Operating Systems

Chapter 11 I/O Management and Disk Scheduling

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

Categories of I/O Devices(I/O设备分类)

- Roughly grouped into three categories:

1. Human readable(人可读)
   - Used to communicate with the user
   - Printers
   - Video display terminals
   - Input terminals
     - Keyboard
     - Mouse

Categories of I/O Devices

2. Machine readable(机器可读)

- Used to communicate with electronic equipment
- Disk and tape drives
- Sensors
- Controllers

Categories of I/O Devices

3. Communication(通信)

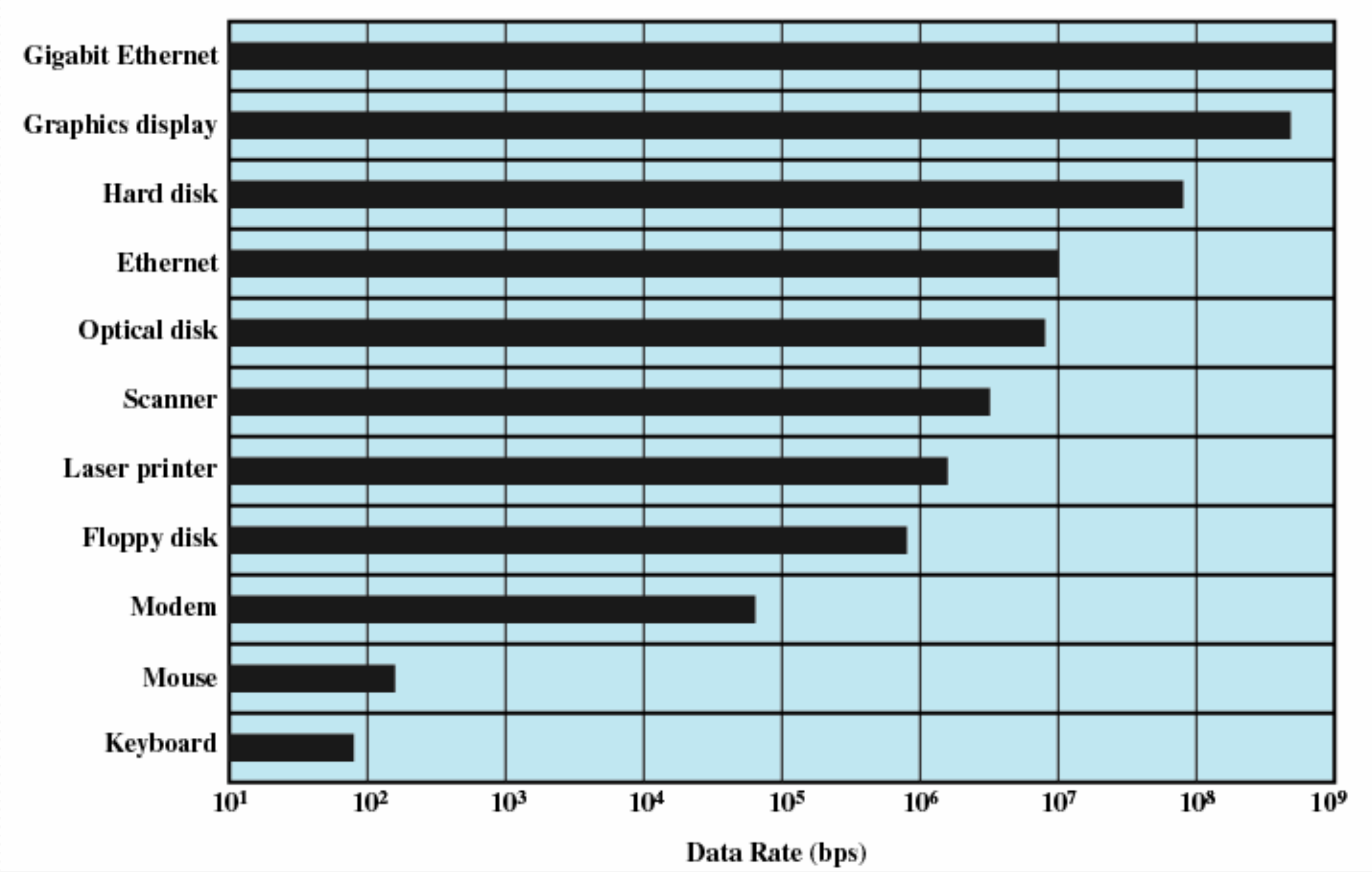
- Used to communicate with remote devices
- Digital line drivers
- Modems
- Network device

Differences in I/O Devices

1. Data rate

    - May be differences of several orders of magnitude(数量级) between the data transfer rates

Typical I/O Device Data Rates



Differences in I/O Devices

2. Application

- Disk used to store files requires file management software
- Disk used to store virtual memory pages needs special hardware and software to support it
- Terminal used by system administrator may have a higher priority

Differences in I/O Devices

3. Complexity of control
4. Unit of transfer（传送单位）
   - Data may be transferred as
     - a stream of bytes， e.g. terminal
     - in larger blocks, e.g. disk
5. Data representation
   - Encoding schemes
6. Error conditions
   - Devices respond to errors differently

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

Performing I/O

1. Programmed I/O(可编程I/O)
   - Process is busy-waiting for the operation to complete
2. Interrupt-driven I/O(中断驱动I/O)
   - I/O command is issued
   - Processor continues executing instructions
   - I/O module sends an interrupt when done
3. Direct Memory Access (DMA,直接存储器访问)
   - DMA module controls exchange of data between main memory and the I/O device
   - Processor interrupted only after entire block has been transferred

Relationship Among Techniques

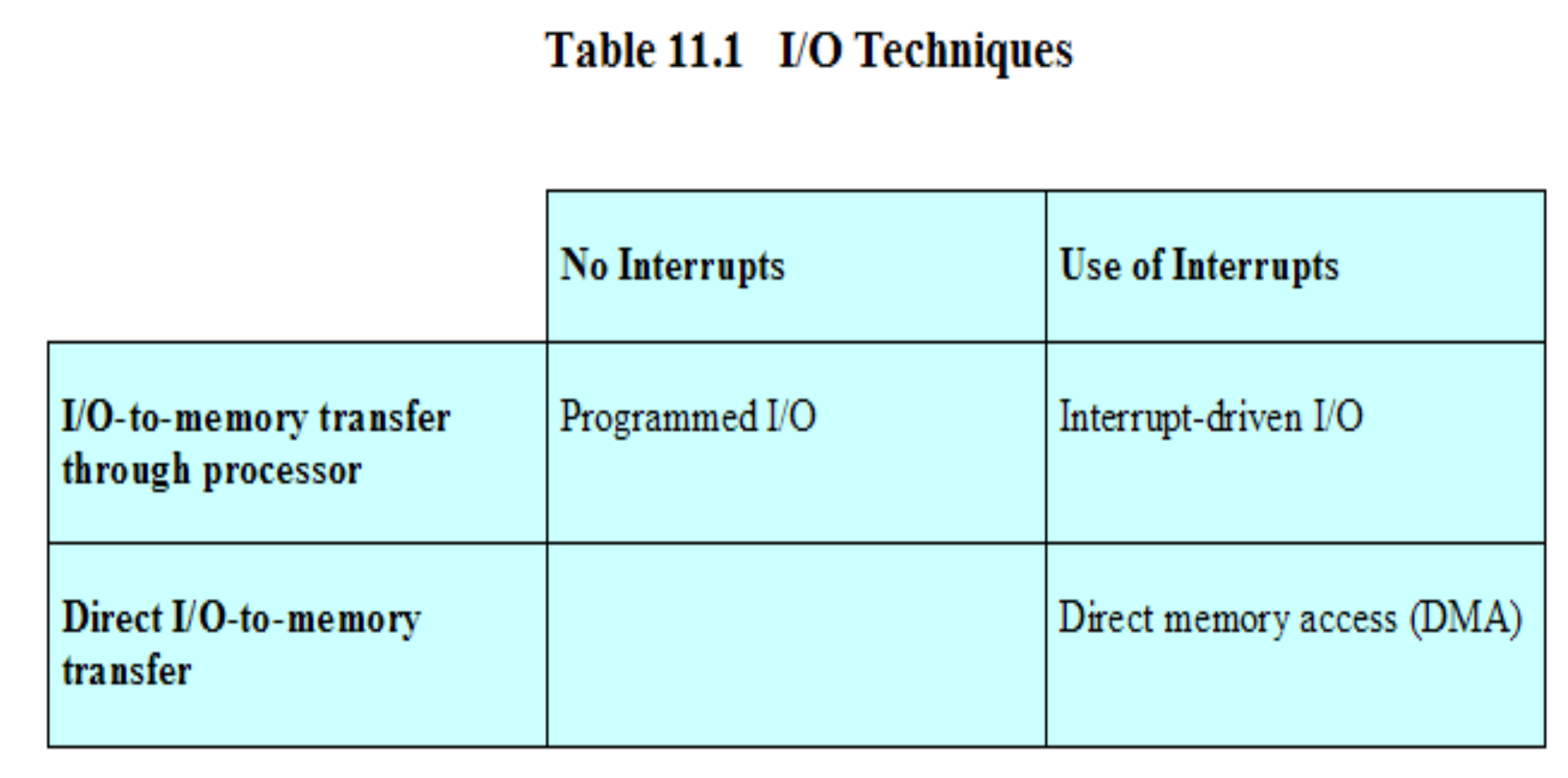


Table 11.1 I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Evolution of the I/O Function (I/O功能的发展)

1. Processor directly controls a peripheral device
   - Processor has to handle details of external devices
2. Controller or I/O module is added
   - Processor uses programmed I/O without interrupts
   - Processor does not need to handle details of external devices

Evolution of the I/O Function

3. Controller or I/O module with interrupts
   - Processor does not spend time waiting for an I/O operation to be performed
4. Direct Memory Access
   - Blocks of data are moved into memory without involving（牵涉） the processor
   - Processor involved at beginning and end only

Evolution of the I/O Function

5. I/O module (I/O channel) is enhanced to a separate processor
   - The central processing unit (CPU) directs the I/O processor to execute an I/O program in main memory.
   - The I/O processor fetches and executes these instructions without processor intervention.
6. I/O processor
   - I/O module has its own local memory
   - Its a computer in its own right

Direct Memory Access(直接存储器访问)

1. Processor delegates(委派) I/O operation to the DMA module
2. DMA module transfers data directly to or form memory
3. When complete DMA module sends an interrupt signal to the processor

DMA Block diagram

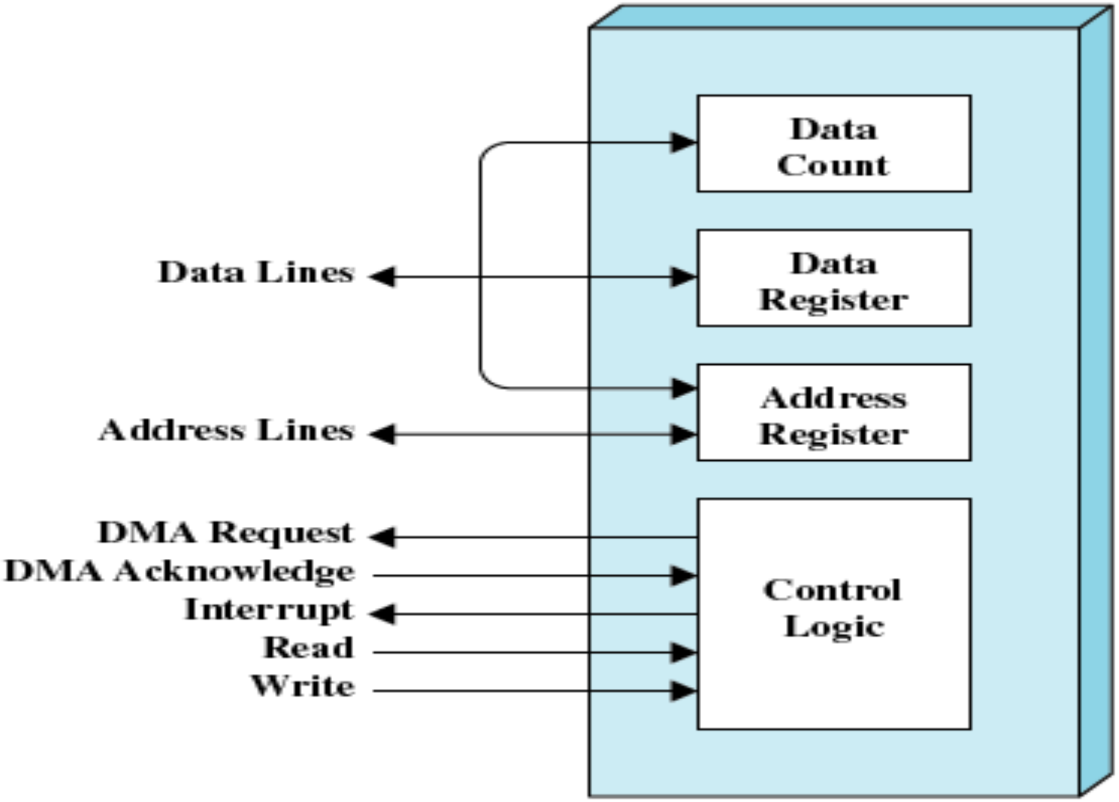
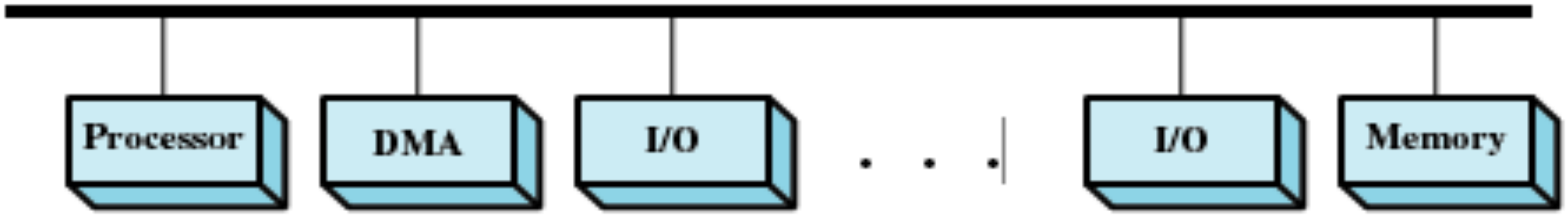
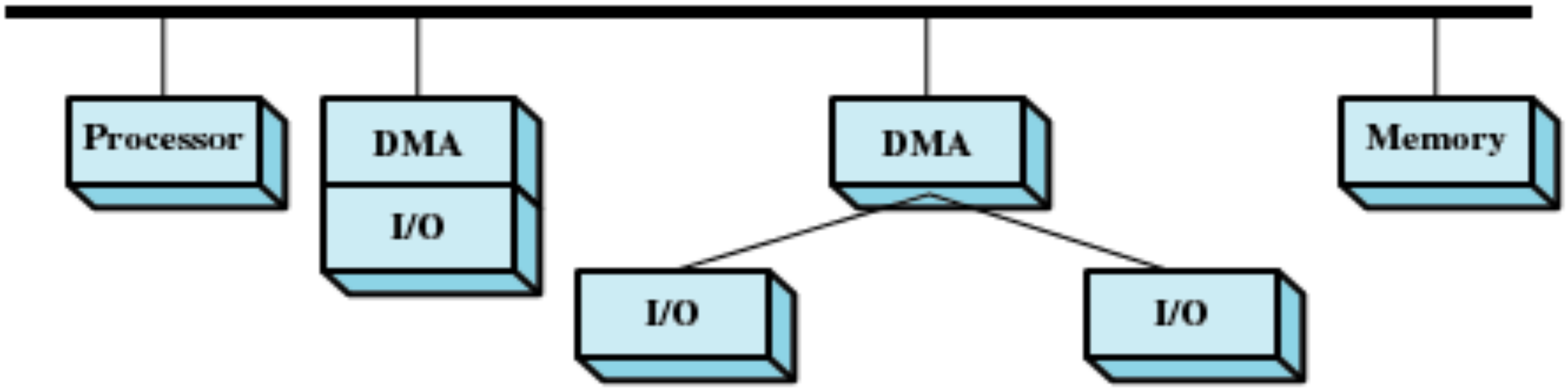


Figure 11.2 Typical DMA Block Diagram

DMA Configurations



(a) Single-bus, detached DMA



(b) Single-bus, Integrated DMA-I/O

DMA Configurations

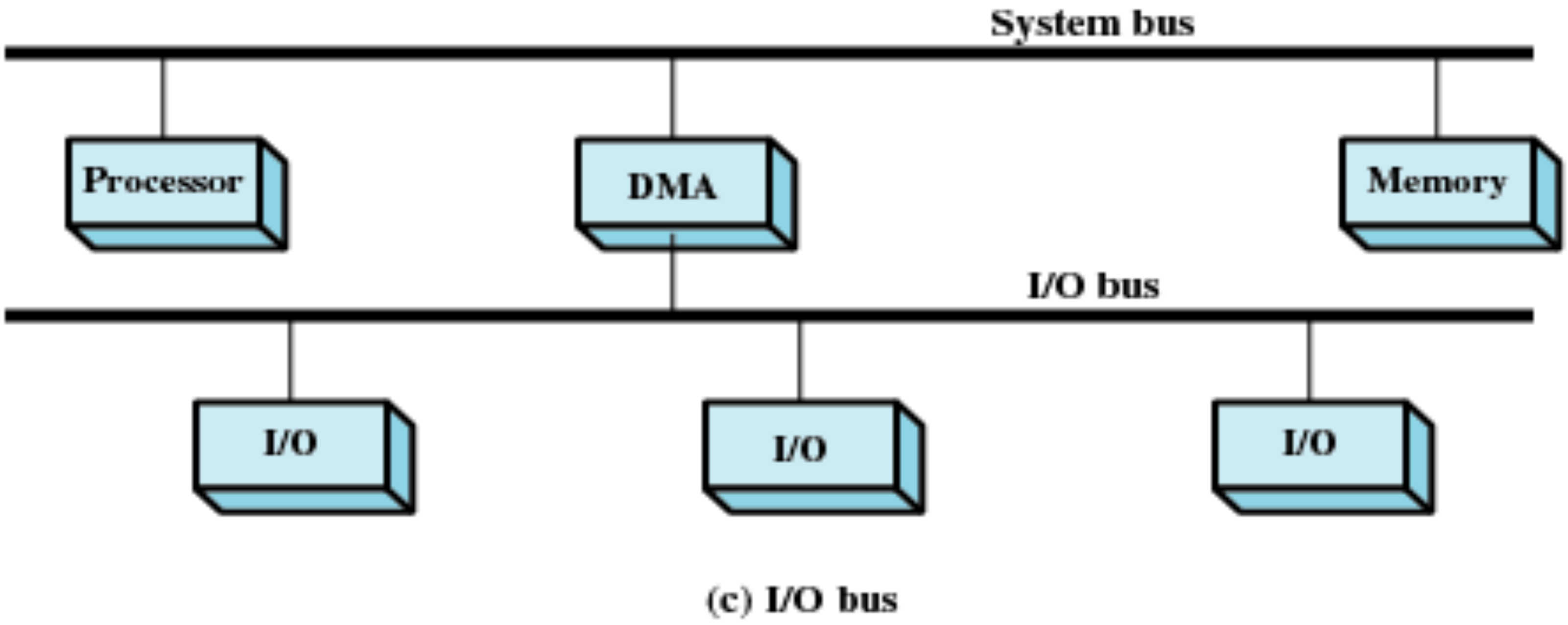


Figure 11.3 Alternative DMA Configurations

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- <u>11.3 Operating System Design Issues</u>
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

Operating System Design Issues(操作系统设计问题)

- Efficiency and generality are most important objectives in designing the I/O facility
- Efficiency (效率)
  - Most I/O devices extremely slow compared to main memory， I/O cannot keep up with processor speed
  - Use of multiprogramming(多道程序) allows for some processes to be waiting on I/O while another process executes
  - Swapping(交换) is used to bring in additional Ready processes which is an I/O operation

Operating System Design Issues

- Generality (通用性)
  - Desirable to handle all I/O devices in a uniform manner(统一模式)
  - Hide most of the details of device I/O in lower-level routines so that processes and upper levels see devices in general terms such as read, write, open, close, lock, unlock

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

I/O Buffering (I/O缓冲)

- Reasons for buffering(缓冲原因)
  - Processes must wait for I/O to complete before proceeding
  - Certain pages must remain in main memory during I/O
- Define of I/O buffering
  - Performs input transfers in advance of requests being made and performs output transfers some time after the request is made(预输入，缓输出)

I/O Buffering

- Block-oriented(面向块)
  - Information is stored in fixed sized blocks
  - Transfers are made a block at a time
  - Used for disks and tapes
- Stream-oriented(面向流)
  - Transfers information as a stream of bytes
  - Used for terminals, printers, communication ports, mouse and other pointing devices, and most other devices that are not secondary storage

Single Buffer(单缓冲)

- Operating system assigns one buffer in the system space for an I/O request
- Block-oriented single buffering
    1. Input transfers are made to buffer
    2. Block is moved to user space when needed
    3. Another block is requested immediately
        - Read ahead(预读), or anticipated input(预输入)

Single Buffer

- Advantages of block-oriented single buffer
  - User process can process one block of data while next block is read in
  - Swapping can occur since input is taking place in system memory, not user memory
- Disadvantages of block-oriented single buffer
  - Operating system keeps track of assignment of system buffers to user processes
  - The swapping logic is affected

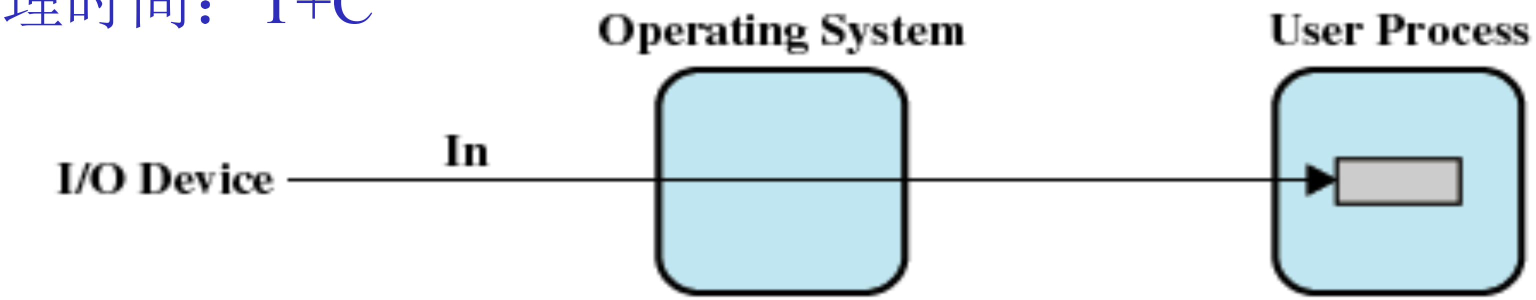
Single Buffer

- Stream-oriented single buffer
  - Line-at-a-time fashion
    - Input from or output to a terminal is one line at a time with carriage return signaling the end of the line
  - Bye-at-a-time fashion
    - Input and Output follow the producer/consumer model

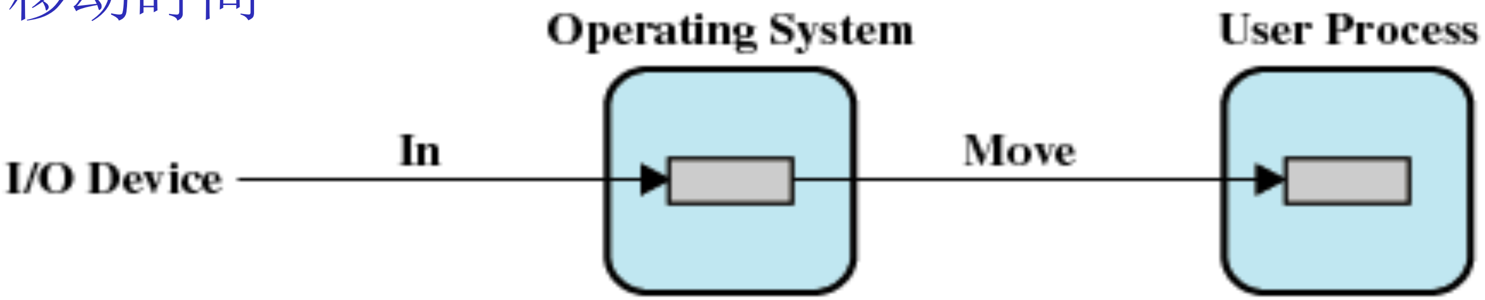
I/O Buffering

处理时间：T+C

T：传输时间
C：计算时间
M：移动时间



(a) No buffering

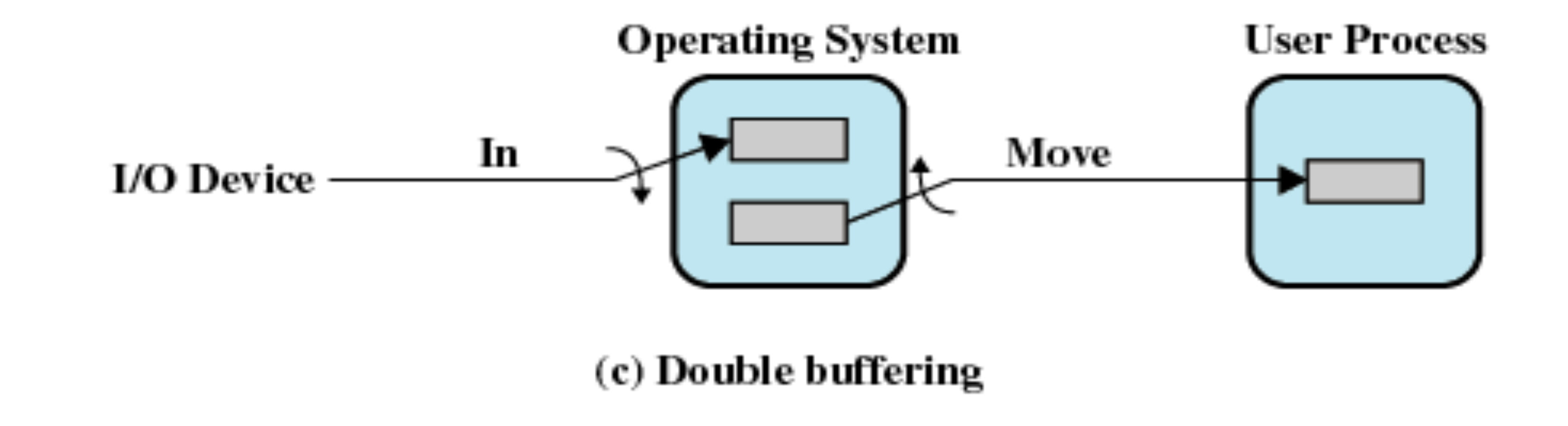


(b) Single buffering

处理时间：max[T, C]+M

Double Buffer(双缓冲)

- Use two system buffers instead of one
- A process can transfer data to or from one buffer while the operating system empties or fills the other buffer

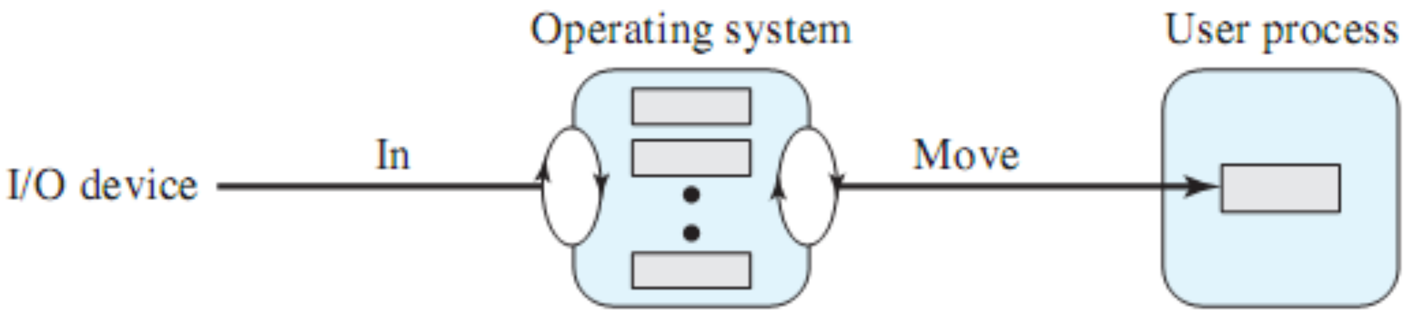


(c) Double buffering

处理时间：max[T, C]

Circular Buffer(循环缓冲)

- More than two buffers are used
- Each individual buffer is one unit in a circular buffer
- Used when I/O operation must keep up with process



(d) Circular buffering

Agenda

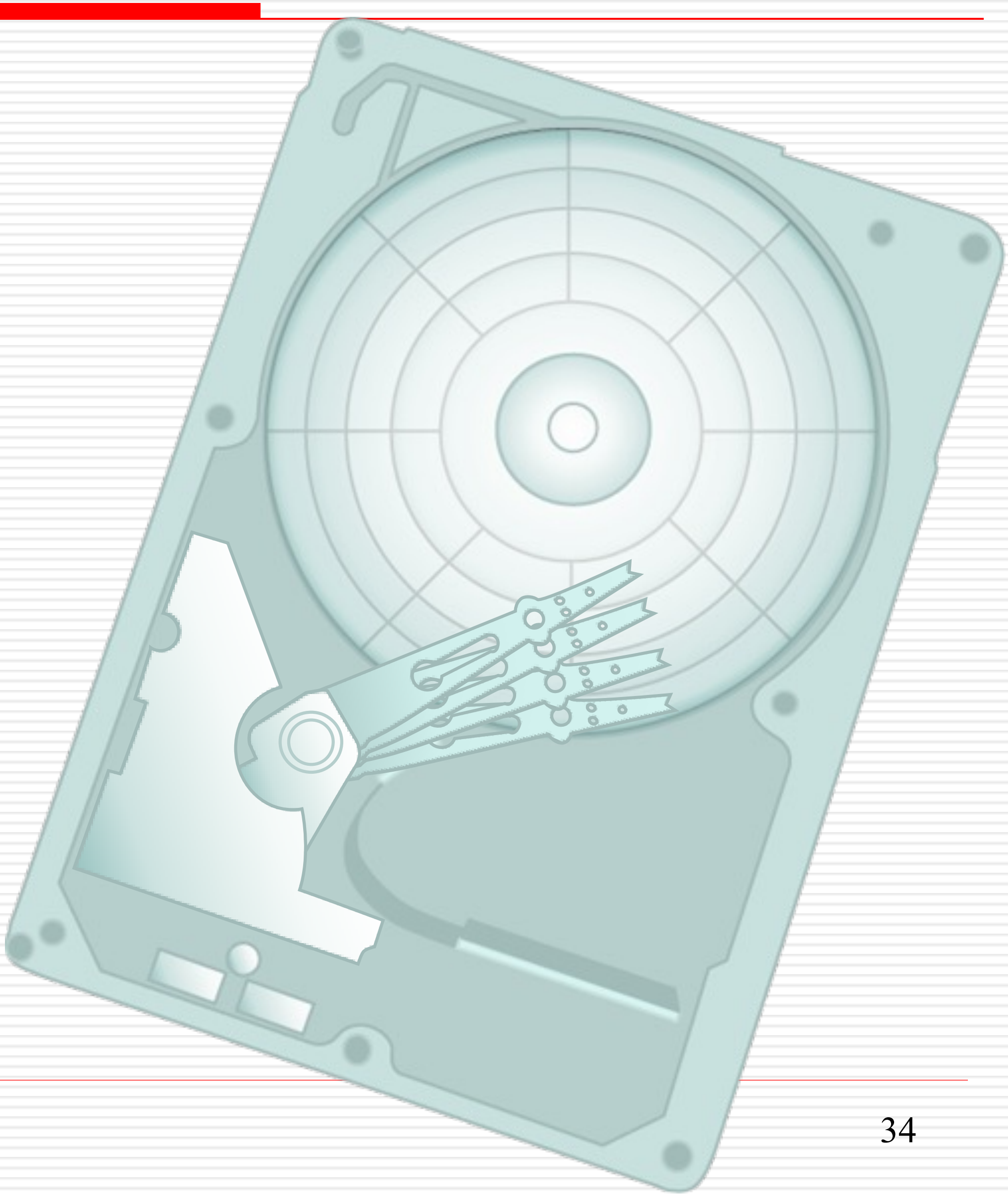
- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

Disk Performance Parameters

- To read or write, the disk head must be positioned at the desired track(磁道) and at the beginning of the desired sector(扇区)

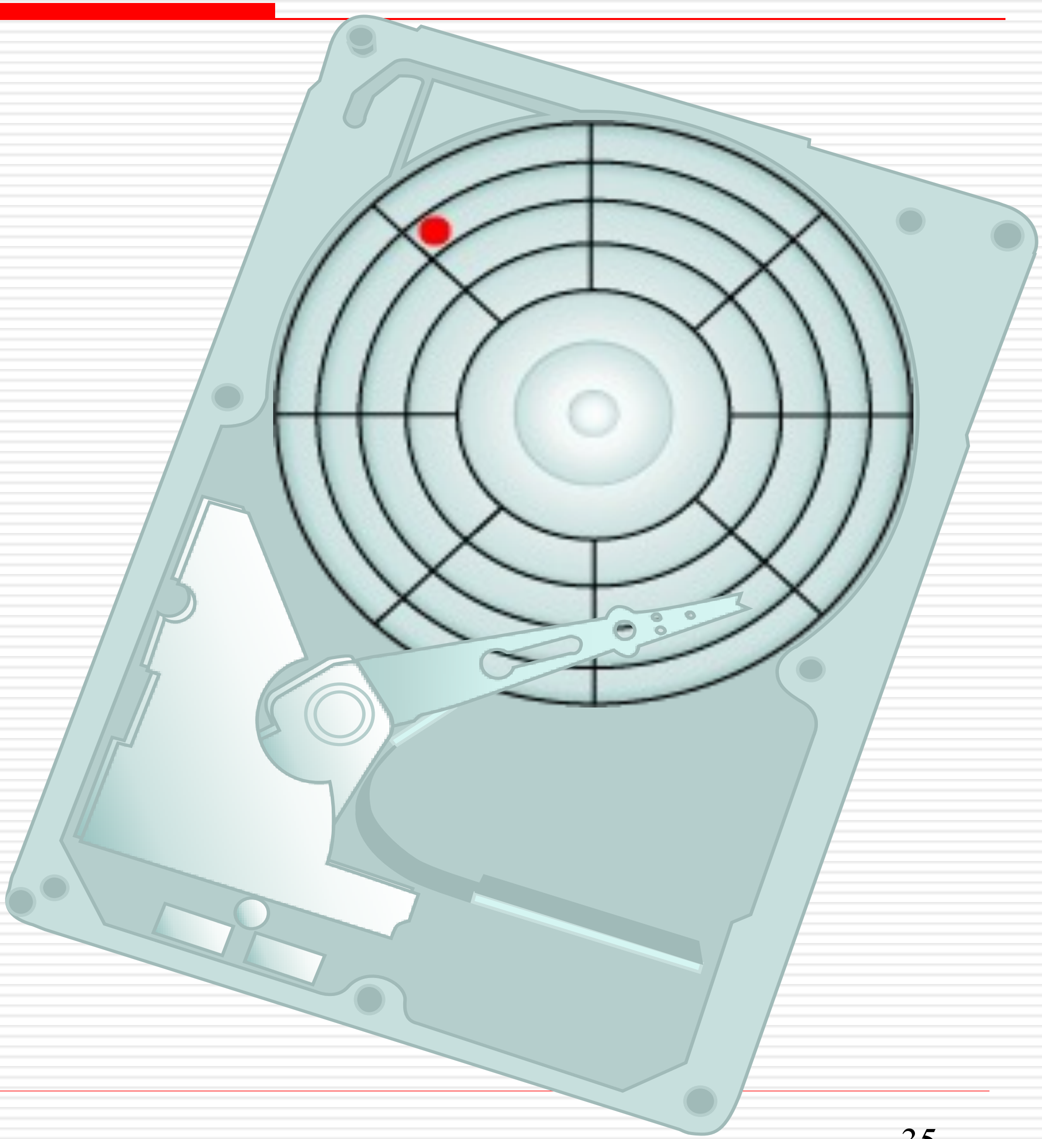
Seek time (寻道时间)

- Time it takes to position the head(磁头) at the desired track



Rotational delay (旋转延迟)

- Time its takes for the beginning of the sector to reach the head



Timing of a Disk I/O Transfer

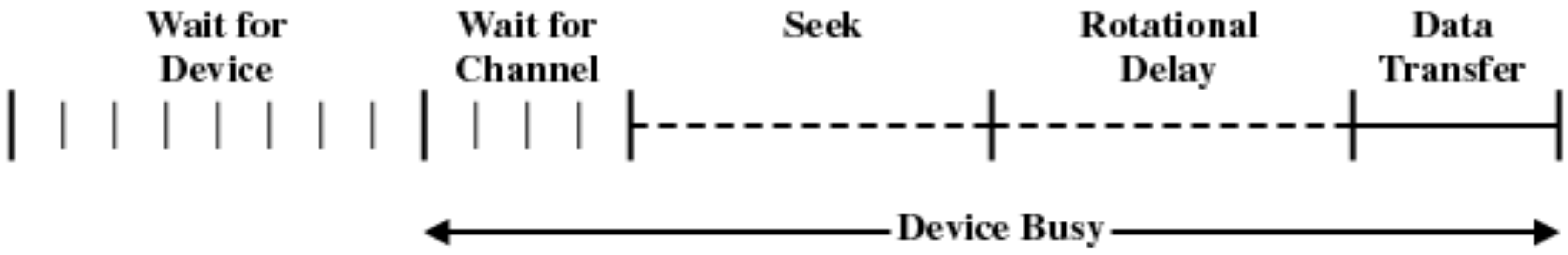


Figure 11.6 Timing of a Disk I/O Transfer

Disk Performance Parameters(磁盘性能参数)

- Access time(存取时间)
  - The time it takes to get in position to read or write
  - Sum of seek time (T_s) and rotational delay (*1/2r*)
- Transfer time(传输时间, *b/*(*rN*))
  - Data transfer occurs as the sector moves under the head
- Thus the total average access time can be expressed as:

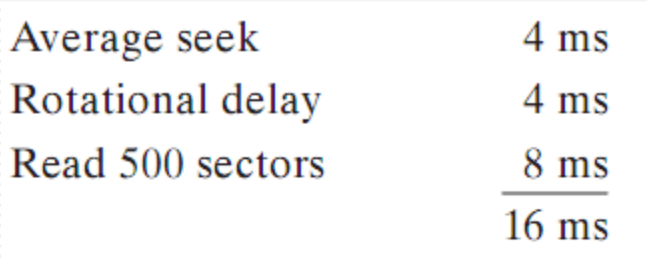
$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

Ta：总平均存取时间
Ts：平均寻道时间
r：旋转速度，转/秒
b：要传送字节数
N：一个磁道的字节数

A Timing Comparison

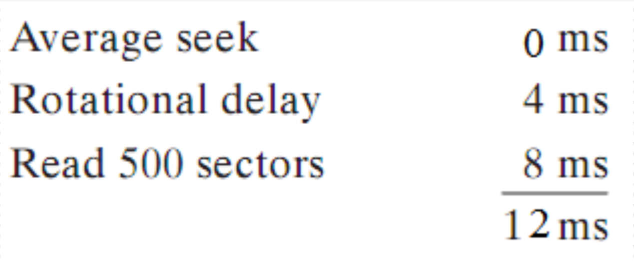
- Sequential access: 5 adjacent tracks, 500 sectors/track, total 2500 sectors，r(转速)7500r/m

First track:

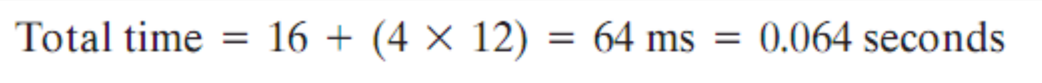


Average seek	4 ms
Rotational delay	4 ms
Read 500 sectors	8 ms
	16 ms

Next 4 tracks:

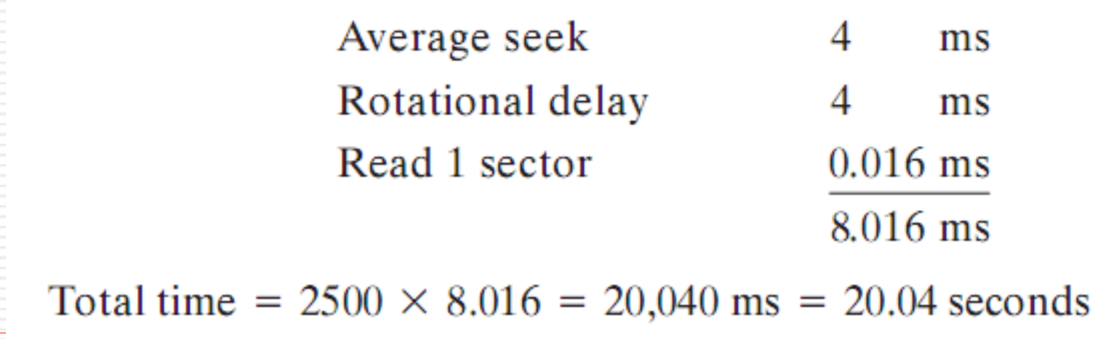


Average seek	0 ms
Rotational delay	4 ms
Read 500 sectors	8 ms
	12 ms



Total time = 16 + (4 × 12) = 64 ms = 0.064 seconds

- Random access: 2500 sectors



Average seek	4 ms
Rotational delay	4 ms
Read 1 sector	0.016 ms
	8.016 ms

Total time = 2500 × 8.016 = 20,040 ms = 20.04 seconds

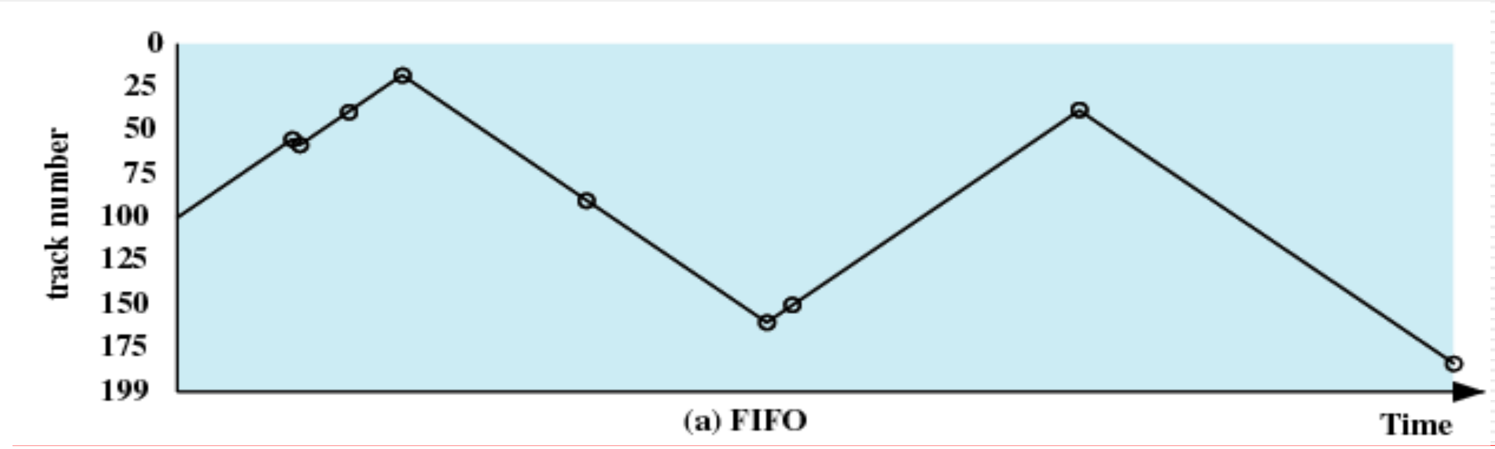
Disk Scheduling Policies (磁盘调度策略)

- Seek time and rotational delay are the reasons for differences in performance
  - The key to increase performance of disk is to minimize seek time
- Random scheduling
  - For a single disk there will be a number of I/O requests
  - If requests are selected randomly(random scheduling, 随机调度), we will poor performance
  - used to evaluate other techniques

Disk Scheduling Policies

- First-in, first-out (FIFO)
  - Process request sequentially
  - Fair to all processes
  - Approaches random scheduling in performance if there are many processes

55、58、39、18、90、160、150、38、184



(a) FIFO

Disk Scheduling Policies

- Priority(优先级)
  - Goal is not to optimize disk use but to meet other objectives
  - Short batch jobs may have higher priority
  - Provide good interactive response time

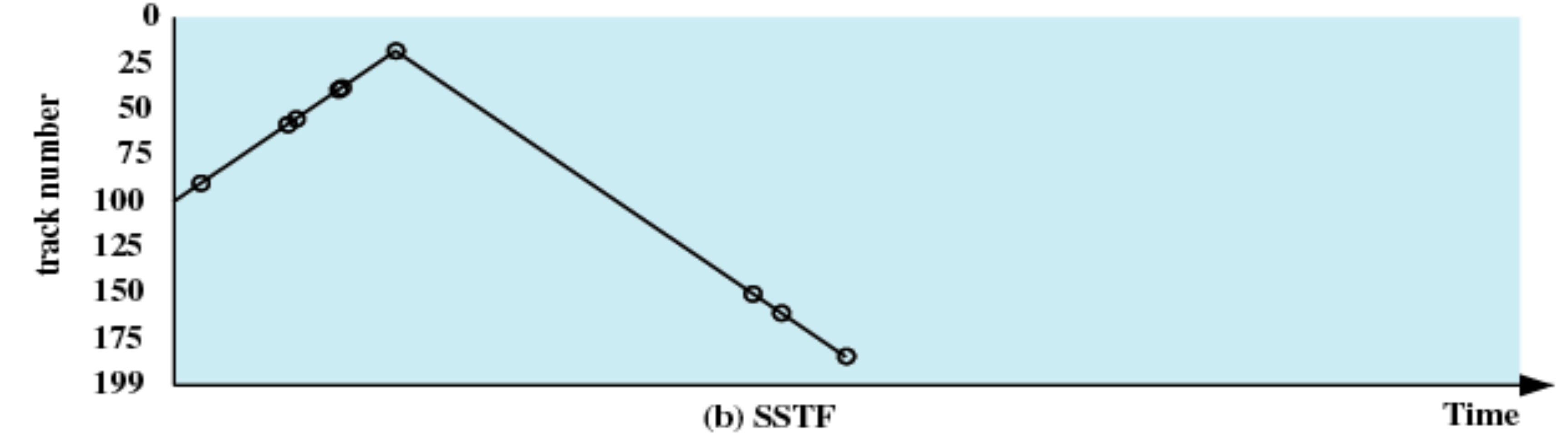
Disk Scheduling Policies

- Last-in, first-out (LIFO)
  - Good for transaction processing systems
    - The device is given to the most recent user so there should be little arm movement
  - Possibility of starvation since a job may never regain the head of the line

Disk Scheduling Policies

- Shortest Service Time First (SSTF)
  - Select the disk I/O request that requires the least movement of the disk arm from its current position
  - Always choose the minimum Seek time

55、58、39、18、90、160、150、38、184

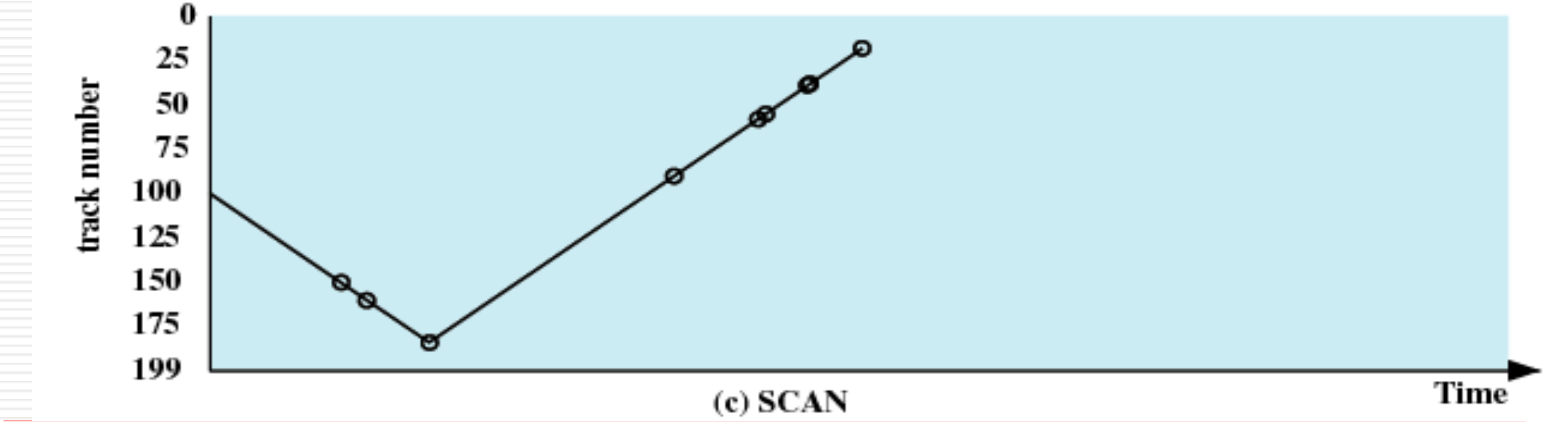


(b) SSTF

Disk Scheduling Policies

- SCAN
  - Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction
  - Direction is reversed

55、58、39、18、90、160、150、38、184

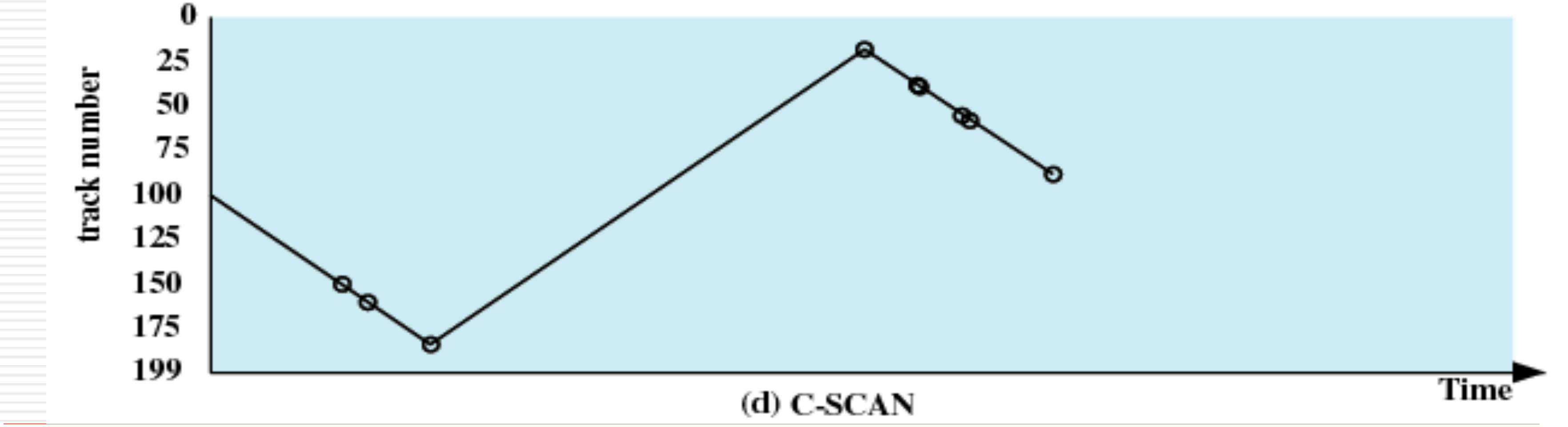


(c) SCAN

Disk Scheduling Policies

- C-SCAN (circular SCAN)
  - Restricts scanning to one direction only
  - When the last track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again

55、58、39、18、90、160、150、38、184



(d) C-SCAN

Disk Scheduling Policies

- N-step-SCAN
  - Segments the disk request queue into subqueues of length N
  - Subqueues are processed one at a time, using SCAN
  - New requests added to other queue when queue is processed
- FSCAN
  - Two queues
  - One queue is empty for new requests

Disk Scheduling Algorithms

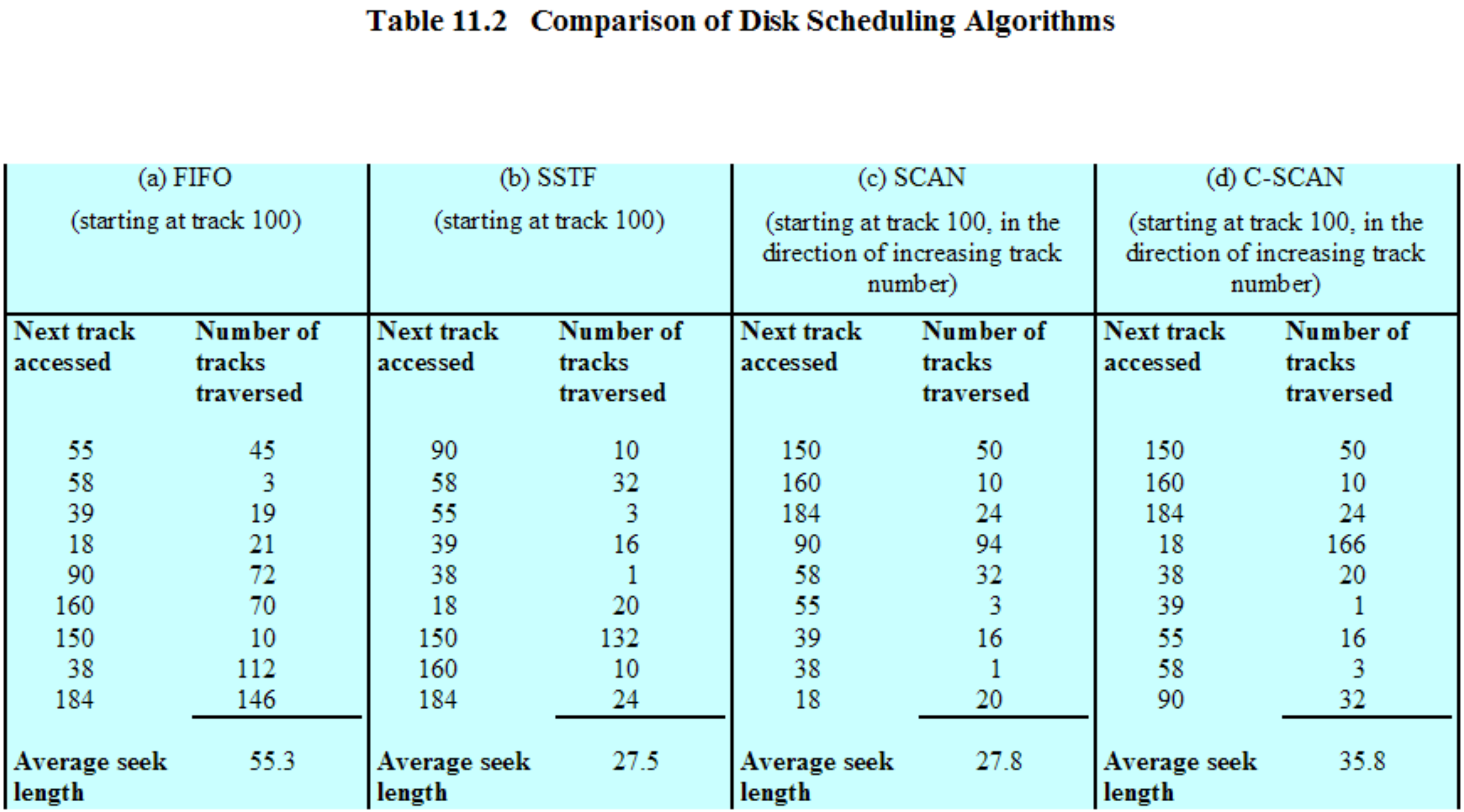


Table 11.2 Comparison of Disk Scheduling Algorithms

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	**Average seek length**	27.5	**Average seek length**	27.8	**Average seek length**	35.8

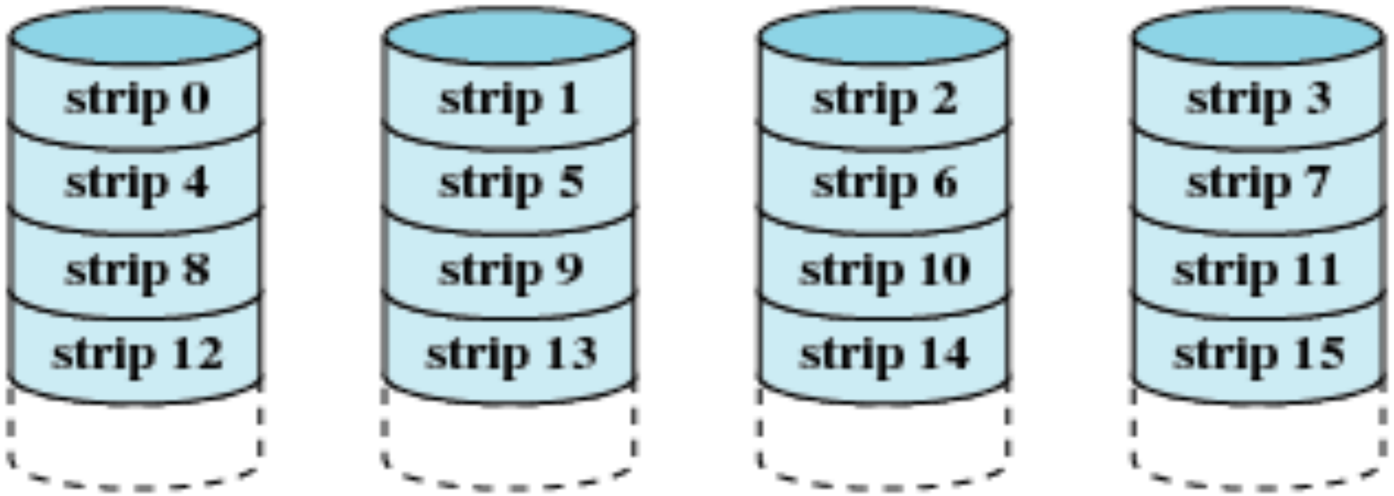
Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

RAID

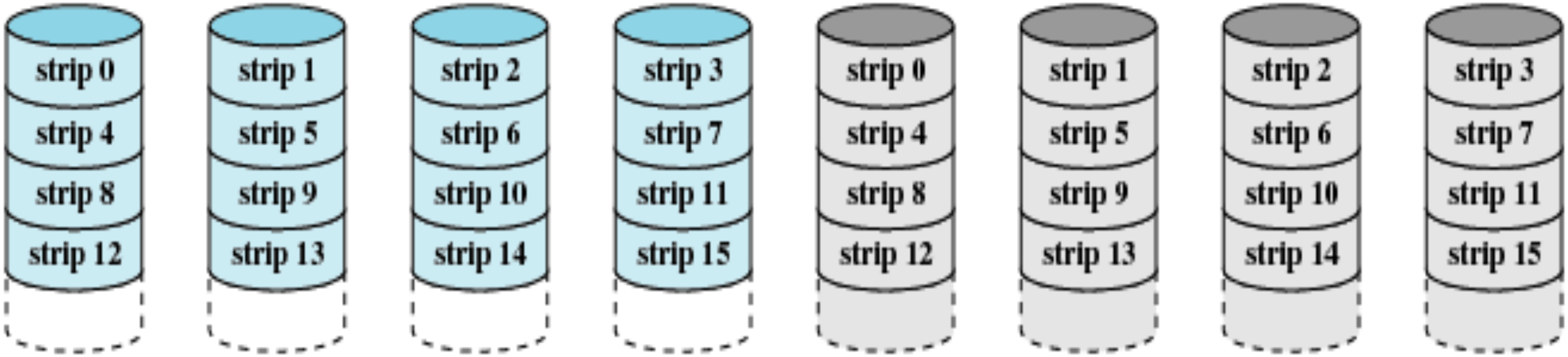
- Redundant Array of Independent Disks (独立冗余磁盘阵列) or Redundant Array of Inexpensive (廉价冗余磁盘整列)
- Set of physical disk drives viewed by the operating system as a single logical drive
- Data are distributed across the physical drives of an array
- Redundant disk capacity is used to store parity information

RAID 0 (non-redundant)



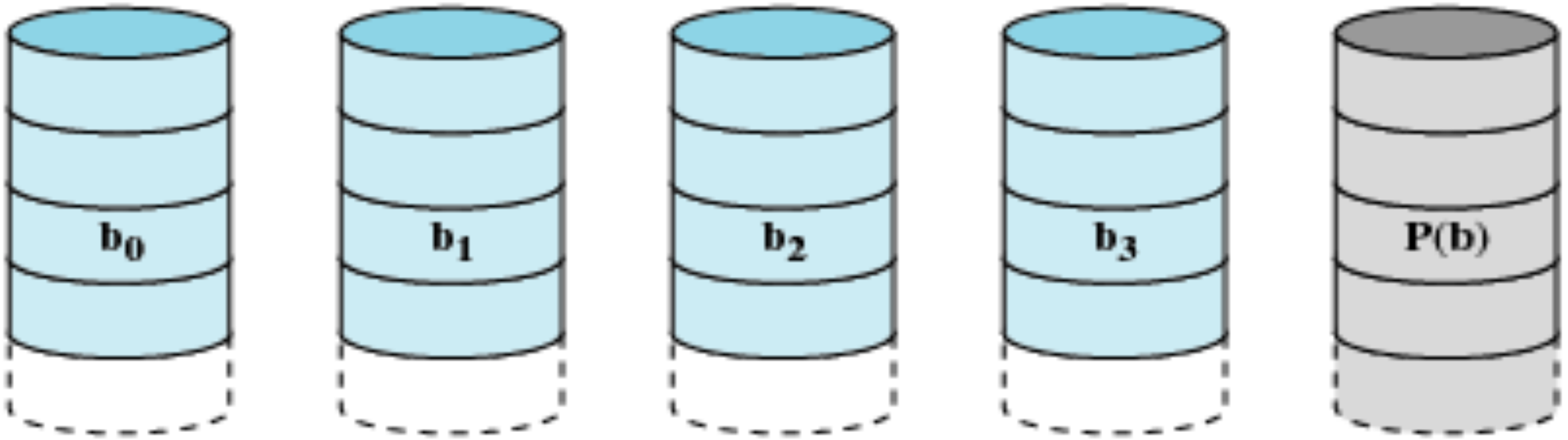
(a) RAID 0 (non-redundant)

RAID 1 (mirrored)



(b) RAID 1 (mirrored)

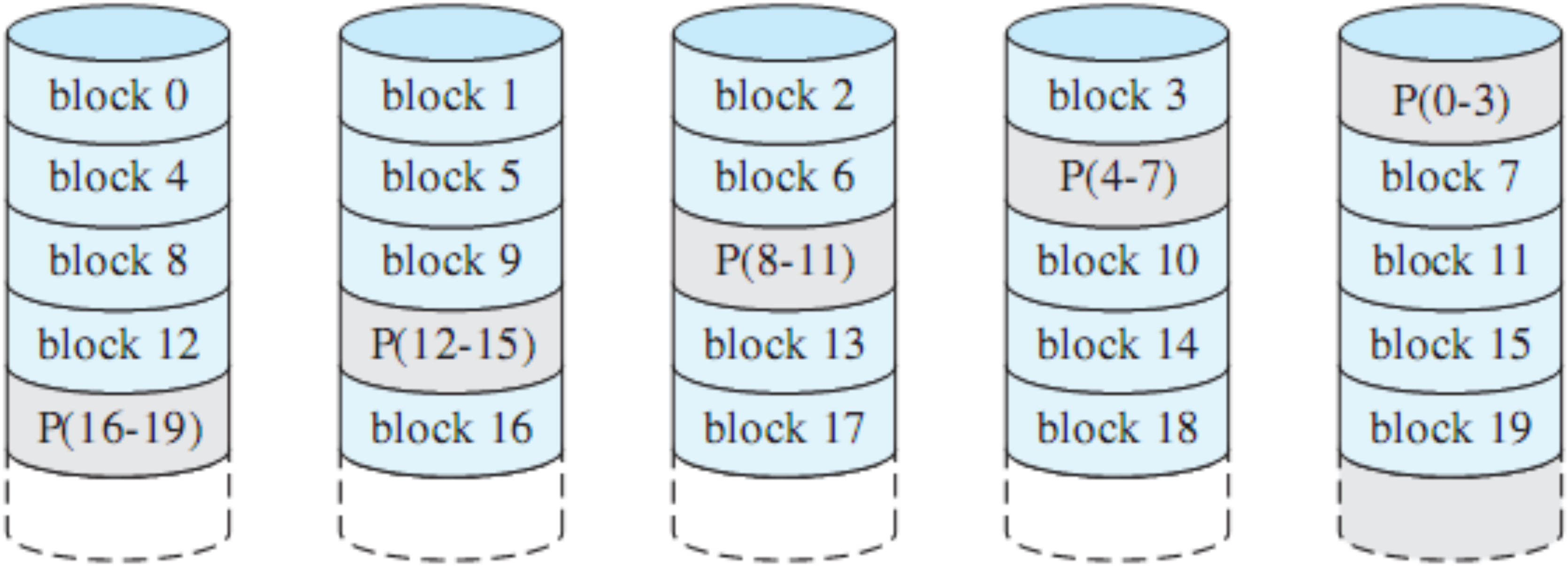
RAID 3 (bit-interleaved parity)



(d) RAID 3 (bit-interleaved parity)

RAID3(交错位奇偶校验)

RAID 5 (block-level distributed parity)



(f) RAID 5 (block-level distributed parity)

交错块分布奇偶校验

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

Disk Cache

- Buffer in main memory for disk sectors
- Contains a copy of some of the sectors on the disk
- Two design issues:
  - Method to transfer the block of data from the disk cache to memory assigned to the user process(缓存数据 与 用户空间 交换)
    - Data move
    - Pointer passing
  - The replacement strategy(置换策略) when the disk cache is full for store new data

Replacement Strategy 1: Least Recently Used (最近最少使用)

1. The block that has been in the cache the longest with no reference to it is replaced
2. The cache consists of a stack of blocks
3. Most recently referenced block is on the top of the stack
4. When a block is referenced or brought into the cache, it is placed on the top of the stack

Replacement Strategy 1: Least Recently Used

5. The block on the bottom of the stack is removed when a new block is brought in
6. Blocks don’t actually move around in main memory，A stack of pointers is used

Replacement Strategy 2: Least Frequently Used (最不常用)

1. The block that has experienced the fewest references is replaced
2. A counter is associated with each block
3. Counter is incremented each time block accessed
4. Block with smallest count is selected for replacement
5. Some blocks may be referenced many times in a short period of time and the reference count is misleading

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary