gagne ©2002

## Device-Status Table

device: card reader 1 status: idle	
device: line printer 3 status: busy	request for line printer address: 38546 length: 1372
device: disk unit 1 status: idle	
device: disk unit 2 status: idle	
device: disk unit 3 status: busy	request for disk unit 3 file: xxx operation: read address: 43046 length: 20000
...	request for disk unit 3 file: yyy operation: write address: 03458 length: 500

Operating System Concepts 2.9 Silberschatz, Galvin and Gagne ©2002

## Direct Memory Access Structure

- Used for high-speed I/O devices able to transmit information at close to memory speeds. Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention.
- Only one interrupt is generated per block, rather than the one interrupt per byte.

Operating System Concepts 2.10 Silberschatz, Galvin and Gagne ©2002

## Storage Structure

- Main memory** only large storage media that the CPU can access directly.
- Secondary storage** extension of main memory that provides large nonvolatile storage capacity.
- Magnetic disks** rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into *tracks*, which are subdivided into *sectors*.
  - The *disk controller* determines the logical interaction between the device and the computer.

Operating System Concepts 2.11 Silberschatz, Galvin and Gagne ©2002

## Moving-Head Disk Mechanism

Operating System Concepts 2.12 Silberschatz, Galvin and Gagne ©2002

## Storage Hierarchy

- Storage systems organized in hierarchy.
  - Speed
  - Cost
  - Volatility

*Caching* copying information into faster storage system; main memory can be viewed as a last *cache* for secondary storage.

Operating System Concepts 2.13 Silberschatz, Galvin and Gagne ©2002

## Storage-Device Hierarchy

Operating System Concepts 2.14 Silberschatz, Galvin and Gagne ©2002

## Caching

- Use of high-speed memory to hold recently-accessed data.
  - Requires a *cache management* policy.
  - Caching introduces another level in storage hierarchy. This requires data that is simultaneously stored in more than one level to be *consistent*.

Operating System Concepts 2.15 Silberschatz, Galvin and Gagne ©2002

## Migration of A From Disk to Register

Operating System Concepts 2.16 Silberschatz, Galvin and Gagne ©2002

## Hardware Protection

- Dual-Mode Operation
  - I/O Protection
  - Memory Protection
  - CPU Protection

Operating System Concepts 2.17 Silberschatz, Galvin and Gagne ©2002

## Dual-Mode Operation

- Sharing system resources requires operating system to ensure that an incorrect program cannot cause other programs to execute incorrectly.
  - Provide hardware support to differentiate between at least two modes of operations.
    - User mode* execution done on behalf of a user.
    - Monitor mode* (also *kernel mode* or *system mode*) execution done on behalf of operating system.

Operating System Concepts 2.18 Silberschatz, Galvin and Gagne ©2002

### Dual-Mode Operation (Cont.)

- Mode bit added to computer hardware to indicate the current mode: monitor (0) or user (1). When an interrupt or fault occurs hardware switches to monitor mode.

Privileged instructions can be issued only in monitor mode.

Operating System Concepts 2.19 Silberschatz, Galvin and Gagne ©2002

### I/O Protection

- All I/O instructions are privileged instructions. Must ensure that a user program could never gain control of the computer in monitor mode (i.e., a user program that, as part of its execution, stores a new address in the interrupt vector).

Operating System Concepts 2.20 Silberschatz, Galvin and Gagne ©2002

### Use of A System Call to Perform I/O

Operating System Concepts 2.21 Silberschatz, Galvin and Gagne ©2002

### Memory Protection

- Must provide memory protection at least for the interrupt vector and the interrupt service routines. In order to have memory protection, add two registers that determine the range of legal addresses a program may access:

- Base register** holds the smallest legal physical memory address.
- Limit register** contains the size of the range. Memory outside the defined range is protected.

Operating System Concepts 2.22 Silberschatz, Galvin and Gagne ©2002

### Use of A Base and Limit Register

Operating System Concepts 2.23 Silberschatz, Galvin and Gagne ©2002

### Hardware Address Protection

Operating System Concepts 2.24 Silberschatz, Galvin and Gagne ©2002

## Hardware Protection

- When executing in monitor mode, the operating system has unrestricted access to both monitor and user's memory. The load instructions for the *base* and *limit* registers are privileged instructions.

Operating System Concepts 2.25 Silberschatz, Galvin and Gagne ©2002

## CPU Protection

- Timer* interrupts computer after specified period to ensure operating system maintains control. Timer is decremented every clock tick. When timer reaches the value 0, an interrupt occurs. Timer commonly used to implement time sharing. Time also used to compute the current time. Load-timer is a privileged instruction.

Operating System Concepts 2.26 Silberschatz, Galvin and Gagne ©2002

## Network Structure

- Local Area Networks (LAN)  
Wide Area Networks (WAN)

Operating System Concepts 2.27 Silberschatz, Galvin and Gagne ©2002

## Local Area Network Structure

Operating System Concepts 2.28 Silberschatz, Galvin and Gagne ©2002

## Wide Area Network Structure

Operating System Concepts 2.29 Silberschatz, Galvin and Gagne ©2002

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.