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

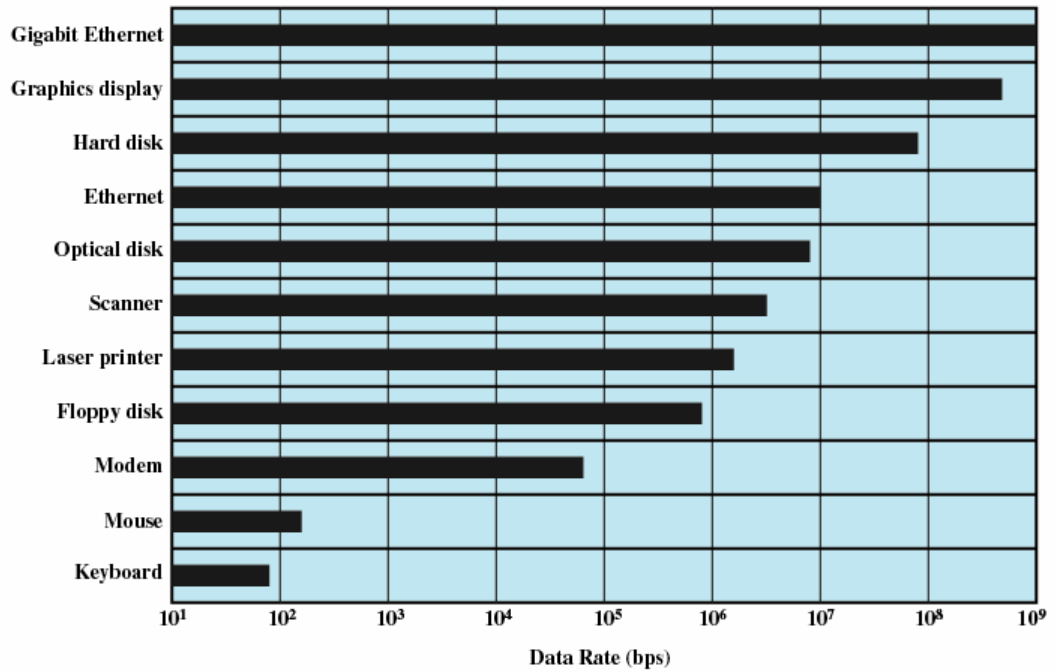
11.1 I/O Devices(1/4)

- Roughly grouped into three categories:I/O 设备分类
  - 1. Human readable(人可读): 字符流
    - Used to communicate with the user
    - Printers 、 Display 、 Keyboard 、 Mouse
  - 2. Machine readable(机器可读) ：字符流和块数据
    - Used to communicate with electronic equipment
    - Disk and tape drives 、 Sensors 、 Controllers
  - 3. Communication(通信) ：报文
    - Used to communicate with remote devices
    - Digital line drivers 、 Modems 、 Network device
- 将导致哪些差异

11.1 I/O Devices(2/4)

Differences in I/O Devices

1. Data rate 传输速率
   - May be differences of several orders of magnitude(数量级) between the data transfer rates
   - 如何减少对系统整体性能的影响？
   - 北桥：快速
   - 南桥：慢速



11.1 I/O Devices(3/4)

Differences in I/O Devices cont

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

   - 不同需求，管理的差异性

11.1 I/O Devices(4/4)

Differences in I/O Devices cont

3. Complexity of control
4. Unit of transfer
   - Data may be transferred as stream of bytes/blocks/network message
5. Data representation
   - Encoding schemes
6. Error conditions
   - Devices respond to errors differently

- 无法以统一标准解决所有 I/O 控制访问

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

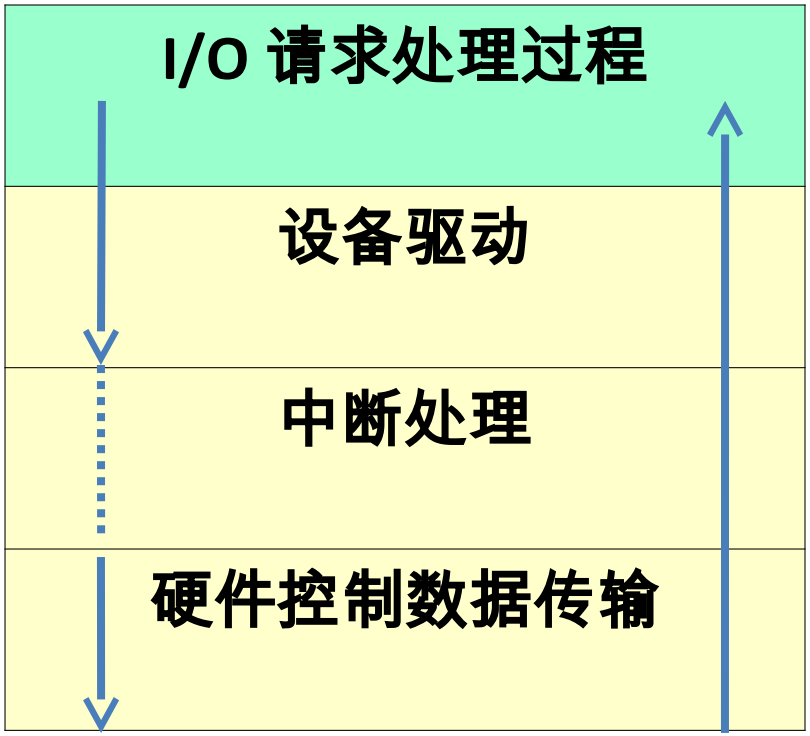
11.2 Organization of the I/O Function(1/8)

- synchronous/asynchronous communication 同步 / 异步通信
  - 同步：
    - Wait ：发出读写请求，然后阻塞，等待服务完成中断后，继续执行
    - Not Wait ：直接读或者写
  - 异步：
    - 利用缓冲区
    - 发出读写请求，及设置缓冲区地址
    - 进程返回继续执行不阻塞
    - 等待服务完成中断通知 CPU

11.2 Organization of the I/O Function(2/8)

用户态

内核态



- three techniques for performing I/O:
  - Programmed I/O: The processor issues an I/O command, on behalf of a process, to an I/O module; that process then busy waits for the operation to be completed before proceeding.
  - Interrupt-driven I/O: the process is nonblocking or blocking
  - Direct memory access (DMA):

11.2 Organization of the I/O Function(3/8)

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

11.2 Organization of the I/O Function(4/8)

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 the processor
   - Processor involved at beginning and end only

11.2 Organization of the I/O Function(5/8)

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(module+memory->'computer')
   - I/O module has its own local memory
   - It's a computer in its own right

11.2 Organization of the I/O Function(6/8)

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 from memory
3. When complete DMA module sends an interrupt signal to the processor

11.2 Organization of the I/O Function(7/8)

Direct Memory Access(直接存储器访问)

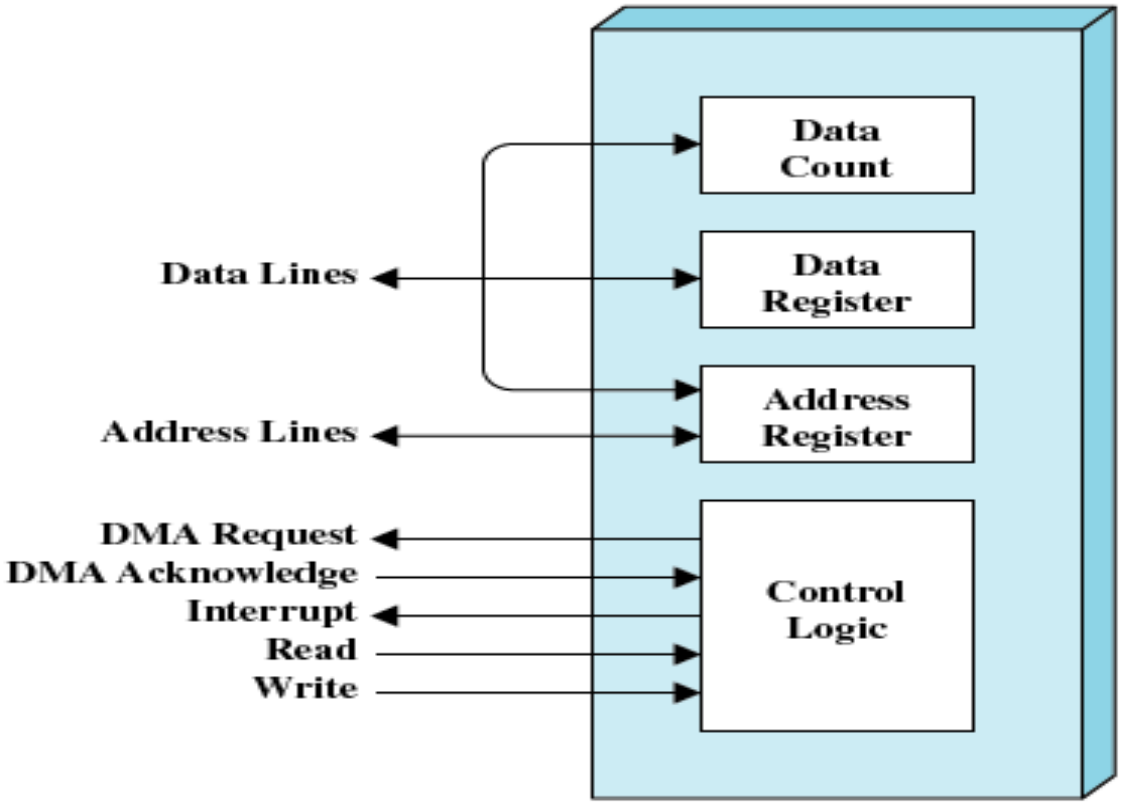


Figure 11.2 Typical DMA Block Diagram

11.2 Organization of the I/O Function(8/8)

DMA Configurations ：不断改进对访问效率的提升，总线剥离 -→ 独立总线

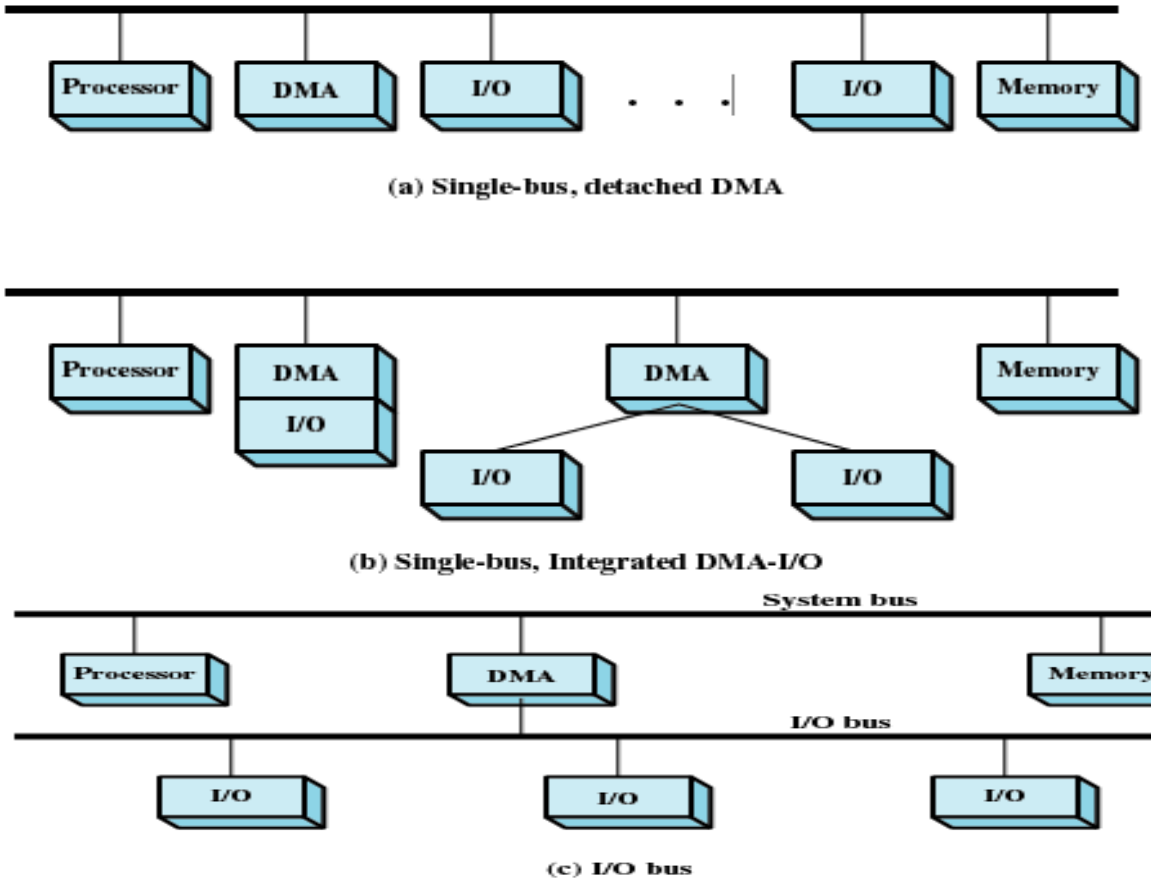


Figure 11.3 Alternative DMA Configurations

Agenda

- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues 操作系统设计问题
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

11.3 Operating System Design Issues(1/4)

- Two most important objectives
  - 1. 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 更多就绪队列换入意味着更多 disk 读写

11.3 Operating System Design Issues(2/4)

- 2. 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**

11.3 Operating System Design Issues(3/4)

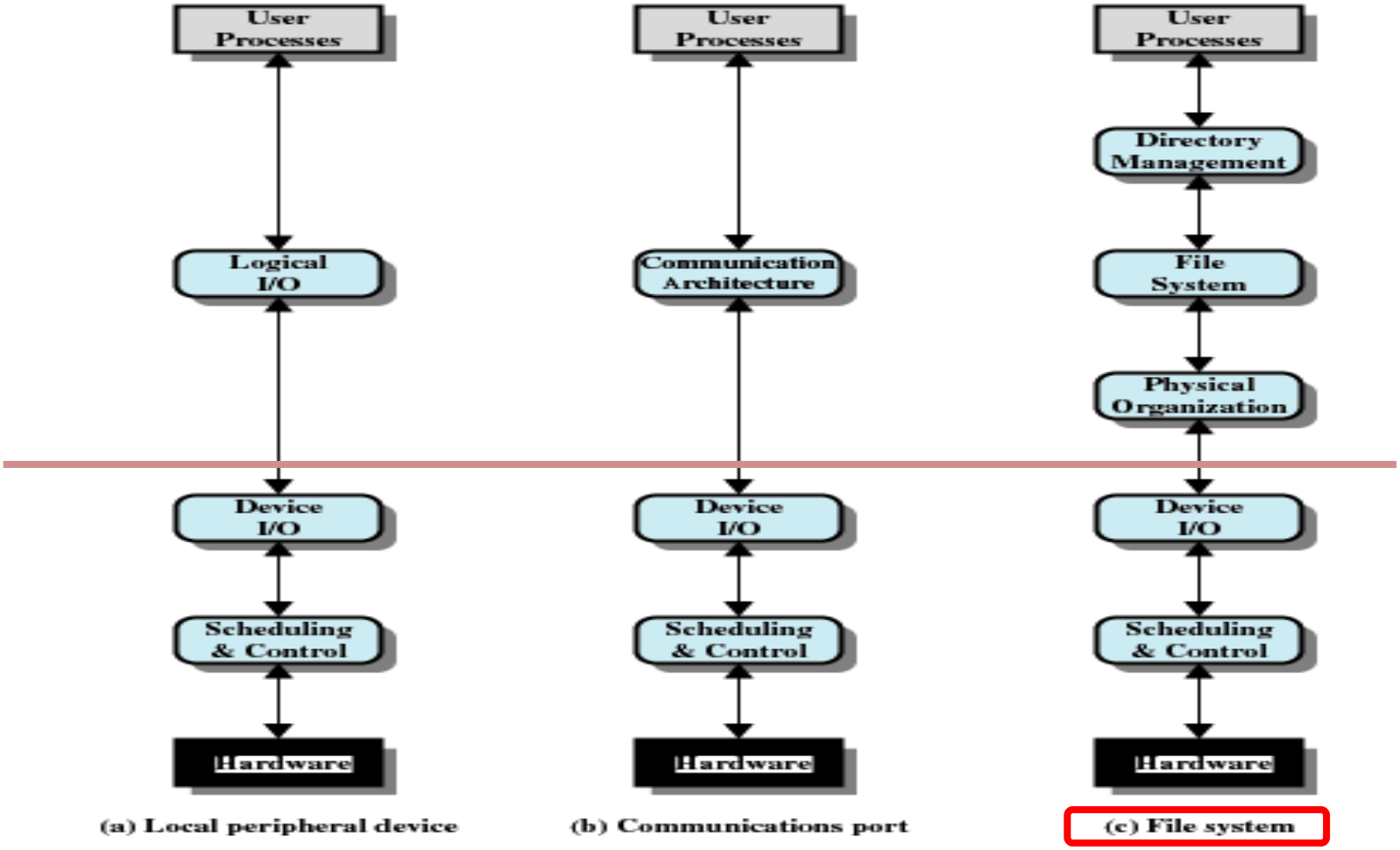


Figure 11.4 A Model of I/O Organization

11.3 Operating System Design Issues(4/4)

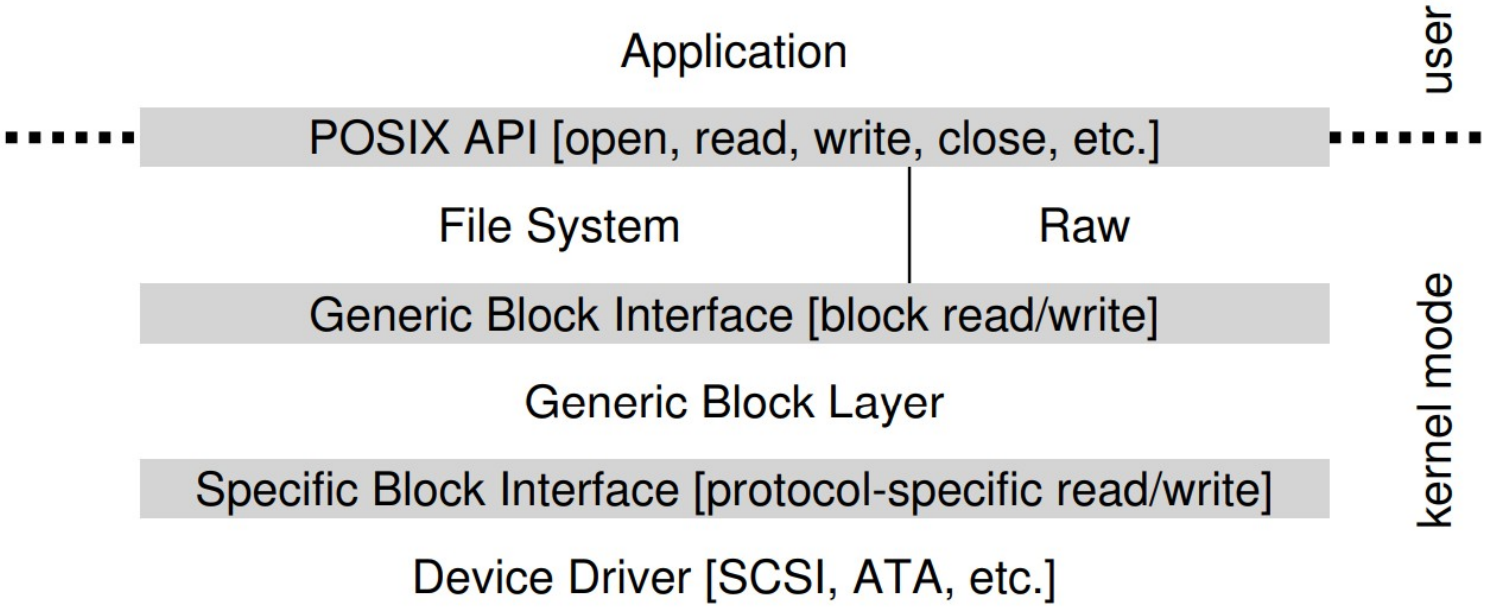


Figure 36.4: **The File System Stack**

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 I/O 缓冲
- 11.5 Disk Scheduling
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

11.4 I/O Buffering(1/8)

- Reasons for buffering(缓存原因)
  - Processes 进程 must wait for I/O to complete before proceeding
  - Certain pages must remain in main memory during I/O
    - block<-->suspend / deadlock

- 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) ： like cache
  - In memory

11.4 I/O Buffering(2/8)

- 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

11.4 I/O Buffering(3/8)

- 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)

11.4 I/O Buffering(4/8)

- Single Buffer(单缓冲)
  - 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 缓冲期间允许把进程换出
  - Disadvantages of block-oriented single buffer
    - Operating system keeps track of assignment of system buffers to user processes 追踪进程对应缓冲区
    - The swapping logic is affected 如果在同一磁盘

11.4 I/O Buffering(5/8)

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

11.4 I/O Buffering(6/8)

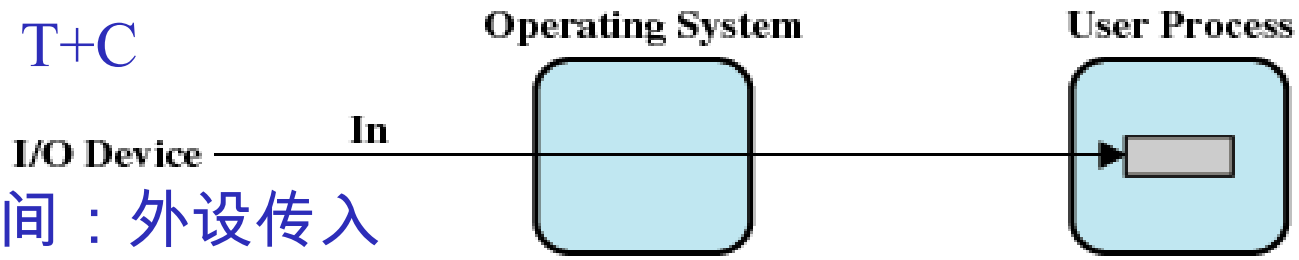
Single Buffer(单缓冲)

处理时间： T+C

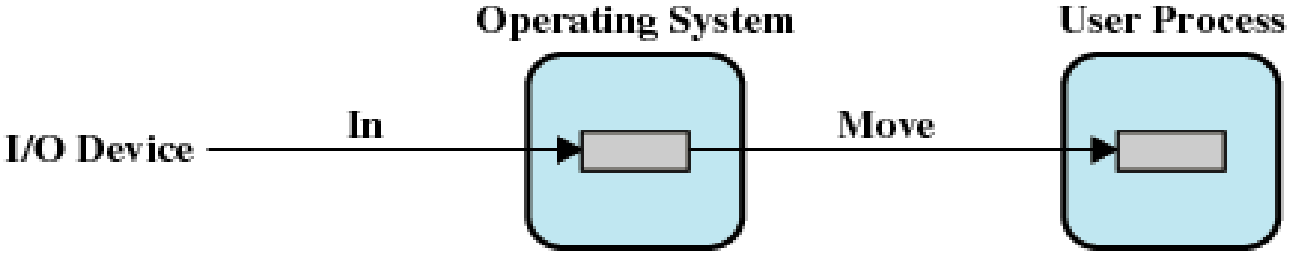
T ：传输时间：外设传入

C ：计算时间

M ：移动时间



(a) No buffering

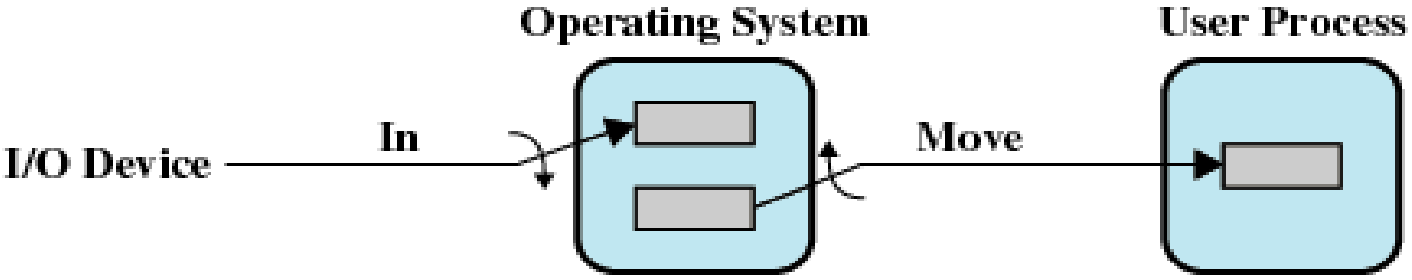


(b) Single buffering

处理时间： max[T, C]+M

11.4 I/O Buffering(7/8)

- 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
  - not typical producer consumer problem,as single buffer

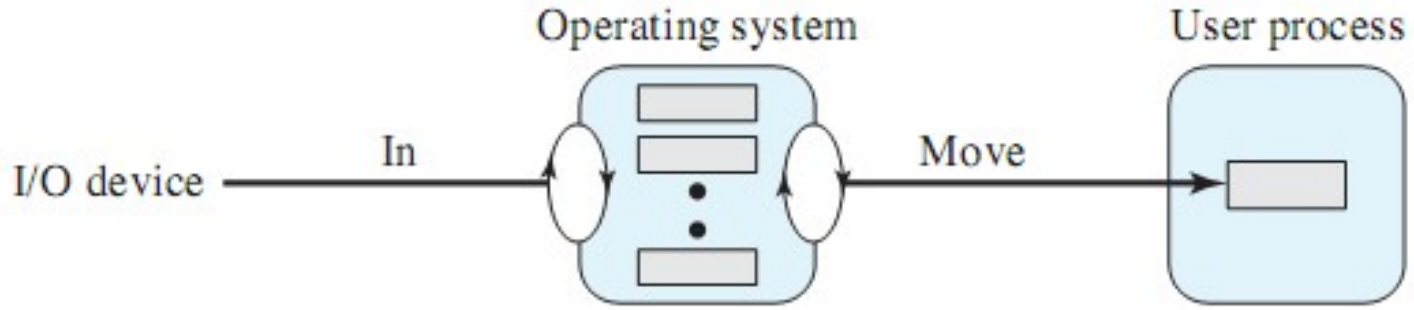


(c) Double buffering

处理时间： max[T, C]　如果 C<T, 则效果更明显。

11.4 I/O Buffering(8/8)

- 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
  - Bounded producer consumer problem



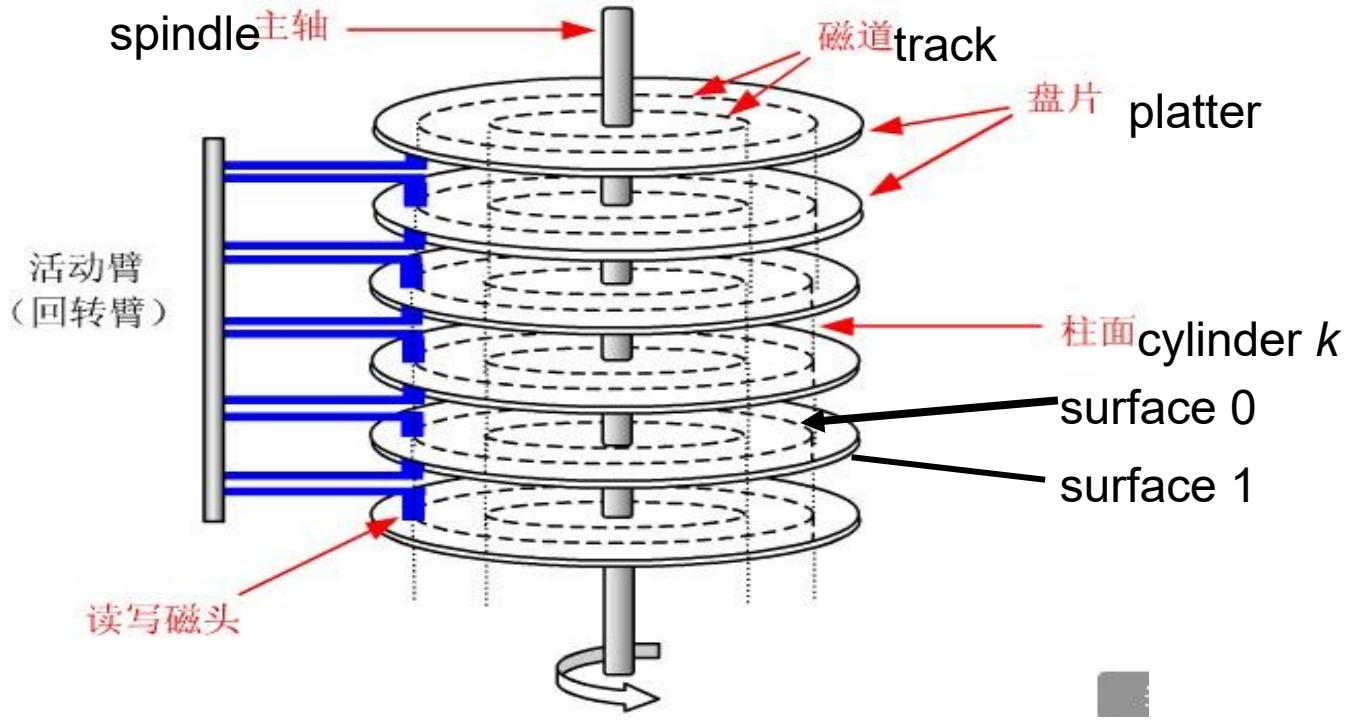
(d) Circular buffering

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
  - 11.5.1Disk Performance Parameters
  - 11.5.2Disk Scheduling Policies
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

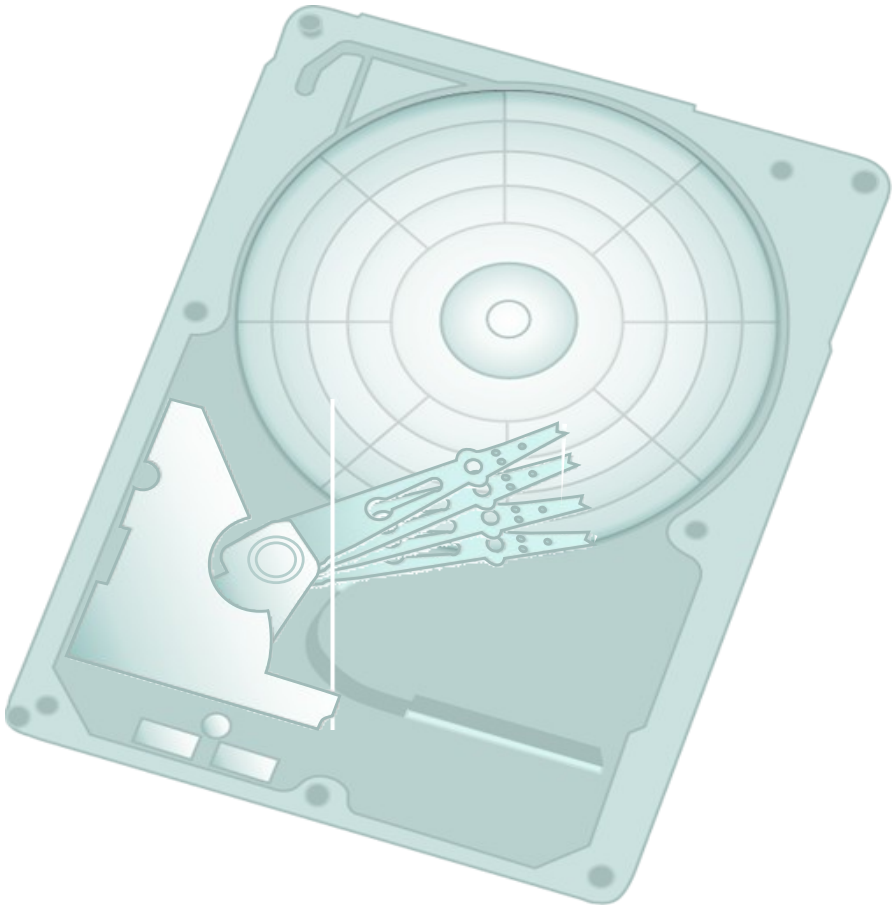
11.5.1Disk Performance Parameters(1/6)

- 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)



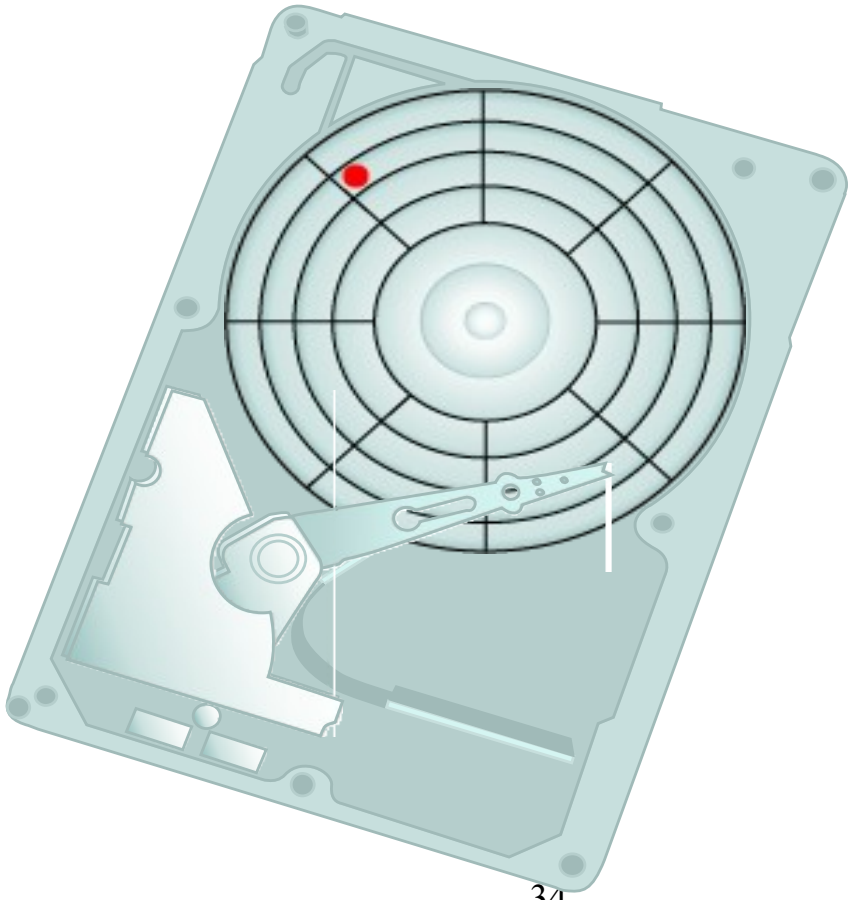
11.5.1Disk Performance Parameters(2/6)

- **Seek time (寻道时间)**
  - Time it takes to position the head(磁头) at the desired track



11.5.1Disk Performance Parameters(3/6)

- **Rotational delay (旋转延迟)**
  - Time its takes for the beginning of the sector to reach the head



11.5.1Disk Performance Parameters(4/6)

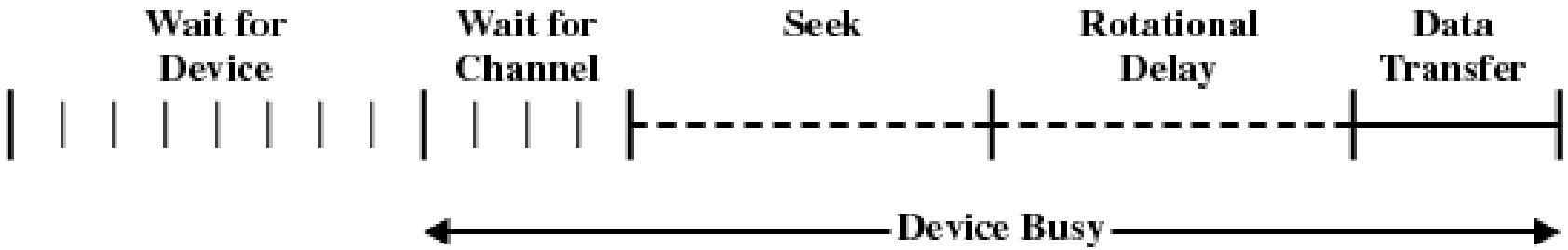


Figure 11.6 Timing of a Disk I/O Transfer

11.5.1Disk Performance Parameters(5/6)

- 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
  - 1/r: 一转的时间　　b/N 多少转
- 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

11.5.1Disk Performance Parameters(6/6)

- Sequential access: 5 adjacent tracks, 500 sectors/track, total 2500 sectors ， Ts 4ms ， 1/r ： 8ms

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

$$\text{Total time} = 16 + (4 \times 12) = 64 \text{ ms} = 0.064 \text{ seconds}$$

- Random access: 2500 sectors

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

$$\text{Total time} = 2500 \times 8.016 = 20{,}040 \text{ ms} = 20.04 \text{ seconds}$$

存放方式
影响速度

Agenda

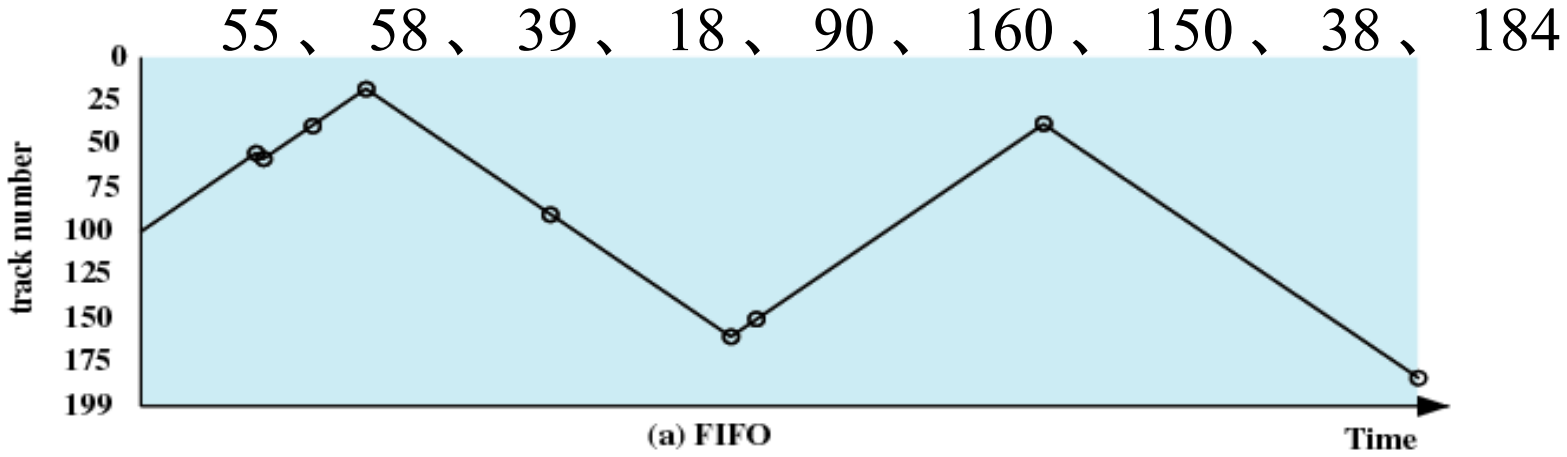
- 11.1 I/O Devices
- 11.2 Organization of the I/O Function
- 11.3 Operating System Design Issues
- 11.4 I/O Buffering
- 11.5 Disk Scheduling
  - 11.5.1Disk Performance Parameters
  - 11.5.2Disk Scheduling Policies
- 11.6 RAID
- 11.7 Disk Cache
- 11.8 Summary

11.5.2Disk Scheduling Policies(1/9)(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 减少寻道时间
- Random scheduling
  - For a single disk there will be a number of I/O requests
  - random scheduling 随机调度 yields poor performance
    - used to evaluate other techniques

11.5.2Disk Scheduling Policies(2/9)

- First-in, first-out (FIFO)
  - Process request sequentially
  - Fair to all processes
  - Approaches random scheduling in performance if there are many processes
- Intial position 100



11.5.2Disk Scheduling Policies(3/9)

- 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

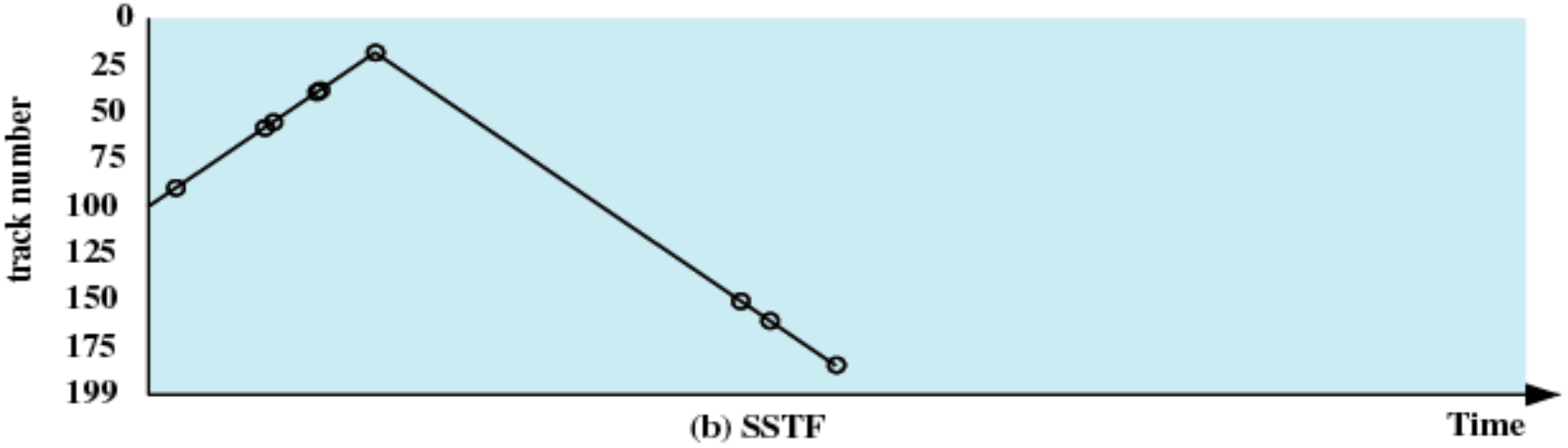
11.5.2Disk Scheduling Policies(4/9)

- Last-in, first-out (LIFO)
  - Good for transaction 事物处理 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

11.5.2Disk Scheduling Policies(5/9)

- 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

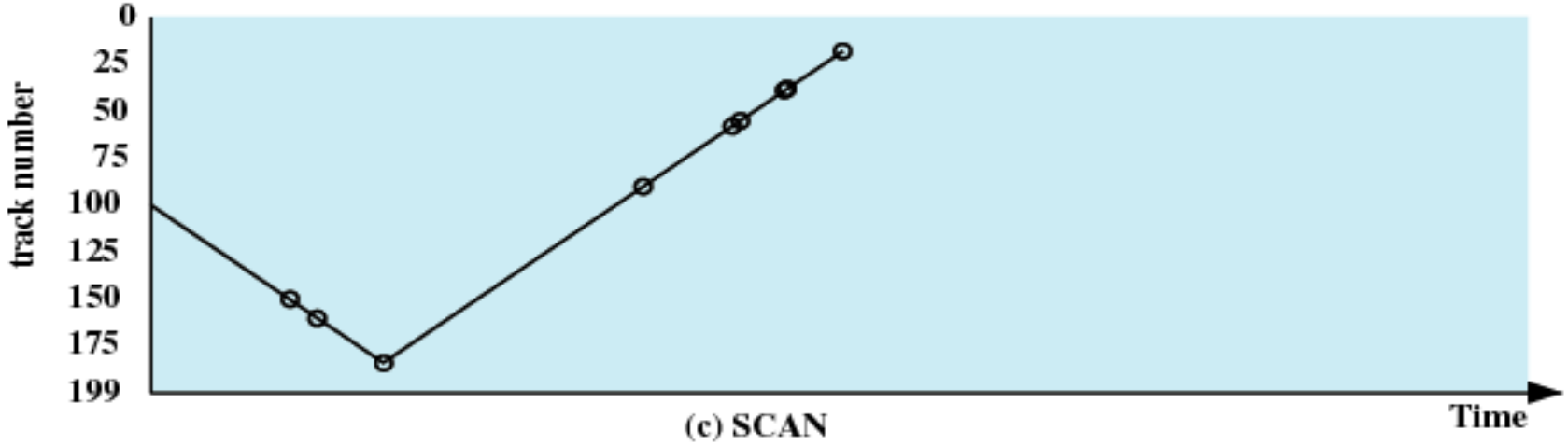
55、58、39、18、90、160、150、38、184



11.5.2Disk Scheduling Policies(6/9)

- SCAN 又名电梯算法
  - Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction
  - Direction is reversed 单调增 / 减直至到头原路折返

55 、 58 、 39 、 18 、 90 、 160 、 150 、 38 、 184

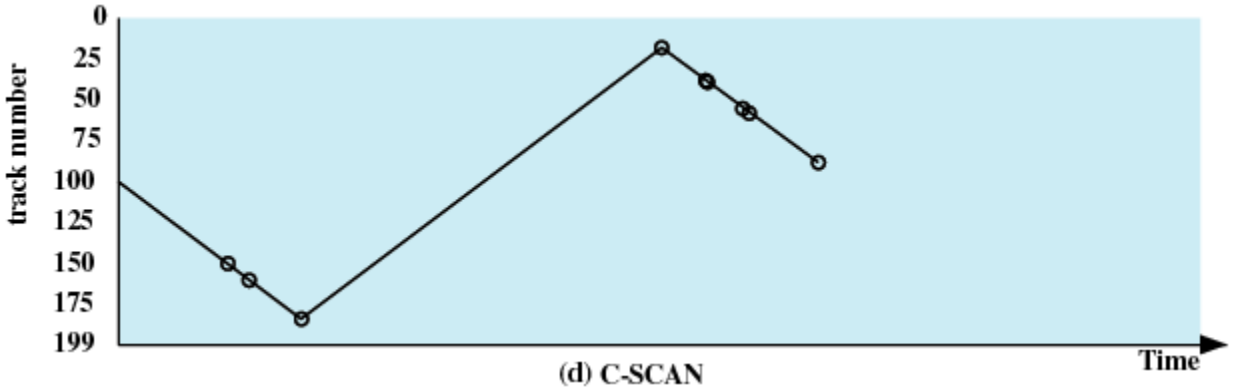


(c) SCAN

11.5.2Disk Scheduling Policies(7/9)

- C-SCAN (circular SCAN)
  - Restricts scanning to one direction only 仅单增
  - 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



11.5.2Disk Scheduling Policies(8/9)

- N-step-SCAN
  - Segments the disk request queue into subqueues of length N 按时间分段
  - 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

11.5 Disk Scheduling Algorithms(9/9)

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

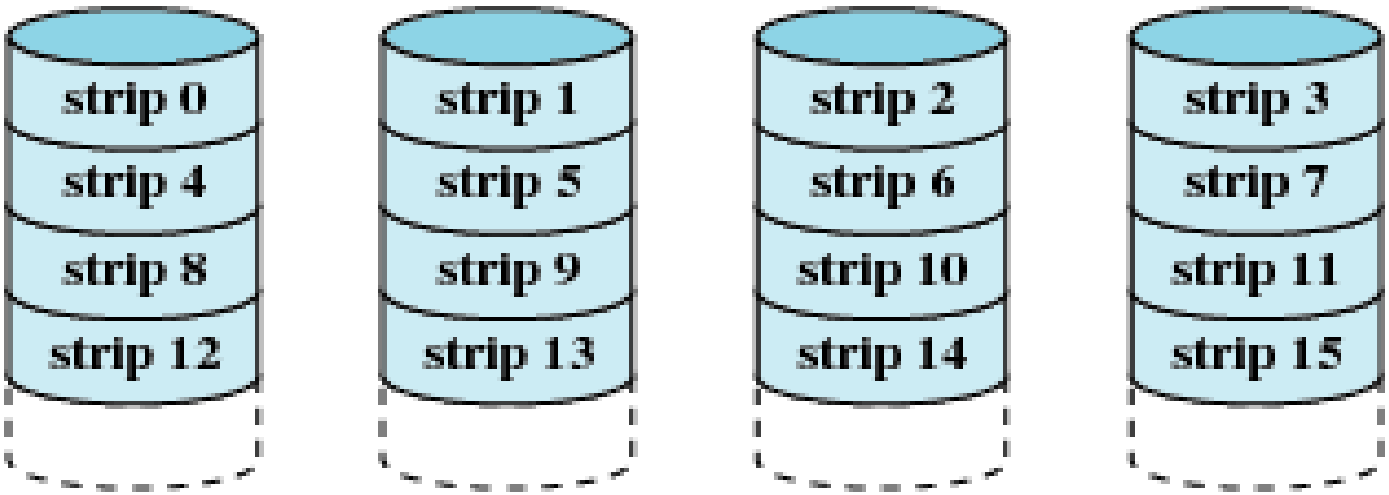
Agenda

- 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

11.6 RAID(1/6)

- Redundant Array of Independent Disks (独立冗余磁盘阵列) or Redundant Array of Inexpensive Disks (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 奇偶校验
  - RAID 0,1,5,6

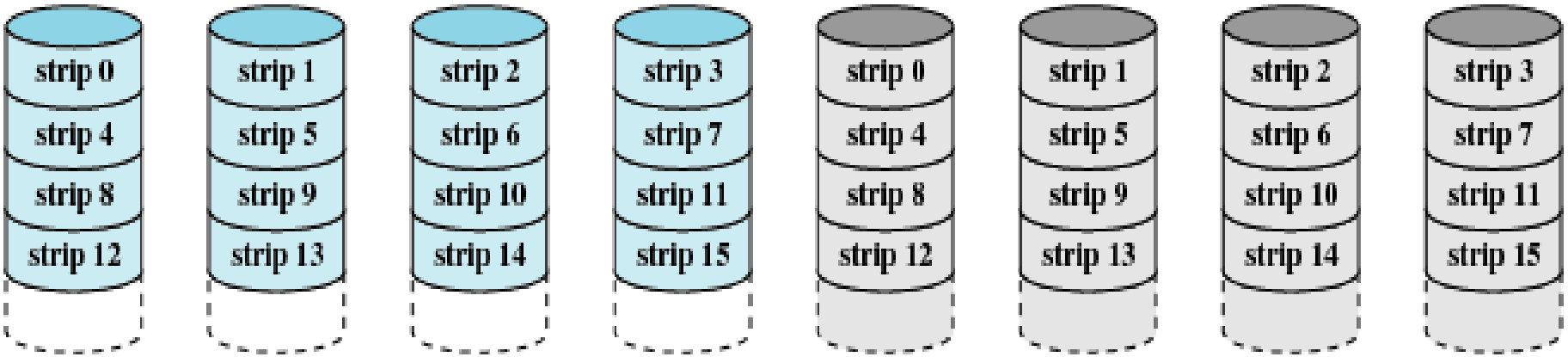
11.6 RAID 0 (non-redundant)(2/6)



(a) RAID 0 (non-redundant)

条带分布数据→并行访问、高响应率
无纠错→非高稳定性

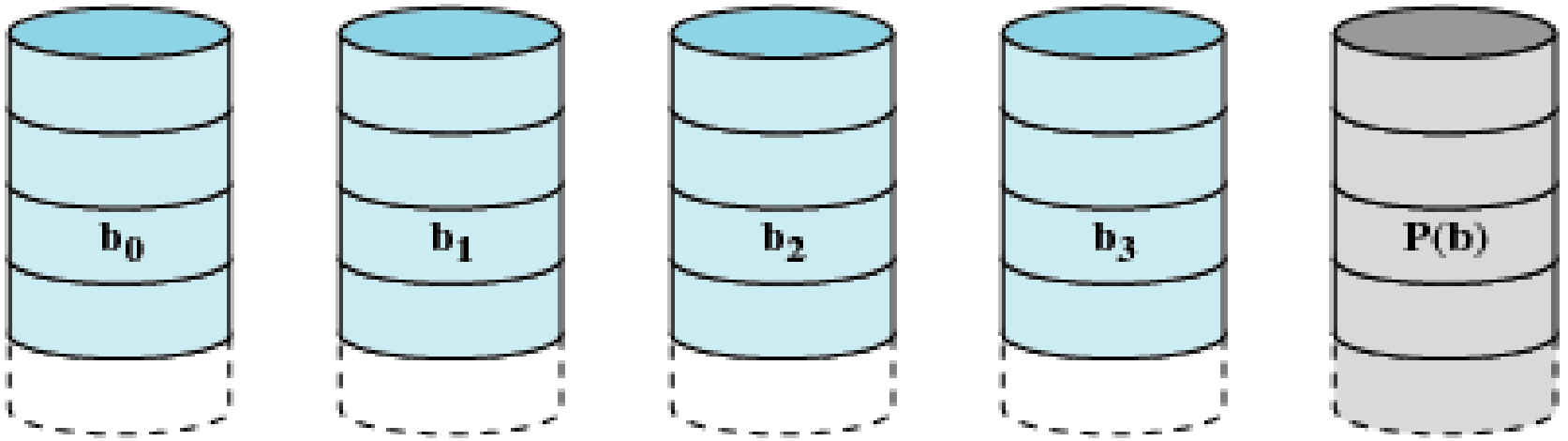
11.6 RAID 1 (mirrored)(3/6)



(b) RAID 1 (mirrored)

条带分布数据→并行访问、高响应率
镜像纠错
写速度降低，读相应好（可以并行任意读取一个）

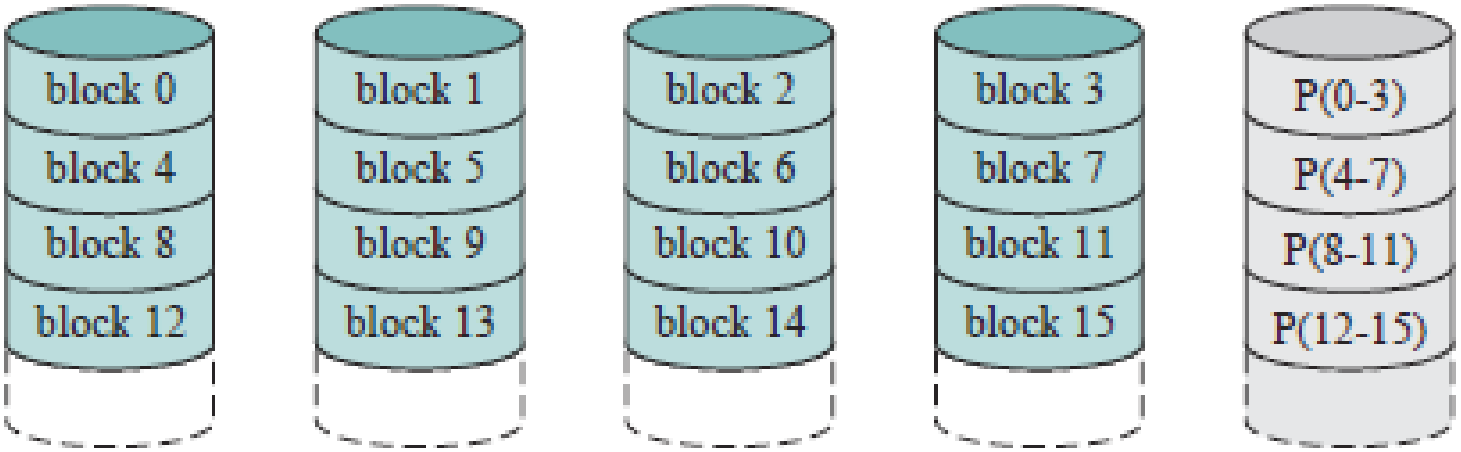
11.6 RAID 3 (bit-interleaved parity)(4/6)



(d) RAID 3 (bit-interleaved parity)

条带分布数据（ size 小 BYTE/WORD ）
同一位置 位 交错奇偶校验位纠错→一次只能一个 I/O 请求
因为要访问所有磁盘

11.6 RAID 4 (block-level parity)(5/6)



(e) RAID 4 (block-level parity)

条带分布数据

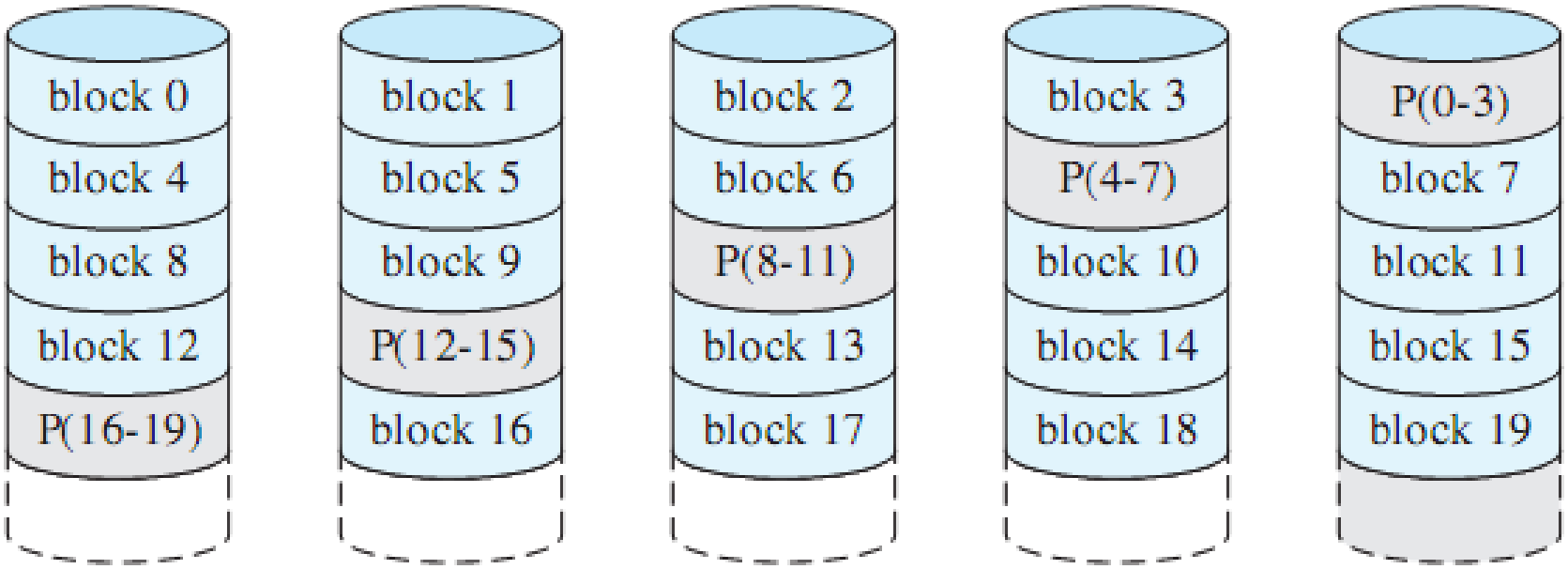
块奇偶校验位纠错→每次写需要操作 2 次读和 2 次写，X4 成为性能瓶颈（必访问）

$$X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i) \quad (11.1)$$

After the update, with potentially altered bits indicated by a prime symbol:

$$\begin{aligned} X4'(i) &= X3(i) \oplus X2(i) \oplus X1'(i) \oplus X0(i) \\ &= X3(i) \oplus X2(i) \oplus X1'(i) \oplus X0(i) \oplus X1(i) \oplus X1(i) \\ &= X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i) \oplus X1(i) \oplus X1'(i) \\ &= X4(i) \oplus X1(i) \oplus X1'(i) \end{aligned}$$

11.6 RAID 5 (block-level distributed parity)(6/6)



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

条带分布数据

分散块奇偶校验位纠错→解决 X4 性能瓶颈

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

11.7 Disk Cache(1/4)

- 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: reader writer problem
  - The replacement strategy(置换策略) when the disk cache is full for store new data

11.7 Disk Cache(2/4)

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

The block that has been in the cache the longest with no reference to it is replaced

1.The cache consists of a stack of blocks

2.Most recently referenced block is on the top of the stack

3.When a block is referenced or brought into the cache, it is placed on the top of the stack

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

11.7 Disk Cache(3/4)

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

The block that has experienced the fewest references is replaced

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

11.7 Disk Cache(4/4)

Different generations

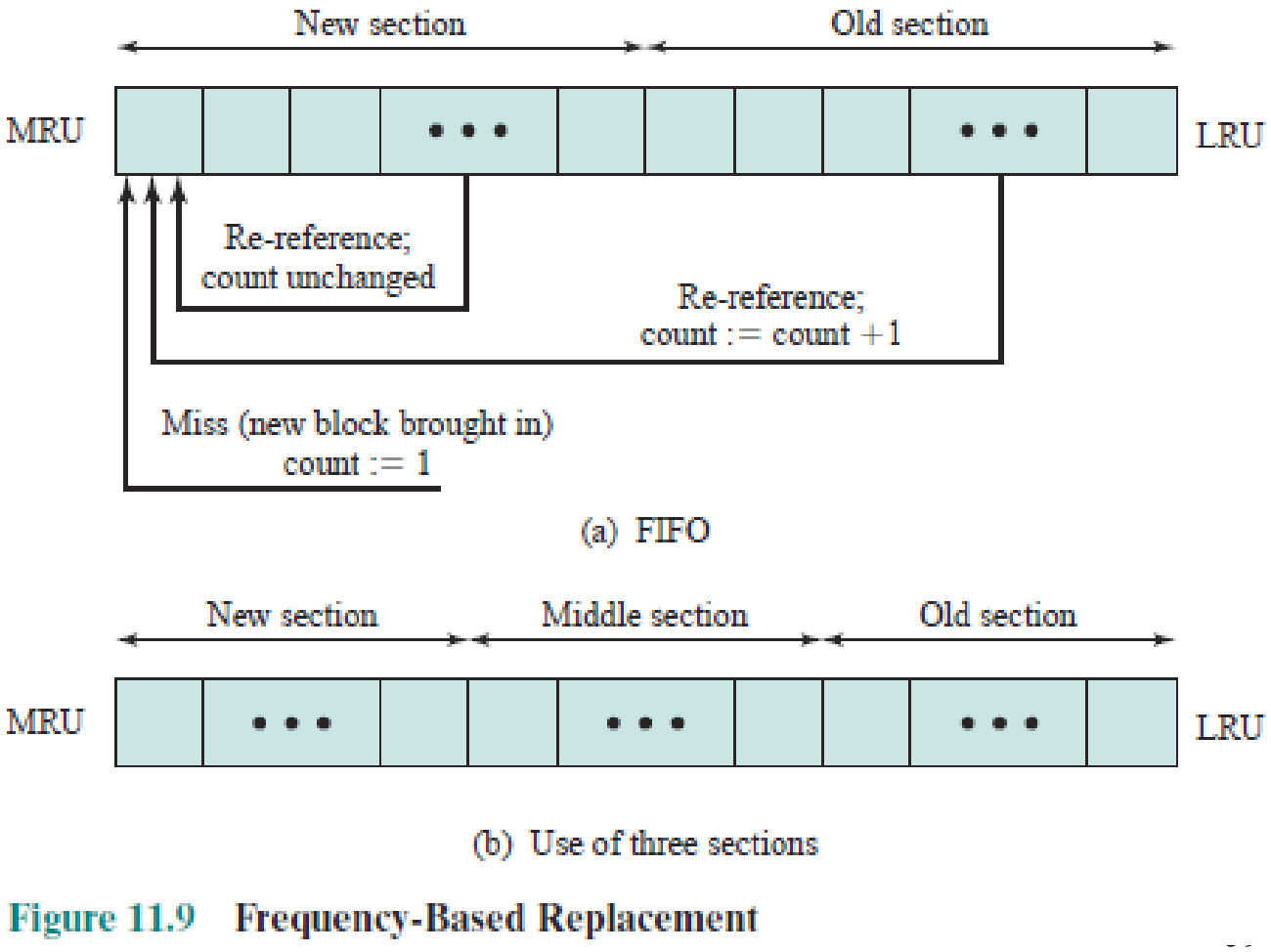


Figure 11.9 Frequency-Based Replacement

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