



Chapter 12: Mass-Storage Systems

大容量存储系统





Chapter 12: Mass-Storage Systems

- 12.1 Overview of Mass Storage Structure
- 12.2 Disk Structure
- 12.3 Disk Attachment
- 12.4 Disk Scheduling
- 12.5 Disk Management
- 12.6 Swap-Space Management
- 12.7 RAID Structure
- 12.8 Stable-Storage Implementation
- 12.9 Tertiary Storage Devices
- 12.10 Summary





Objectives

- 大容量存储器结构：磁盘结构，传输时间，寻道时间，延迟时间，主机附属存储，网络附属存储NAS，存储区域网络SAN。
- 磁盘调度：调度时机，FCFS算法，SSTF算法，SCAN算法，C-SCAN算法，LOOK算法，C-LOOK算法。
- 磁盘管理：磁盘格式化，主引导块MBR。
- 交换空间管理。
- RAID结构。

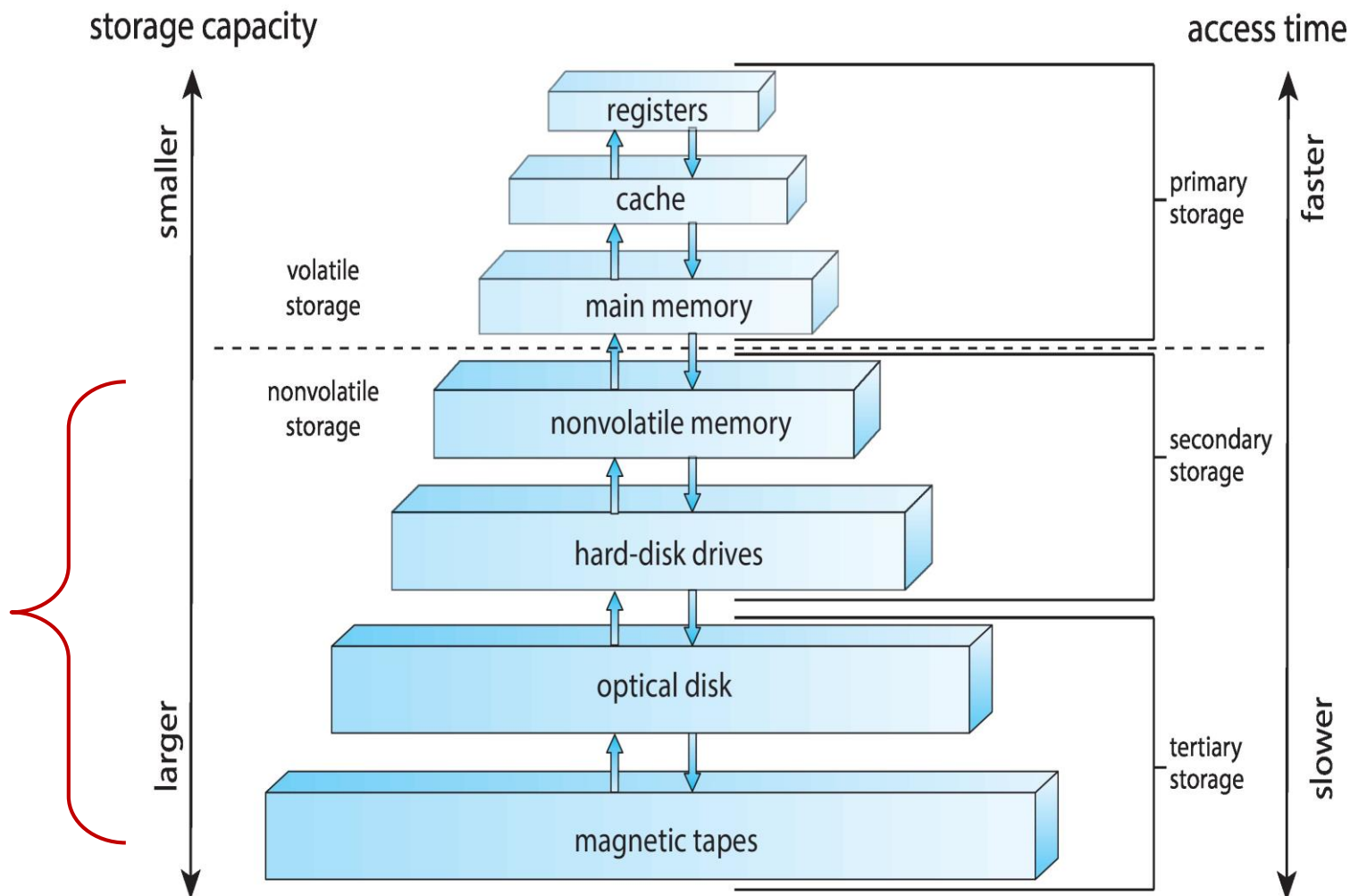




Hierarchical Storage Architecture

数据即资产
数据即权力

大容量
存储器





12.1 Overview of Mass Storage Structure

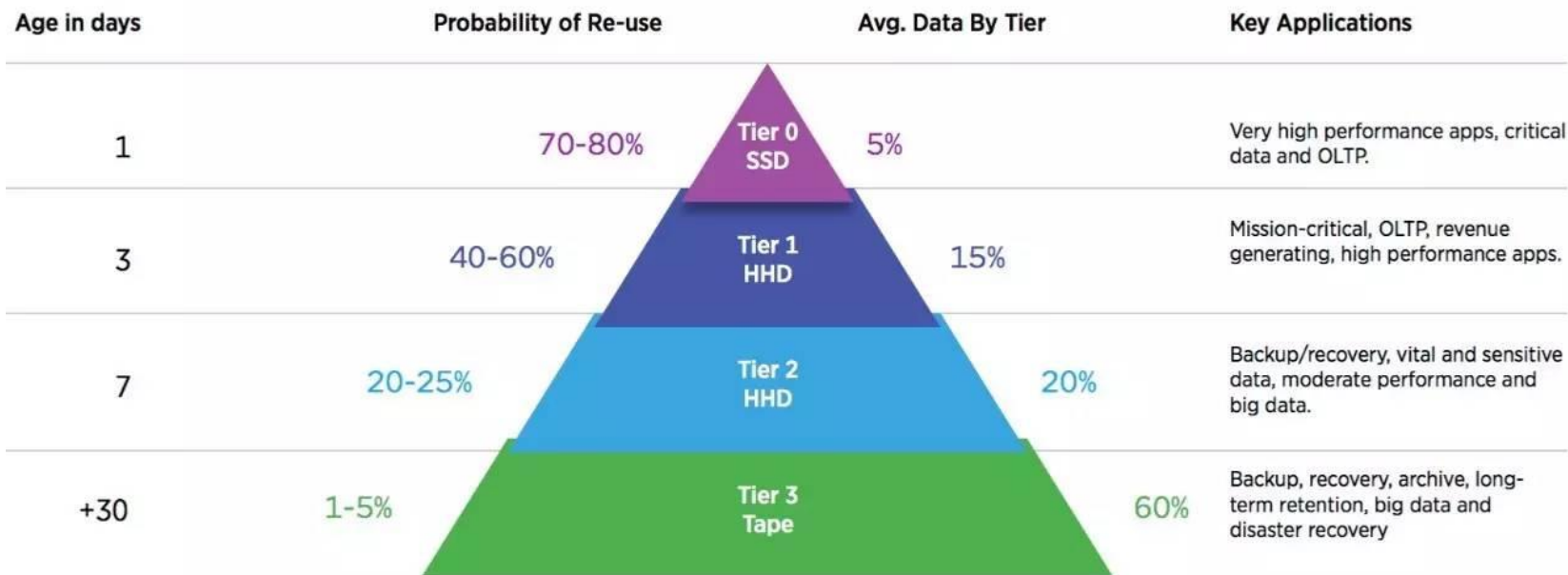




Overview of Mass Storage Structure

■ 分层存储体系

Data Aging Profile



Archive-as-a-Service (AaaS),
Disaster-Recovery-as-a-Service (DRaaS),
Backup Recovery-as-a-Service (BRaaS).





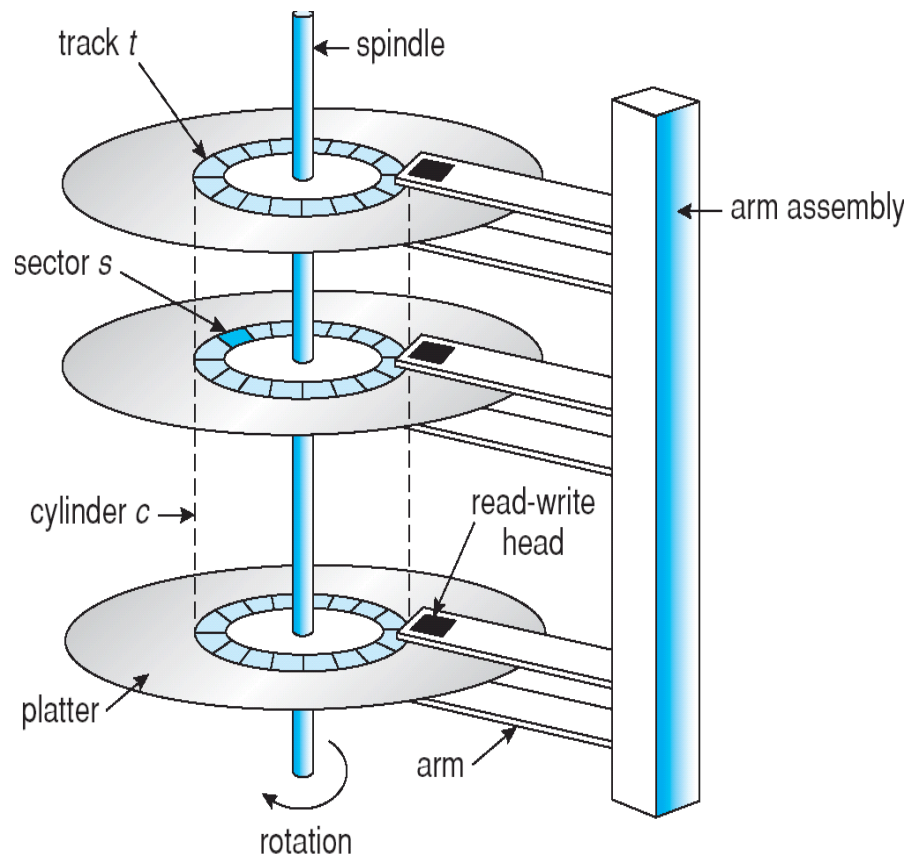
Overview of Mass Storage Structure

- **Magnetic disks (磁盘)** provide bulk of secondary storage of modern computers
 - **Transfer rate** is rate at which data flow between drive and computer
 - **Positioning time (random-access time)** is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
 - **Head crash** results from disk head making contact with the disk surface
 - ▶ That's bad
- **Disks** can be removable
- Drive attached to computer via **I/O bus**
 - Busses vary, including **EIDE, ATA, SATA, USB, Fibre Channel, SCSI**
 - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array



磁盘结构

- 磁盘设备是以一种逻辑块的一维大数组的形式编址的，这里的逻辑块（512B）是传输的最小单位。
- 逻辑块的一维数组映射到磁盘上一些相连的扇区。
 - 0扇区是最外边柱面的第一个磁道的第一个扇区。
 - 数据首先都映射到一个磁道，其余的数据映射到同一柱面的其他磁道，然后按照从外向里的顺序映射到其余的柱面。



Overview of Mass Storage Structure (Cont)

■ **固态驱动器 (Solid State Drives)**，称固态硬盘，固态硬盘用固态电子存储芯片阵列制成的硬盘，由控制单元和存储单元（**FLASH芯片**、**DRAM芯片**）组成。



- 第一只SSD出现在1978年（STK 4305，每MB售价8800美元，DRAM）。
- 全闪存阵列（AFAS）和混合闪存阵列（HFA）呈爆发式增长。
- 现在SSD的容量超过15TB。
- 非易失性、低功耗（只有HDD的三分之一）。
- 无活动部件、可靠性高——位误码率（BER） 1×10^{17}
- 读取存取时间：**500MB/s ~ 4000MB/s**，存取时间比HDD大概快 100倍。



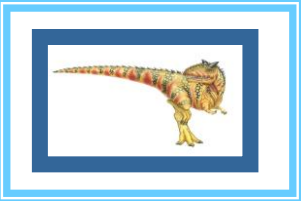
■ Magnetic tape (磁带)

- 出货的磁带驱动器中超过85%是LT0 (Linear Tape Open) 。
- 磁带驱动器的可靠性、数据传输速率和容量已超过磁盘。
- 磁带的原生容量超过**10TB**，压缩容量超过**25TB**。(LT0-10:48TB)
- 磁带的原生数据传输速率为360MB/s。
- **LTFS(Liner Tape File System)**为磁带提供了一种通用、开放的文件系统。
- 由于总体拥有成本，云采用磁带解决方案用于归档服务。
- 对企业级磁带和LT0而言，磁带介质的寿命至少是30年。





12.3 Disk(外存) Attachment





Disk Attachment

- Disks may be attached one of two ways:
 1. Host attached via an I/O port
 2. Network attached via a network connection
- 当前三种方式:
 - **DAS** (Direct(Host-) Attached Storage)
 - **NAS** (Network Attached Storage 网络附加存储)
 - **SAN** (Storage-Area Network 存储区域网)





Host-attached storage

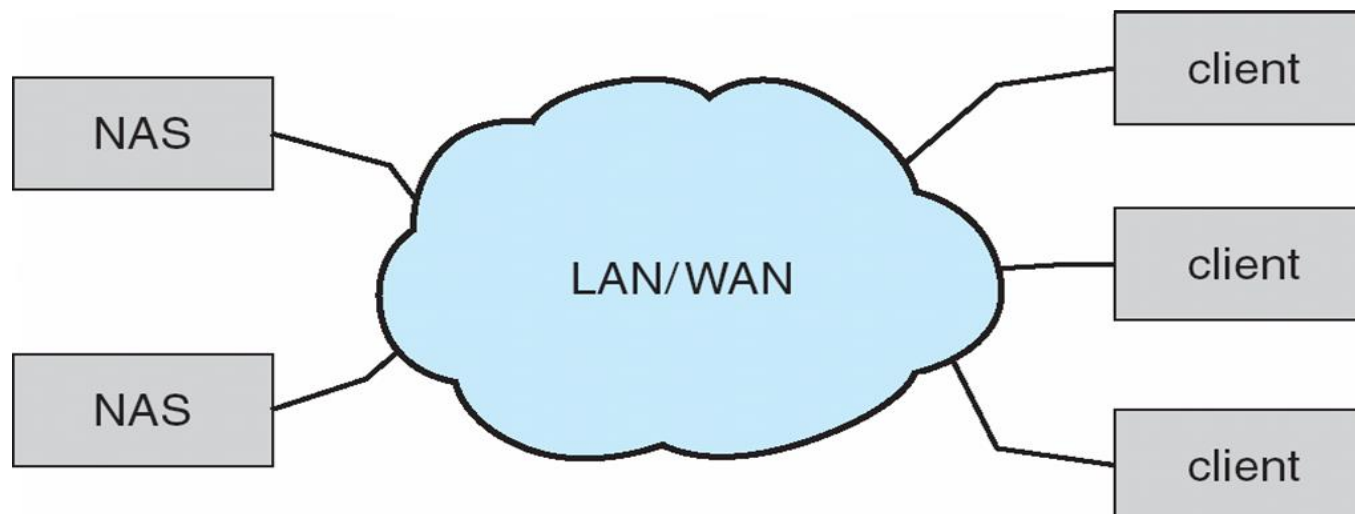
- **Host-attached storage** accessed through I/O ports talking to I/O busses
- I/O bus like **IDE**
 - a maximum of 2 drives per I/O bus
- **SCSI** itself is a bus, up to 16 devices on one cable, **SCSI initiator** requests operation and **SCSI targets** perform tasks
 - Each target can have up to 8 **logical units** (disks attached to device controller)
- **FC** (Fibre Channel, 光纤通道) is high-speed serial architecture
 - Can be switched fabric with 24-bit address space – the basis of **storage area networks (SANs)** in which many hosts attach to many storage units
 - Can be **arbitrated loop (FC-AL)** of 126 devices





Network-Attached Storage

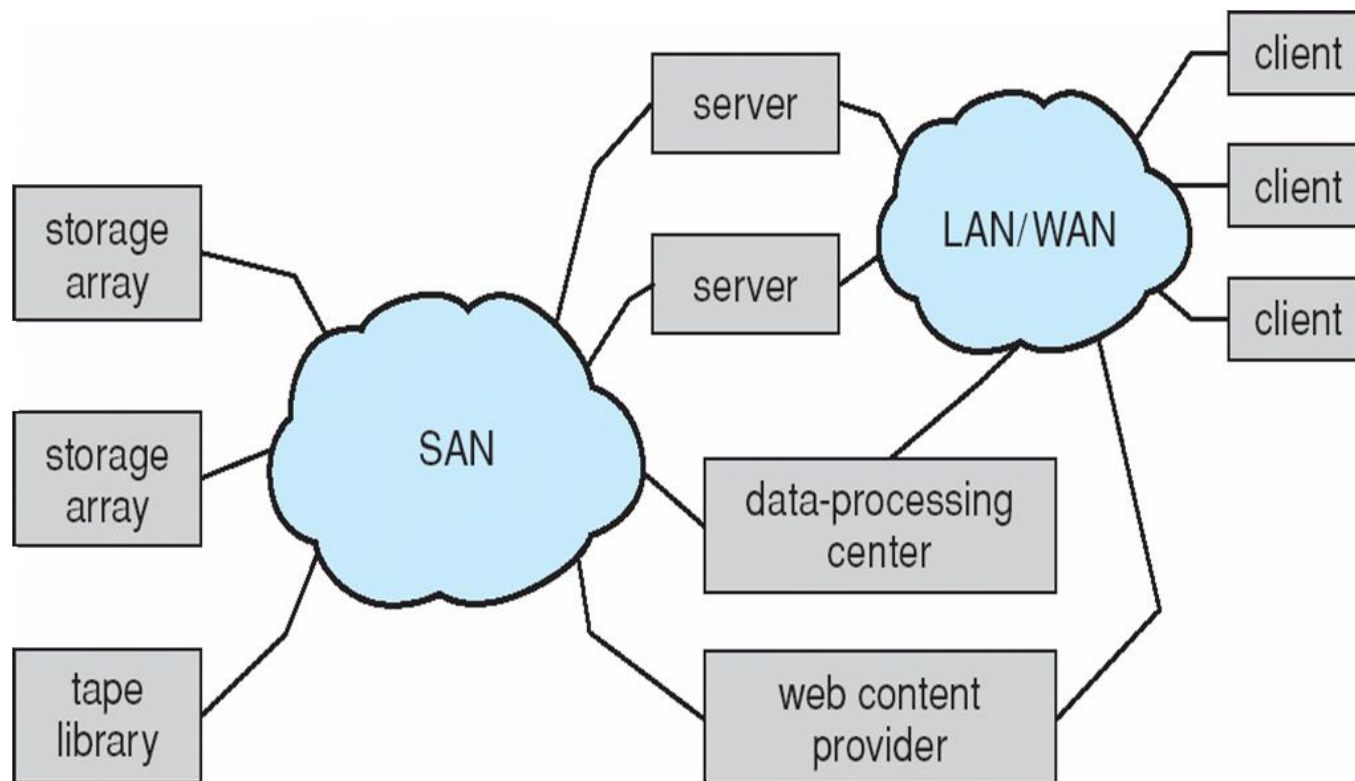
- **Network-attached storage (NAS)** is storage made available over a network rather than over a local connection (such as a bus)
- NFS and CIFS are common protocols
- Implemented via remote procedure calls (RPCs) between host and storage
- New **iSCSI** protocol uses IP network to carry the SCSI protocol





Storage Area Network(SAN)

- Common in large storage environments (and becoming more common)
- Multiple hosts attached to multiple storage arrays - flexible



■ **SNIA** (Storage Networking Industry Association, 存储网络联合会) 官方对于Virtualization (存储虚拟化技术) 的定义, 如下:

- 是将存储(子)系统内部功能与具体应用、主机及通用网络资源分离、隐藏及抽象的行为。以期达到存储或数据管理的网络无关性。
- 对于存储服务及设备的虚拟化应用, 以期达到整合设备功能、隐藏复杂细节以及向已经存在的底层存储资源添加新的应用。

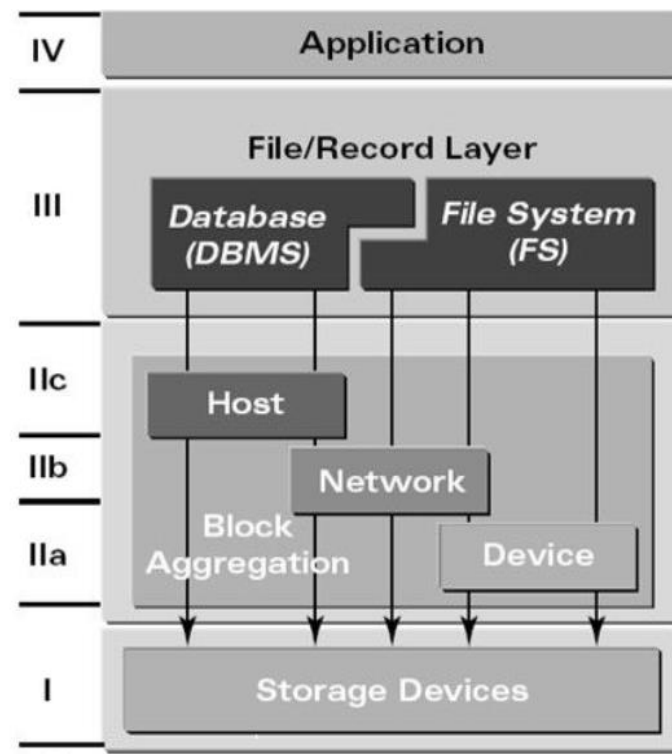


图 1 SNIA 共享存储模型





12.4 Disk Scheduling (注：传统机械式磁盘)





Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
- Access time has three major components
 - **Seek time** (寻道时间) is the time for the disk are to move the heads to the cylinder containing the desired sector
 - **Rotational latency** (旋转延迟) is the additional time waiting for the disk to rotate the desired sector to the disk head
 - **Transfer time** (传输时间)
- Minimize seek time
 - **Seek time \approx seek distance** 寻道时间 \approx 寻道距离
- Disk **bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer





数据计算

- 7200(转 / 每分钟)的硬盘，每旋转一周所需时间为 $60 \times 1000 \text{ (毫秒)} \div 7200 = 8.33 \text{ 毫秒}$ ，
则平均旋转延迟时间为 $8.33 \div 2 = 4.17 \text{ 毫秒}$ (平均情况下，需要旋转半圈)。
- 7200转机械硬盘的寻道时间一般为12-14毫秒，固态硬盘可以达到0.1毫秒甚至更低。
- 固态硬盘持续读写速度超过500MB/s
- 机械硬盘读写速度超过50~200MB/s (接口不同)
- 磁带的原生数据传输速率为360MB/s。





Disk Scheduling (Cont)

- Several algorithms exist to schedule the servicing of disk I/O requests
- 常用的磁盘调度算法有：先来先服务(FCFS)、最短寻道时间优先(SSTF)、扫描(SCAN)算法和循环扫描(C-SCAN)算法等

- We illustrate them with a request queue (0-199)

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53

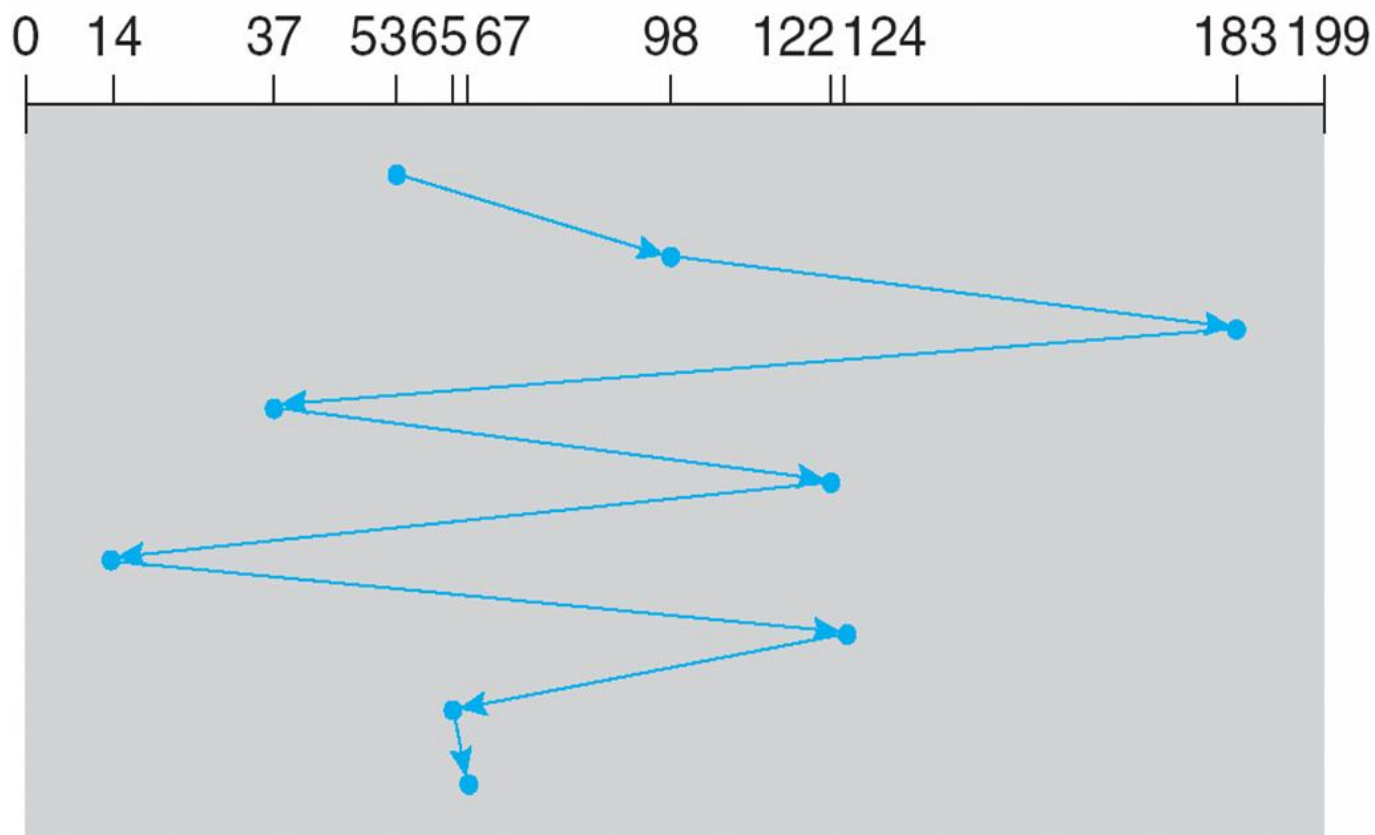




1. FCFS 先来先服务

- **基本思想：**根据进程请求访问磁盘的先后次序进行调度。
- Illustration shows total head movement of **640** cylinders

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53





2、最短寻道时间优先SSTF调度

- **基本思想：** 选择从当前磁头位置所需寻道时间最短的请求。
- **SSTF**(Shortest Seek Time First)是SJF调度的一种形式；有可能引起某些请求的饥饿。
- 如图所示，磁头移动的总距离是236柱面。

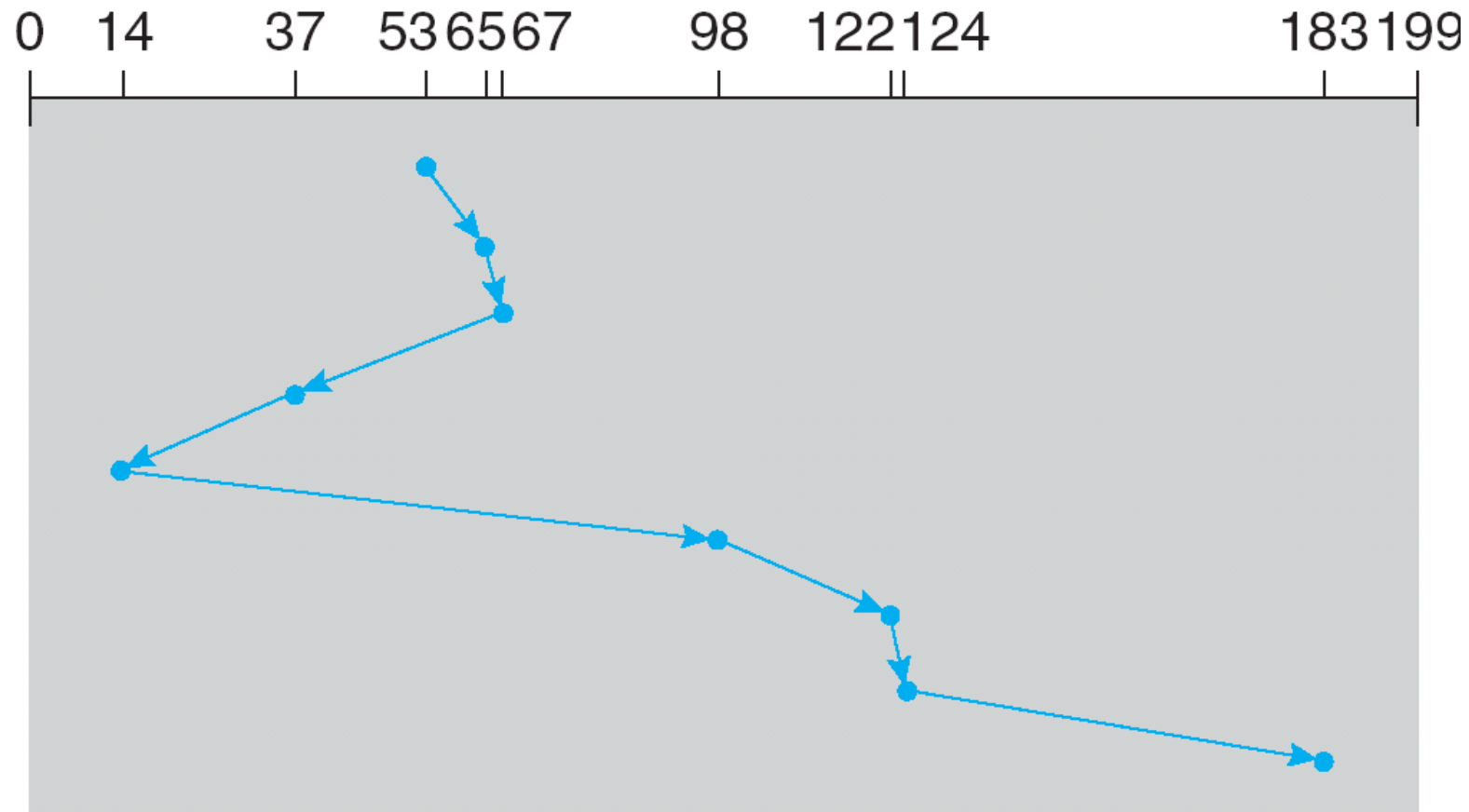




SSTF (Cont)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



total head movement of 236 cylinders





3、扫描调度 SCAN

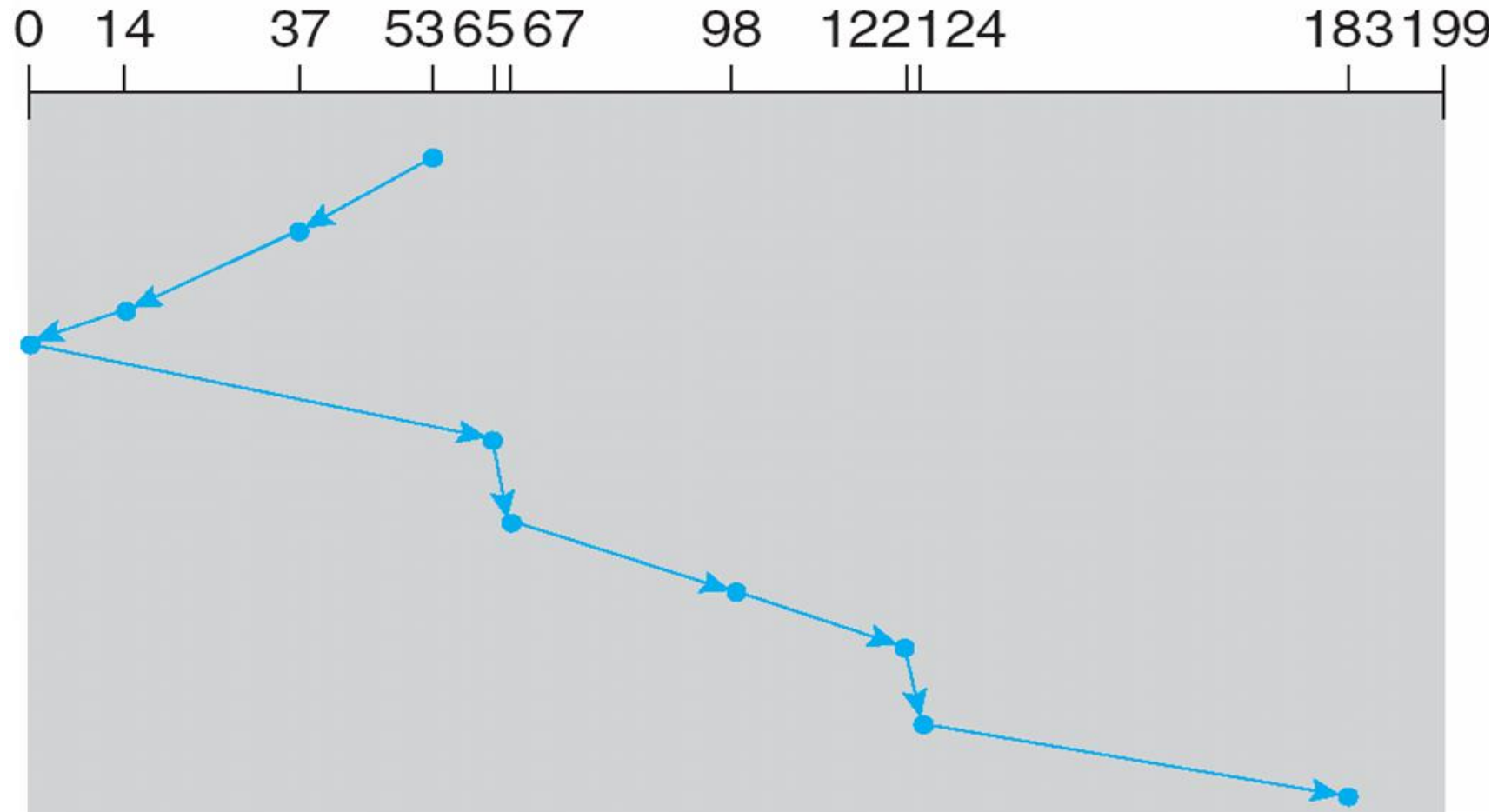
- **基本思想**：磁头从磁盘的一端开始向另一端移动，沿途响应访问请求，直到到达了磁盘的另一端，此时磁头反向移动并继续响应服务请求。
- 也称为**电梯算法** *elevator algorithm*。
- 如图所示，磁头移动的总距离是**236**柱面。
- **LOOK**:208





SCAN (Cont.)

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53





4、C-SCAN Scheduling

- 提供比扫描算法更均衡的等待时间。
- **基本思想**：磁头从磁盘的一段向另一端移动，沿途响应请求。当它到了另一端，就立即回到磁盘的开始处，在返回的途中不响应任何请求。
- 把所有柱面看成一个循环的序列，最后一个柱面接续第一个柱面。

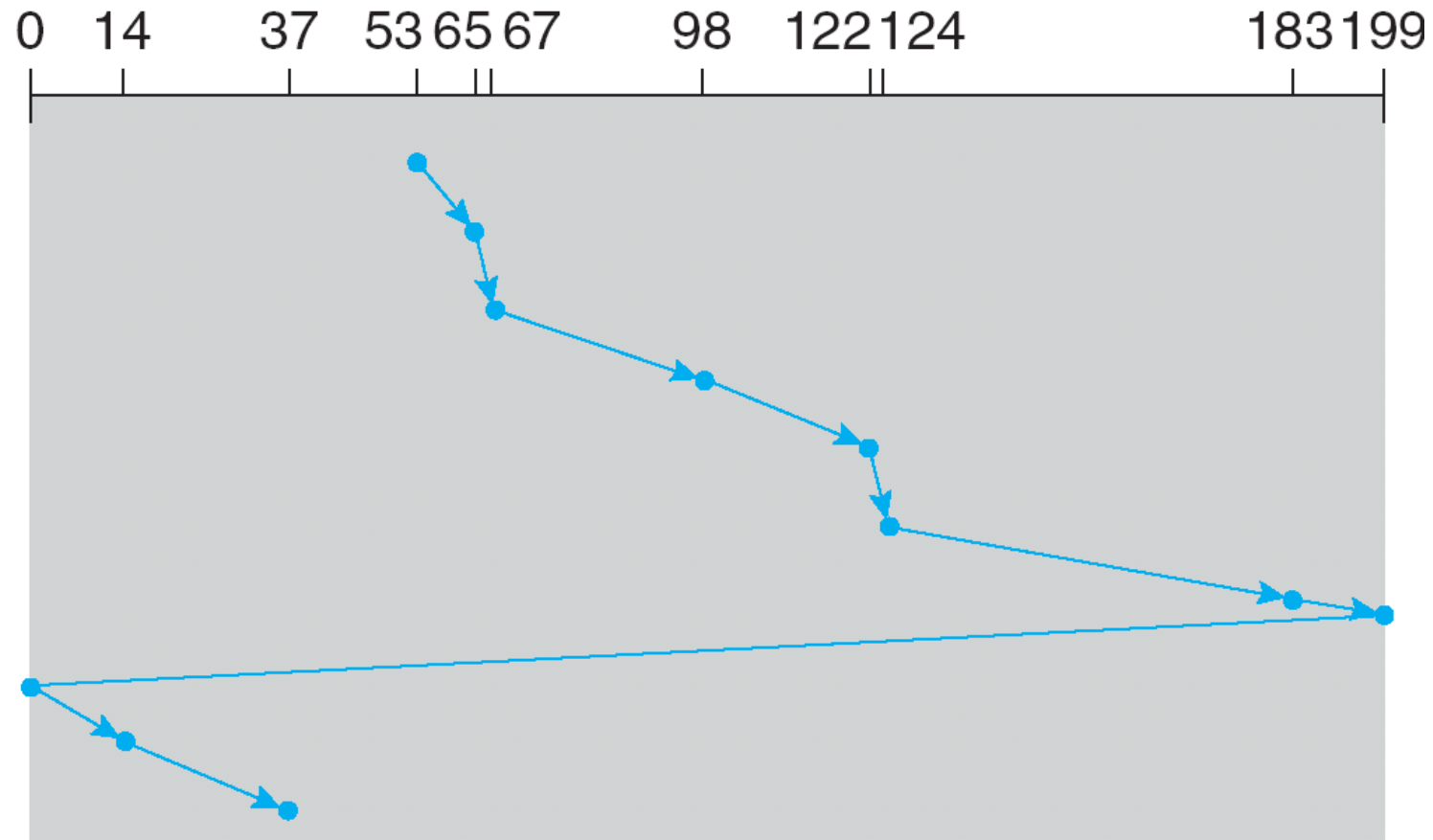




C-SCAN (Cont)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53





5、LOOK、 C-LOOK Scheduling

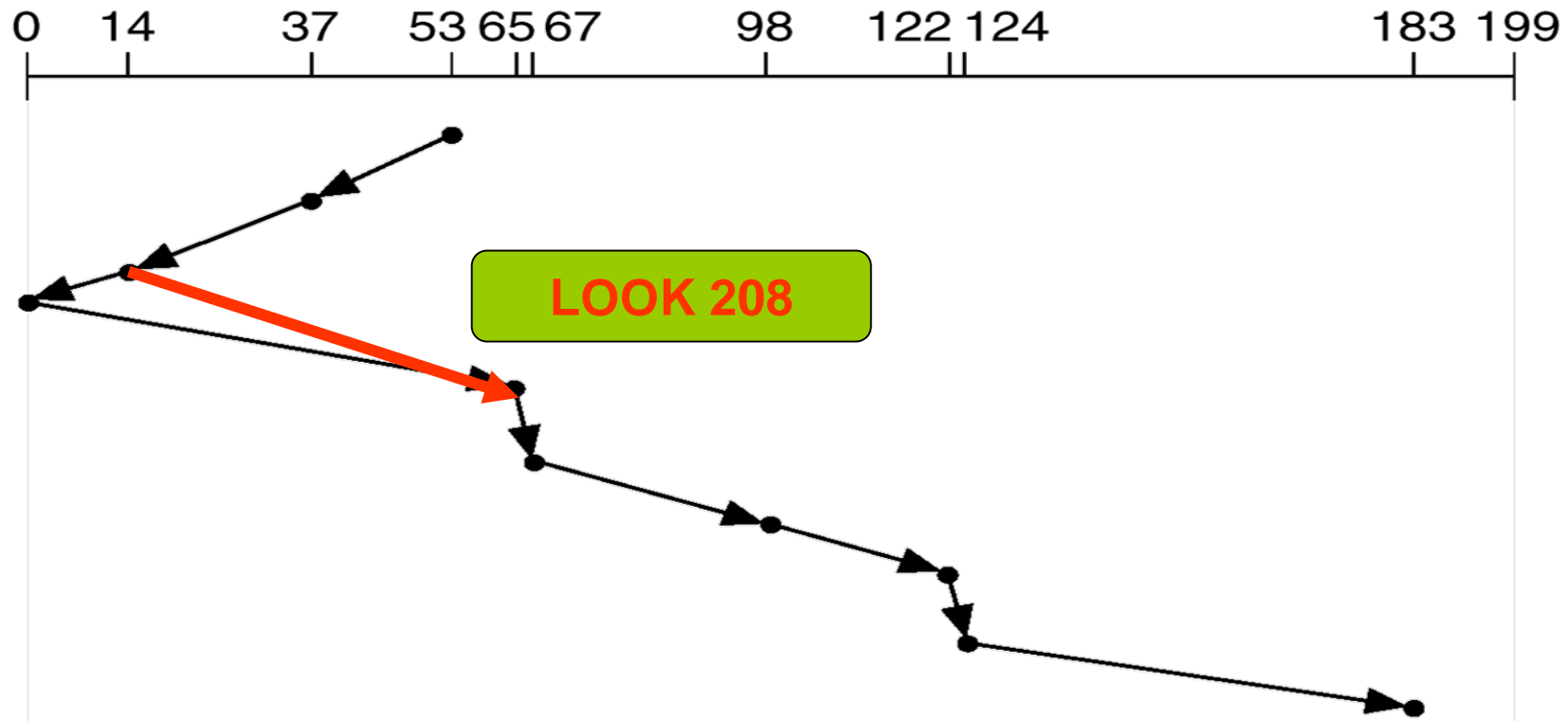
- **LOOK**--- SCAN算法的一种形式。
- **C-LOOK**-C-SCAN算法的一种形式。
- 磁臂在每个方向上仅仅移动到最远的请求位置，然后立即反向移动，而不需要移动到磁盘的一端。





LOOK-- Version of SCAN

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



total head movement of **208** cylinders

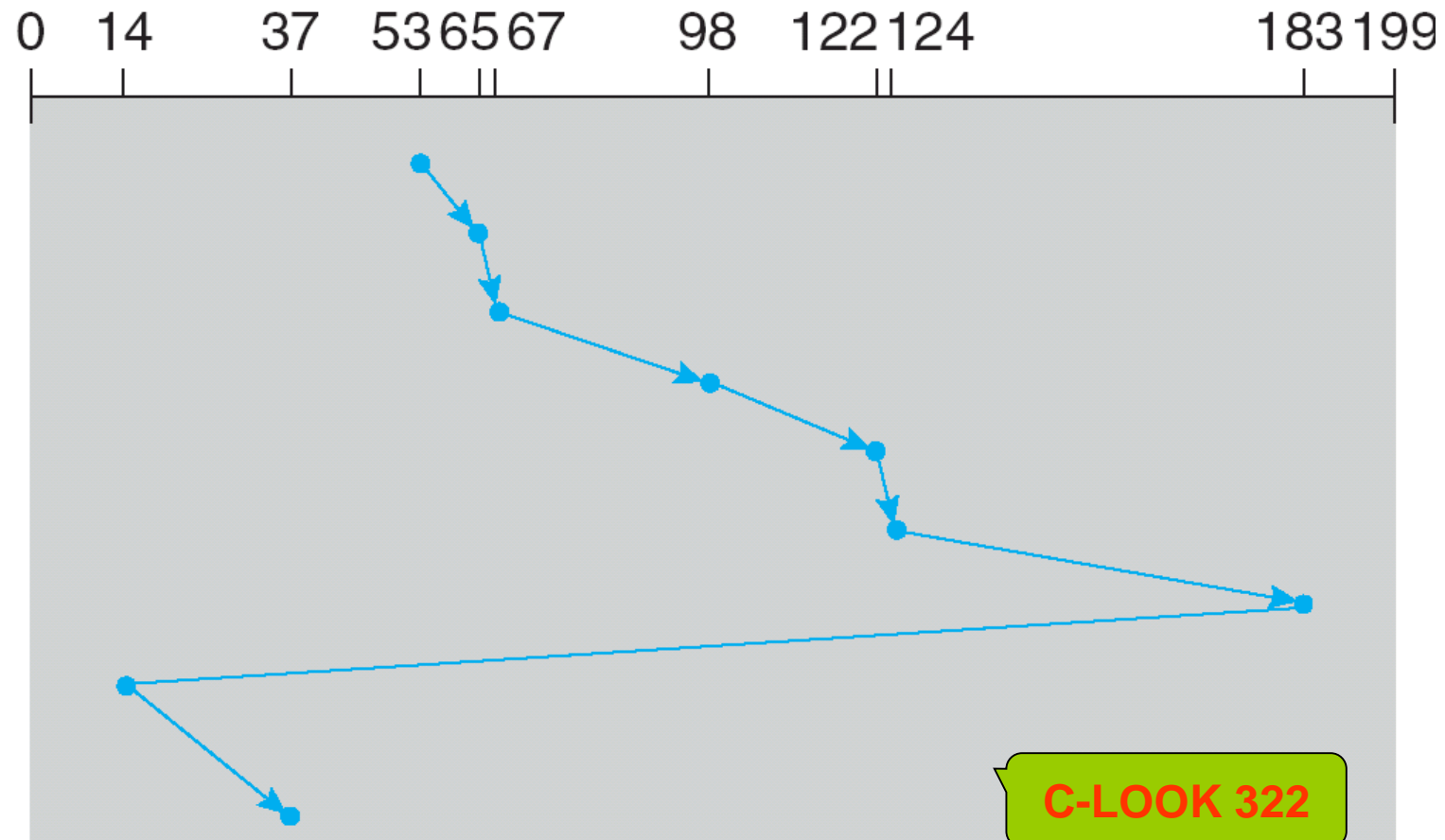




C-LOOK (Cont)

queue 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53





Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm





12.5 Disk Management





磁盘格式化 *Disk Format*

- 低级格式化，或物理格式化 (*Low-level formatting*, or *physical formatting*) —— 把磁盘划分成扇区，以便磁盘控制器可以进行读写。
- 分区 (*Partition*)：把磁盘划分成一个或多个柱面组。
- 逻辑格式化或“创建文件系统” *Logical formatting* or “*making a file system*”。





■ 启动块Boot Block

- 启动块初始化系统
 - ▶ 引导 (bootstrap自举) 程序存储在ROM中
 - ▶ 引导程序装载程序。
- Fig 13.6 MS-DOS Disk Layout

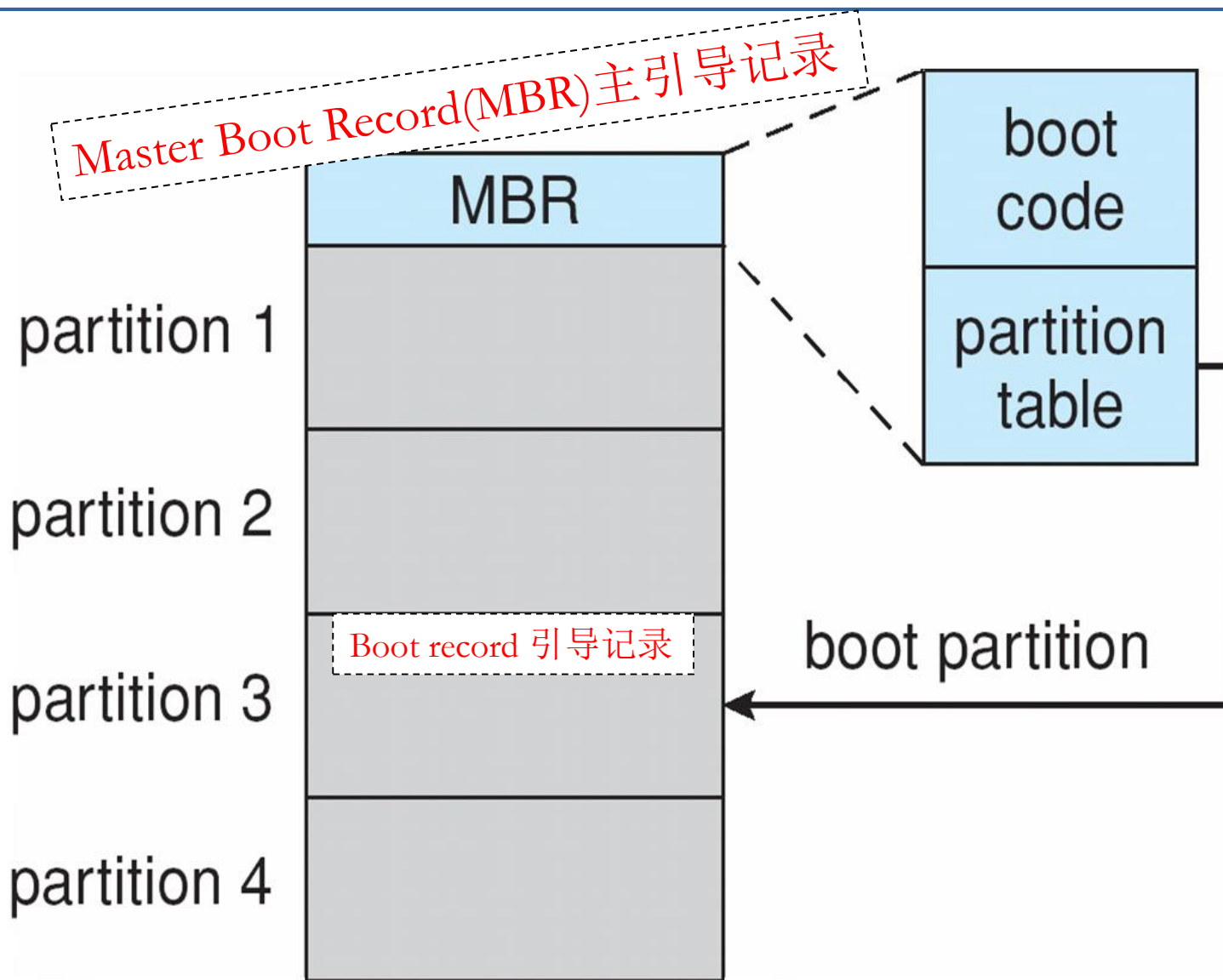
■ 坏块Bad Block

- 坏块的处理方法
- MS-DOS的处理方法: format,chkdsk命令





Booting from a Disk in Windows





12.6 Swap-Space Management





Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
- Swap-space can be carried out in two forms:
 - in the **normal file system**
e.g. **Windows** family
 - in a **separate disk partition**
e.g. **Linux**、**Unix**、solaris

pagefile.sys文件

SWAP分区





交换空间管理

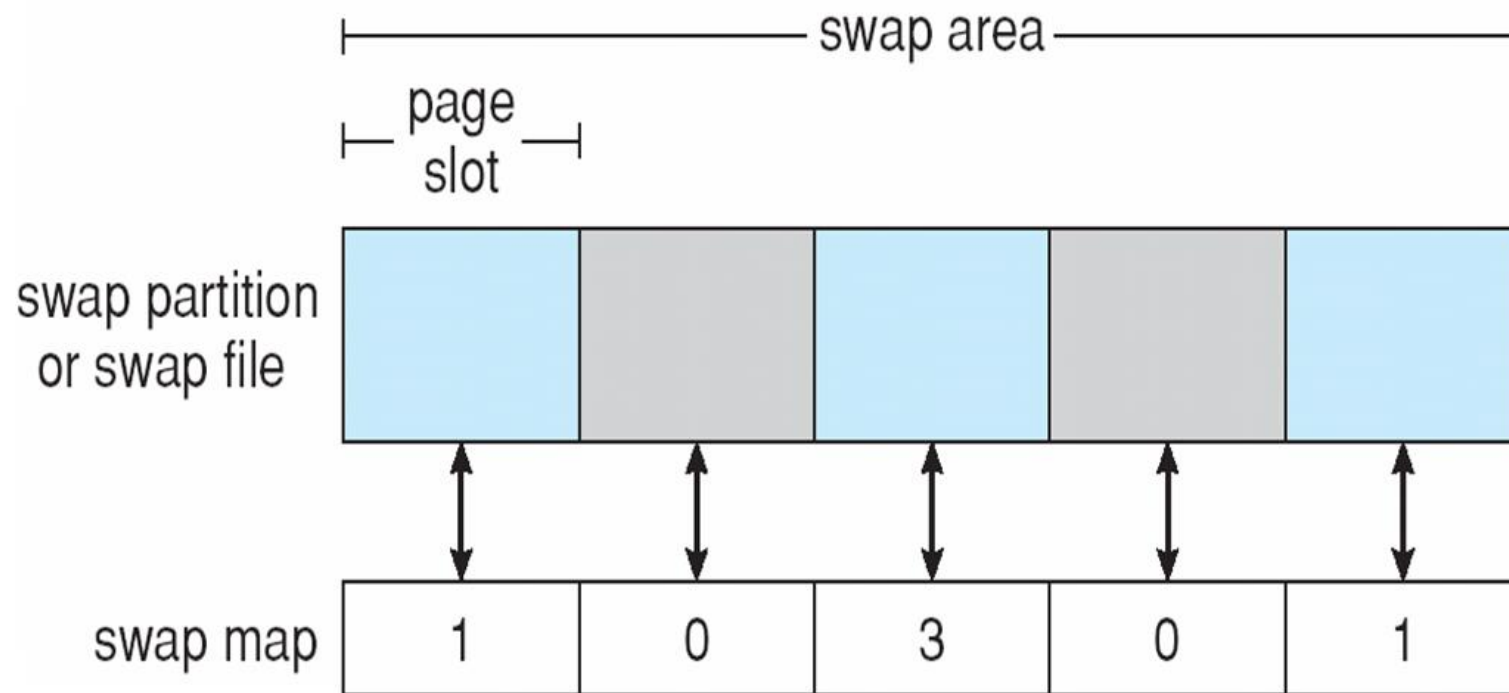
■ 交换空间的位置

- 交换空间在普通文件系统上加以创建。通常是文件系统内的一个简单大文件（如Windows: **pagefile.sys文件**）。这种方式实现简单但效率较低。（外部碎片）
- 交换空间创建在独立的**磁盘分区上**（如Unix/Linux）。（内部碎片）
- 有些OS较为灵活，可以由系统管理员来选择使用以上哪种方式。





Data Structures for Swapping on Linux Systems





12.7 RAID Structure





RAID Structure

- **RAID** : Redundant Arrays of Inexpensive (independent) Disks (**冗余廉价磁盘阵列**) . RAID是一种把多块独立的硬盘（物理硬盘）按不同的方式组合起来形成一个硬盘组（逻辑硬盘），从而提供比单个硬盘更高的存储性能和提供数据备份技术。
- **Inexpensive -> Independent**
- RAID – multiple disk drives provides reliability via **redundancy**
- Increases the **mean time to failure**
- Frequently combined with **NVRAM** to improve write performance
- RAID is arranged into **six different levels**（较早），RAID 7、10、53、5E、5EE





RAID (Cont)

- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively
- Disk **striping** (条带化) uses a group of disks as one storage unit
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
 - **Mirroring** (镜像) or **shadowing** (RAID 1) keeps duplicate of each disk
 - Striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides high performance and high reliability
 - **Block interleaved parity** (RAID 4, 5, 6) uses much less redundancy





RAID (Cont)

- RAID within a storage array can still fail if the array fails, so automatic replication of the data between arrays is common
- Frequently, a small number of hot-spare disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them



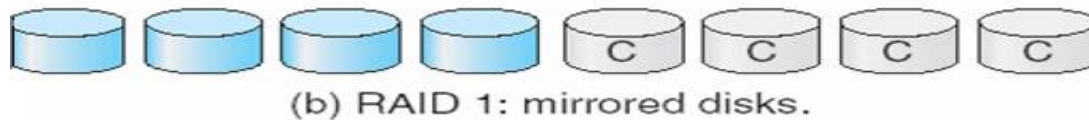


RAID Levels



P:纠错位

C:数据的第二拷贝



RAID2、3按字节
或位striping

Hamming码



奇偶校验

按块striping



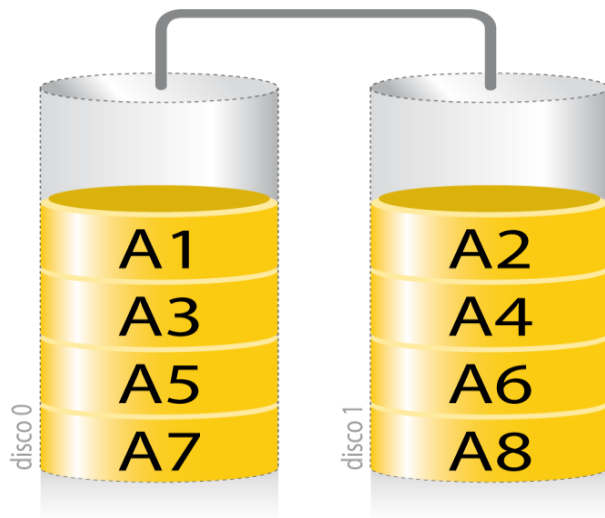
P+Q冗余，差错
纠正码



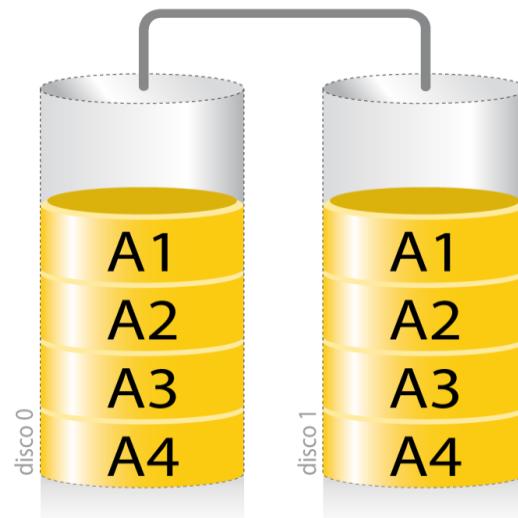


RAID Levels

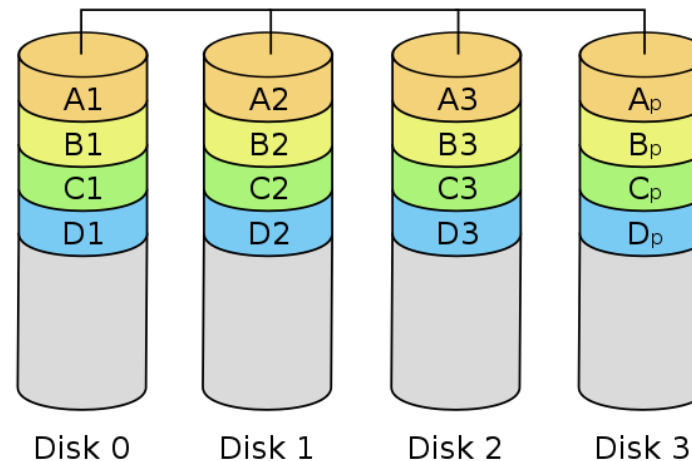
RAID 0



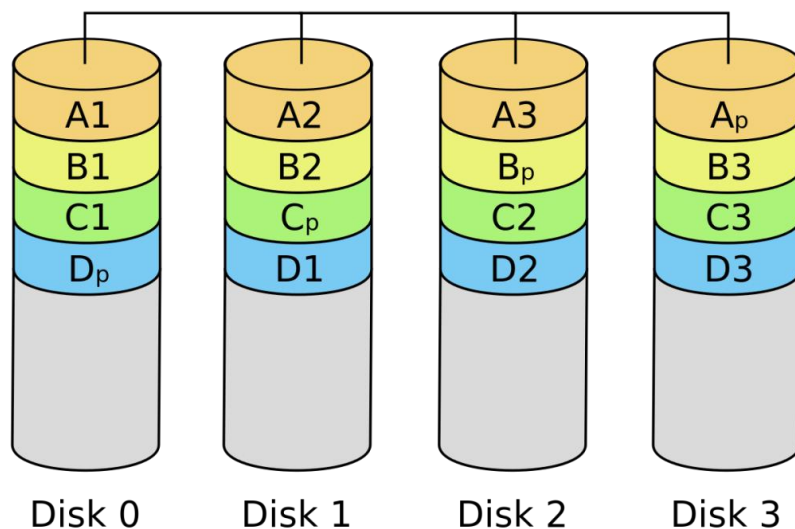
RAID 1



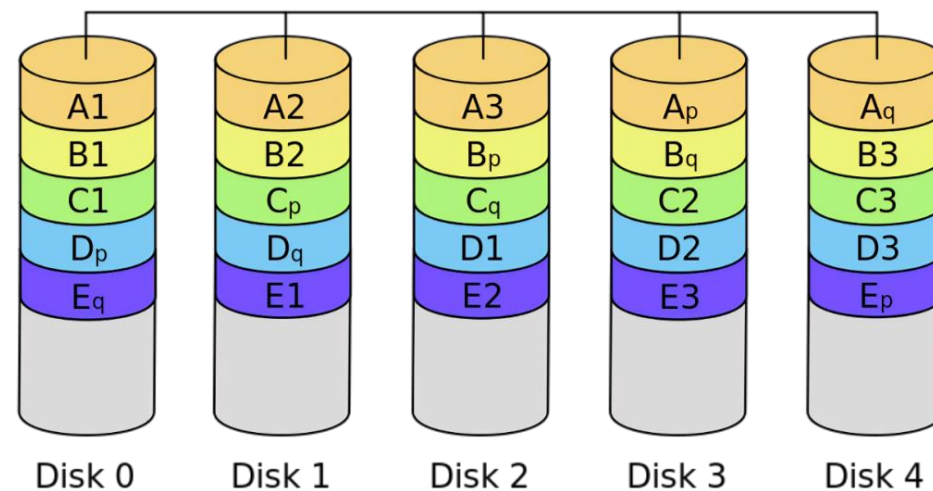
RAID 4



RAID 5



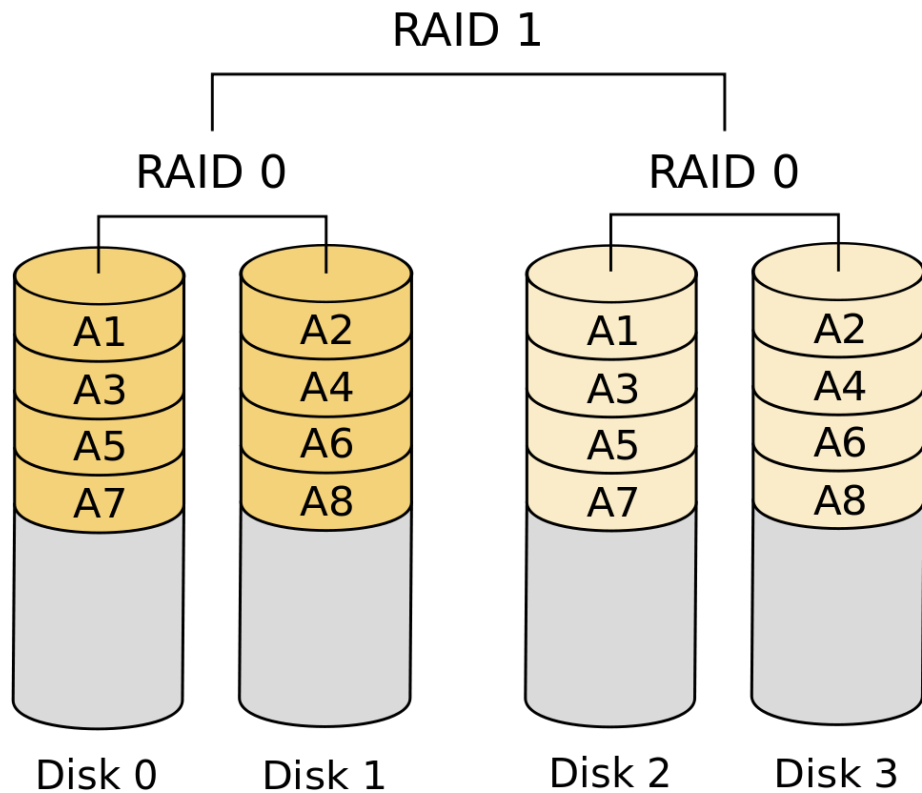
RAID 6



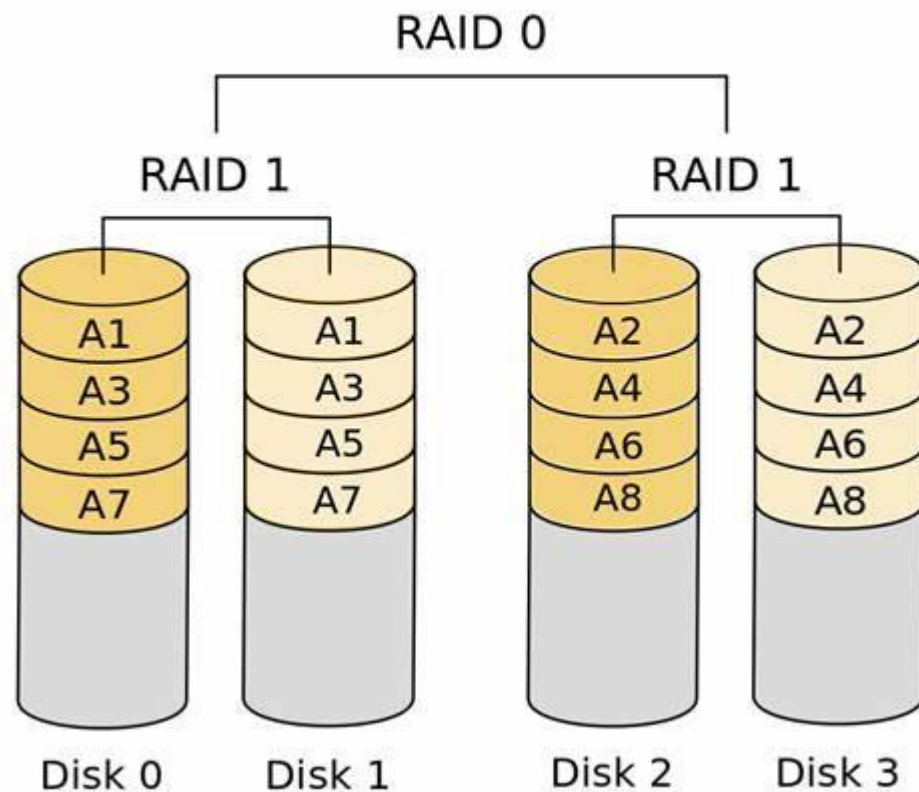


RAID Levels

RAID 0+1



RAID 1+0





Review Questions

- 有哪些磁盘调度算法？
- 磁盘的逻辑格式化作用是什么？
- 什么时候使用交换空间？
- RAID的作用是什么？





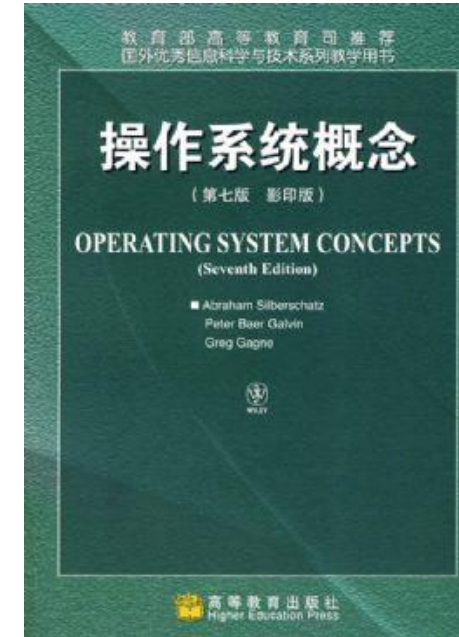
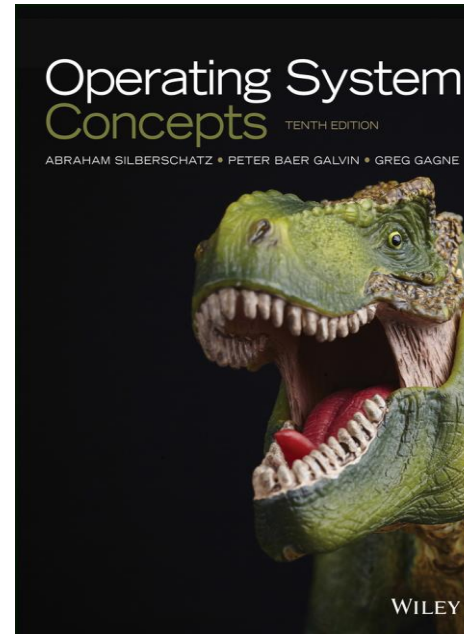
Reading Assignments

■ Read for this week:

- Chapters 12
of the text book:

■ Read for next week:

- Chapters 13
of the text book:





End of Chapter 12

