Chapter 12 Mass-Storage Structure

LI Wensheng, SCS, BUPT

Strong points:

- **Disk Structure**
- **Disk Scheduling**
- **Disk Management**

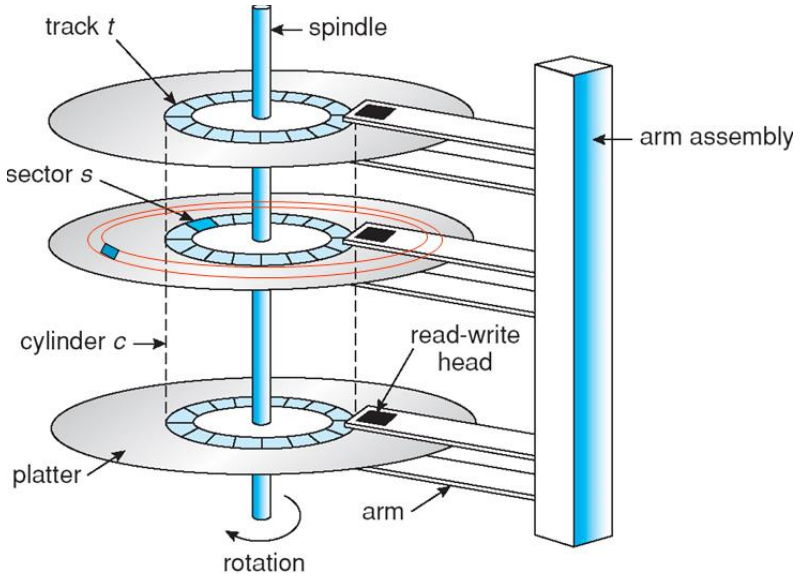
Chapter Objectives

- **Describe the physical structure of disk and disk scheduling algorithms**
- **Describe the policies of disk management**

Contents

12.1 Overview of Mass Storage Structure

- **Magnetic disks provide bulk of secondary storage of modern computers**
- **Hard Disks**
  - **Platters diameters: 3.5", 2.5", 1.8".**
  - **Capacity: from 30GB to 3TB.**
  - **Transfer Rate: theoretical, 6 Gb/sec real: 1 Gb/sec.**
  - **Seek time: from 3ms to 12ms. 9ms common for desktop drives.**
  - **Average seek time measured or calculated based on 1/3 of tracks.**
  - **Latency based on spindle speed. Rotate 60 to 250 times per second. Average latency=1/2 latency.**



Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

Hard Disk Performance

- **Transfer rate** is rate at which data flow between drive and computer.
- **Positioning time (random-access time):**
  - **seek time,** the time to move disk arm to desired cylinder.
  - **rotational latency,** the time for desired sector to rotate under the disk head.
- **Access Latency** = **Average access time**
  = average seek time + average latency
  - For fastest disk, 3ms + 2ms = 5ms
  - For slow disk, 9ms + 5.56ms = 14.56ms

Hard Disk Performance (Cont.)

Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

- **Average I/O time**
  = average access time +
  (amount to transfer / transfer rate) +
  controller overhead.
- **E.g. Transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a 0.1ms controller overhead.**
  - **Average I/O time = 5ms + 4.17ms + transfer time + 0.1ms**
    = 9.27 ms + transfer time
  - **transfer time = 4KB / 1Gb/s =4KB/(1/8*1024^{2}KB/s)**
    = 32 / (1024^2) s = 0.031 ms
  - **Average I/O time for 4KB block**
    = 9.27ms +0 .031ms = 9.301ms

Overview of Mass Storage Structure(Cont.)

- **Head crash** results from disk head making contact with the disk surface.
  - Cannot be repaired, the entire disk must be replaced.
- Disks can be removable
  - generally consist of one platter, held in a plastic case to prevent damage while not in the disk drive.
  - Other forms: CDs, DVDs, and Blu-ray discs, flash drives.
- A disk drive is attached to a computer via **I/O bus.**
  - advanced technology attachment (**ATA**), serial ATA (**SATA**), External Serial ATA (**eSATA**).
  - universal serial bus (**USB**)
  - fiber channel (**FC**)

Overview of Mass Storage Structure(Cont.)

- **Host controller in computer uses bus to talk to disk controller built into drive or storage array.**
  E.g. To perform a disk I/O operation:
  - **The computer places a command into the host controller, typically using memory-mapped I/O ports.**
  - **The host controller sends the command via messages to the disk controller.**
  - **The disk controller operates the disk-drive hardware to carry out the command.**
  - **Disk controllers have a built-in cache.**
    - **Data transfer at the disk drive happens between the cache and the disk surface.**
    - **Data transfer to the host, at fast electronic speeds, occurs between the cache and the host controller.**

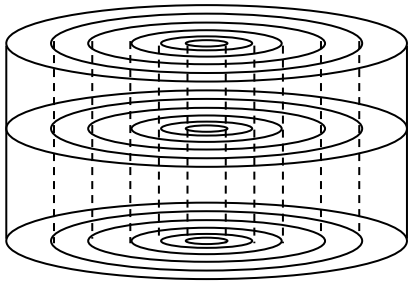
* Solid-State Disks (SSD)

- Nonvolatile memory used like a hard drive
- Can be more reliable than HDDs
  - No moving parts
- Faster than HDDs
  - no seek time or rotational latency
- More expensive per MB
- Consume less power
- Less capacity
- Maybe have shorter life span
- Used in
  - storage arrays, where they hold file-system metadata that require high performance.
  - some laptop computers to make them smaller, faster, and more energy-efficient.
- Standard bus can be too slow -->
  SSDs are designed to connect directly to PCI for example.

Magnetic tape

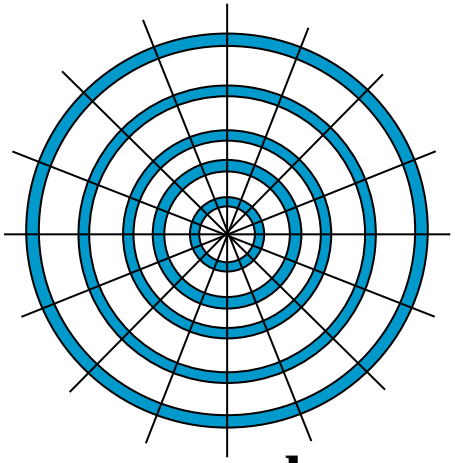
- Was early secondary-storage medium
- Relatively permanent and holds large quantities of data
- Access time slow
- Random access ~1000 times slower than disk
- Mainly used for backup, storage of infrequently-used data, transfer medium between systems
- Kept in spool and wound or rewound past read-write head
- Once data under head, transfer rates comparable to disk
  - 140MB/sec and greater
- 200GB to 1.5TB typical storage
- Categorized by width, including 4, 8, and 19 millimeters and 1/4 and 1/2 inch.
- Named according to technologies, LTO-5, SDLT.
  - Linear Tape Open
  - Super digital linear tape

12.2 Disk Structure



- Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer.
  - Low-level formatting creates logical blocks on physical media.
  - block size is usually 512 bytes.
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.
  - Sector 0 is the first sector of the first track on the outermost cylinder.
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost.

Disk Structure (Cont.)



- Mapping logical to physical address should be easy
  - Except for bad sectors
  - Non-constant # of sectors per track via constant angular velocity
- Constant Linear Velocity (常数线性周转率)
  - the density of bits per track is uniform.
  - Tracks in the outermost zone typically hold 40 percent more sectors than do tracks in the innermost zone.
  - CD-ROM, DVD-ROM
- Constant Angular Velocity (常数角周转率)
  - the density of bits decreases from inner tracks to outer tracks.
  - hard disk

12.3 Disk Attachment

- Computer access disk storage in two ways
  - Via I/O ports (host-attached storage)
  - Via a remote host in a distributed file system (NAS)

Host-attached storage

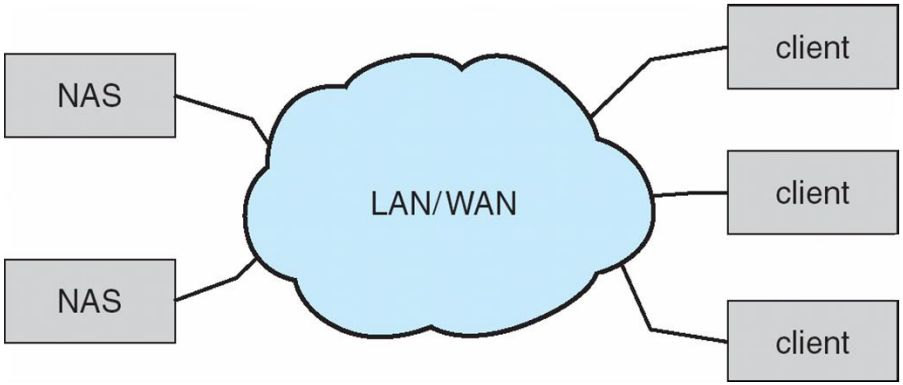
- Is accessed through I/O ports talking to I/O busses
- I/O bus architecture
  - IDE(Integrated Drive Electronics), ATA(Advanced Technology Attachment)
    - support maximum of 2 drives per I/O bus, Desktop PC
  - SATA (Serial ATA), similar protocol, but simplified cabling.
  - SCSI (Small Computer System Interface), itself is a bus
    - up to 16 devices on one cable.
    - SCSI initiator (controller card in the host) requests operation and SCSI targets (storage devices) perform tasks
    - A SCSI disk is a common SCSI target
    - Each target can have up to 8 logical units (disks attached to device controller)

Host-attached storage (Cont.)

- I/O bus architecture (Cont.)
  - FC (Fibre Channel), is a high-speed serial architecture, can operate over optical fiber or over a 4-conductor copper cable, two variants:
    - switched fabric with 24-bit address space – the basis of storage area networks (SANs) in which many hosts attach to many storage units
    - arbitrated loop (FC-AL，Fibre Channel Arbitrated Loop) of 126 devices
- I/O directed to bus ID, device ID, logical unit (LUN)

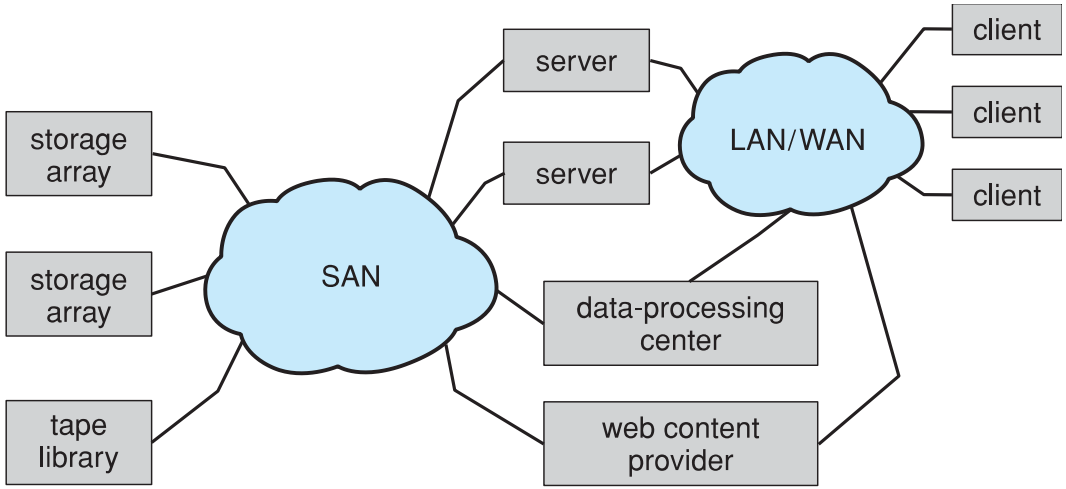
Network-Attached Storage (NAS)

- NAS is a special-purpose storage, accessed remotely over a data network.
- Clients access NAS via a RPC interface between host and storage. NFS and CIFS (Common Internet File System) are common protocols.
- RPCs are carried via TCP or UDP on IP network.
- The NAS unit is usually implemented as a RAID array with software that implements the RPC interface.
- New iSCSI protocol uses IP network protocol to carry the SCSI protocol. Remotely attaching to devices (blocks)



Storage Area Network (SAN)

- Private network connecting servers and storage units, using storage protocols.
- Common in large storage environments.
- Multiple hosts attached to multiple storage arrays – flexible
- FC is the most common SAN interconnect.
  - The simplicity of iSCSI is increasing its use.
  - InfiniBand, a special-purpose bus architecture
    - provides hardware and software support for high-speed interconnection networks for servers and storage units.



12.4 Disk Scheduling

- The OS is responsible for using hardware efficiently.
  - for disk drives, means having a fast access time and disk bandwidth.
- To read or write, the disk head must be positioned at the desired track and at the beginning of the desired sector.
- Access time has two major components:
  - *Seek time* is the time for the disk arm to move the heads to the cylinder containing the desired sector.
  - *Rotational latency* is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time ≈ seek distance
- 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.

Disk Scheduling (Cont.)

- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer.
- OS maintains queue of requests, per disk or device.
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms only make sense when a queue exists

Disk Scheduling (Cont.)

- Disk drive controllers have small buffers and can manage a queue of I/O requests.
- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue with requests for I/O to blocks on cylinders (0-199):

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

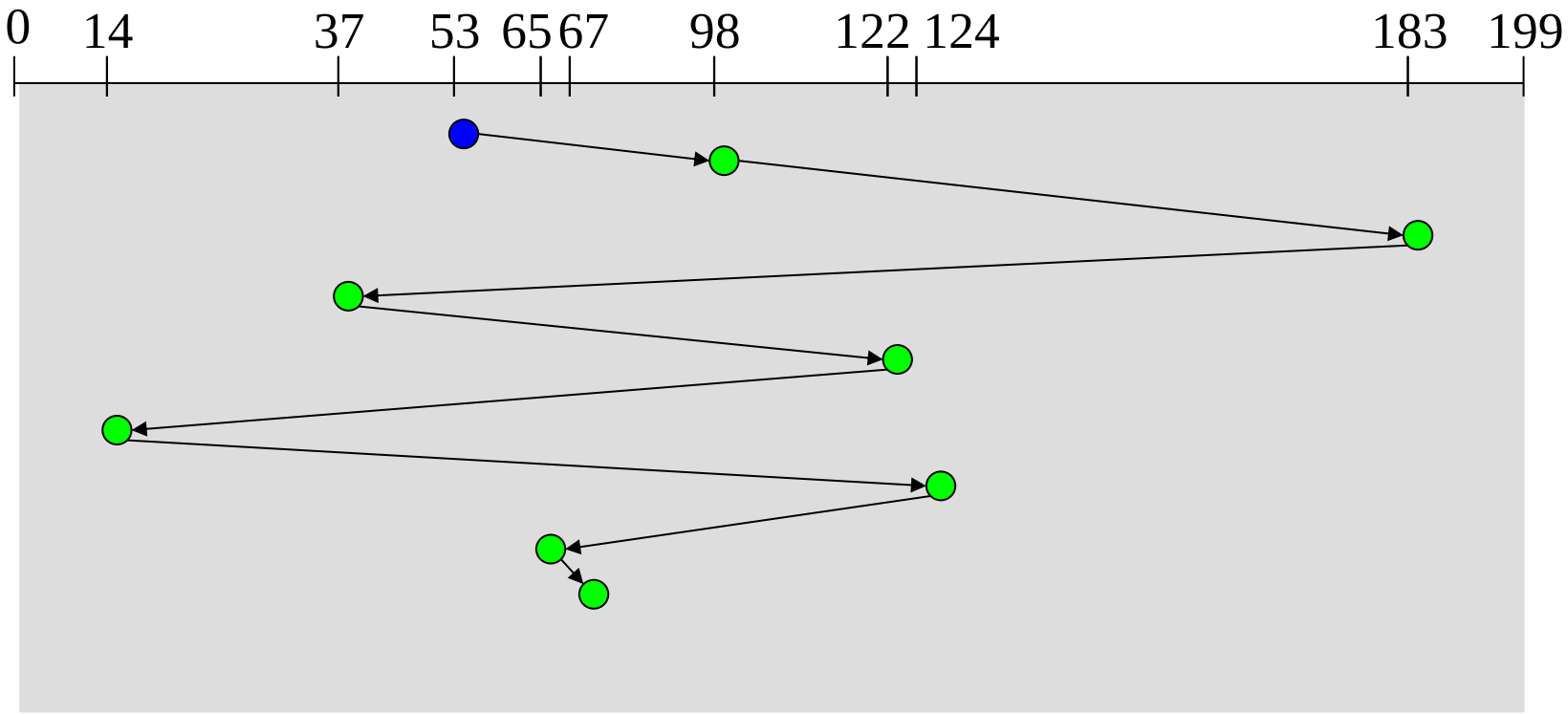
  Head pointer 53

- FCFS Scheduling
- SSTF Scheduling
- SCAN Scheduling
- C-SCAN Scheduling
- LOOK Scheduling
- C-LOOK Scheduling

FCFS--First Come First Served

排队

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



- **total head movement of 640 cylinders.**

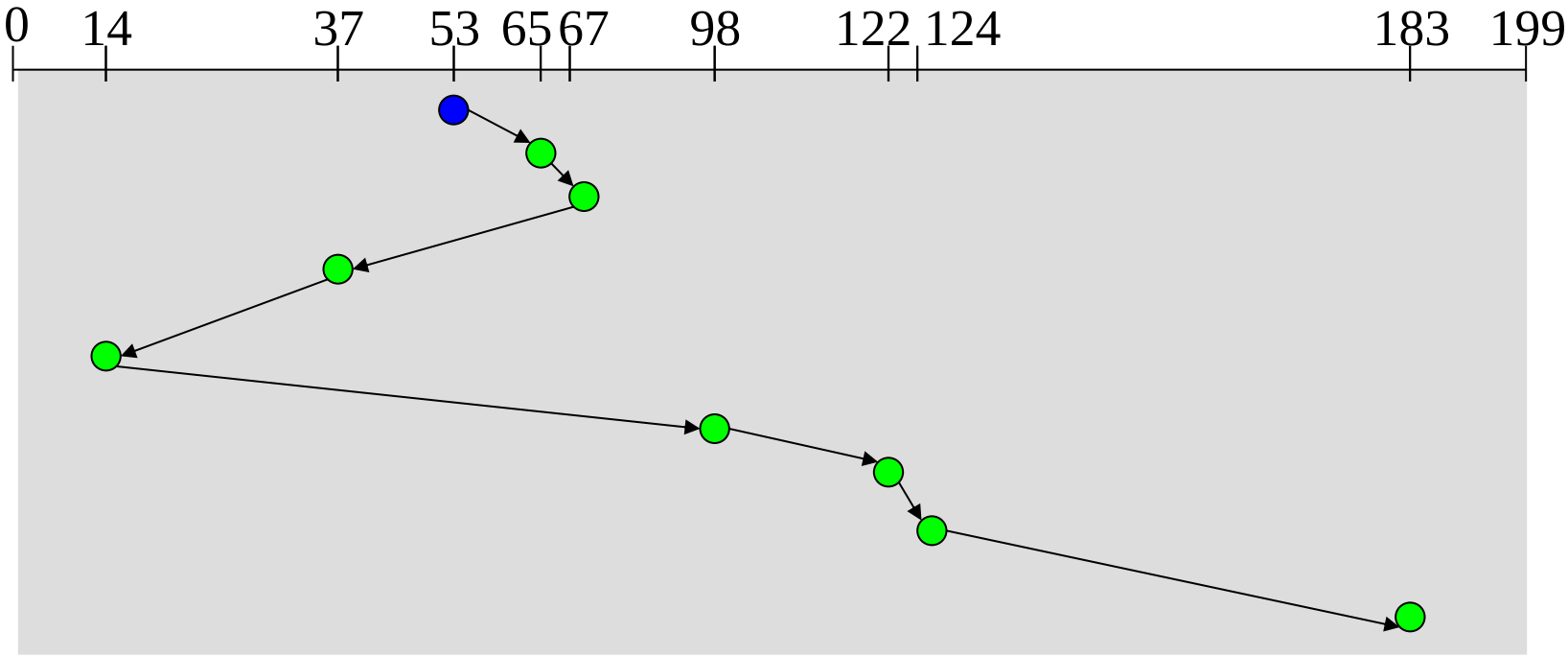
SSTF--Shortest Seek Time First

近者优先
效率

- **Selects the request with the least seek time from the current head position.**

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

  head starts at 53



- **Head movement of 236 cylinders.**
- **starvation**

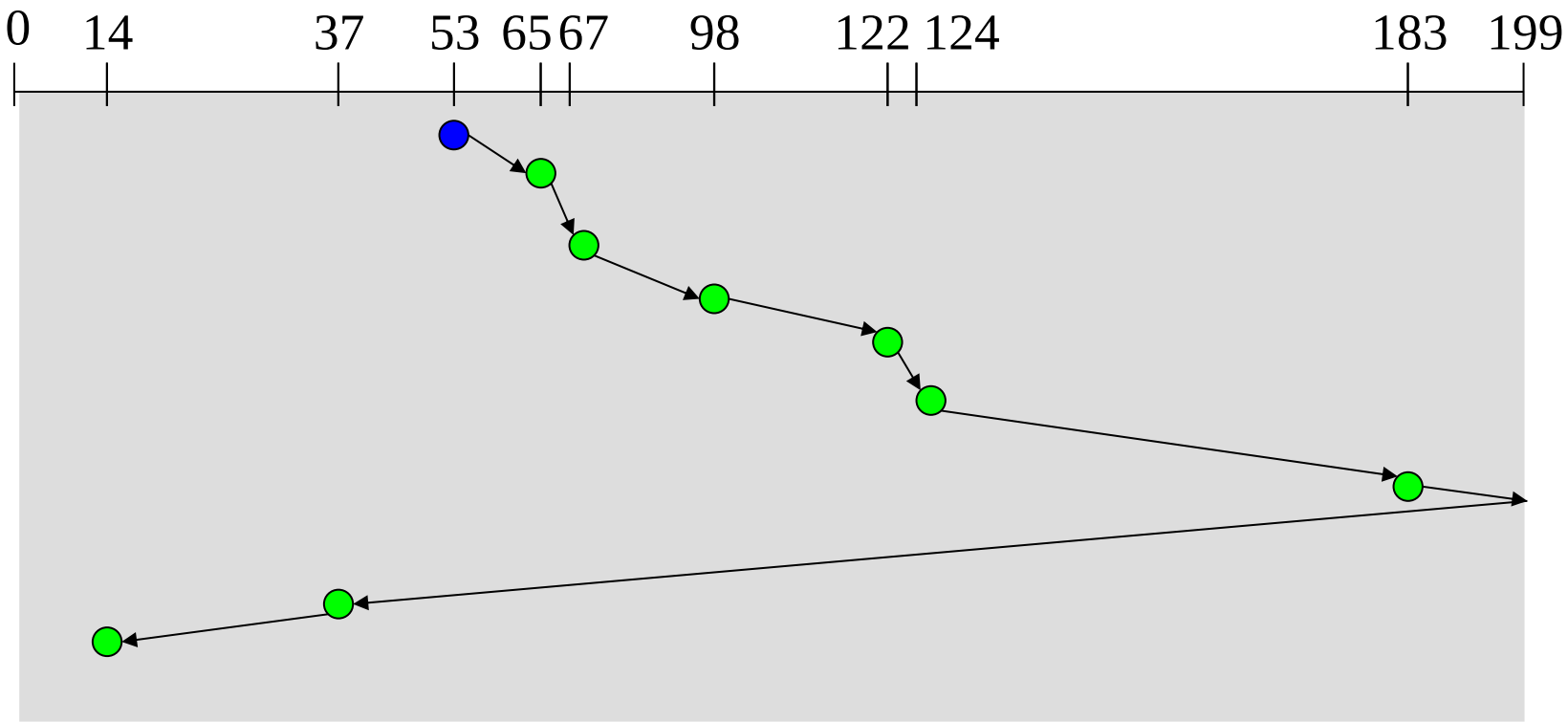
SCAN Scheduling

- **The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.**
- **Sometimes called the *elevator algorithm*.**

SCAN (Cont.)

公平/效率

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53, moving toward 199.



- **total head movement of 331 cylinders.**

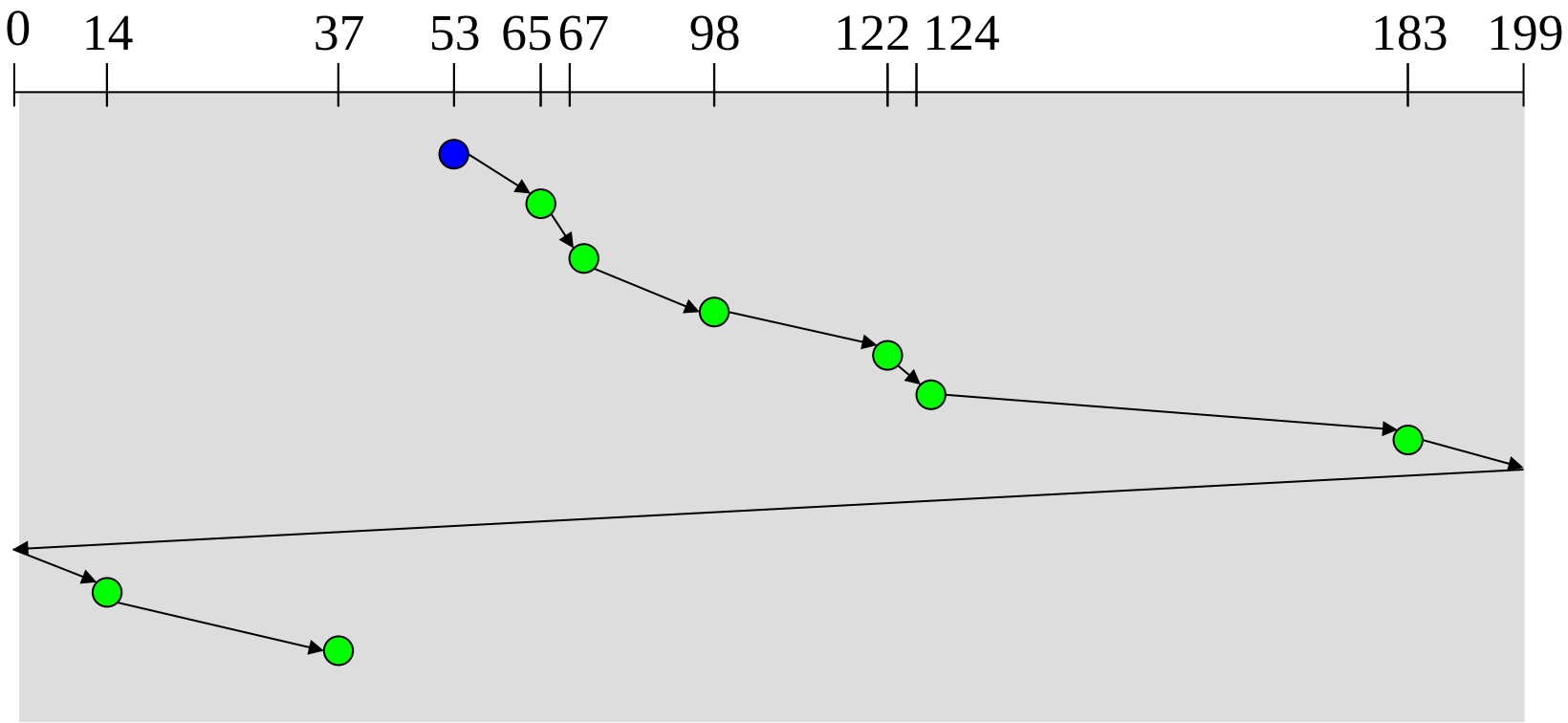
C-SCAN Scheduling— Circular SCAN

- **Provides a more uniform wait time than SCAN.**
- **The head moves from one end of the disk to the other. servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.**
- **Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.**

C-SCAN (Cont.)

公平

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53, moving toward 199



- **total head movement of 382 cylinders.**

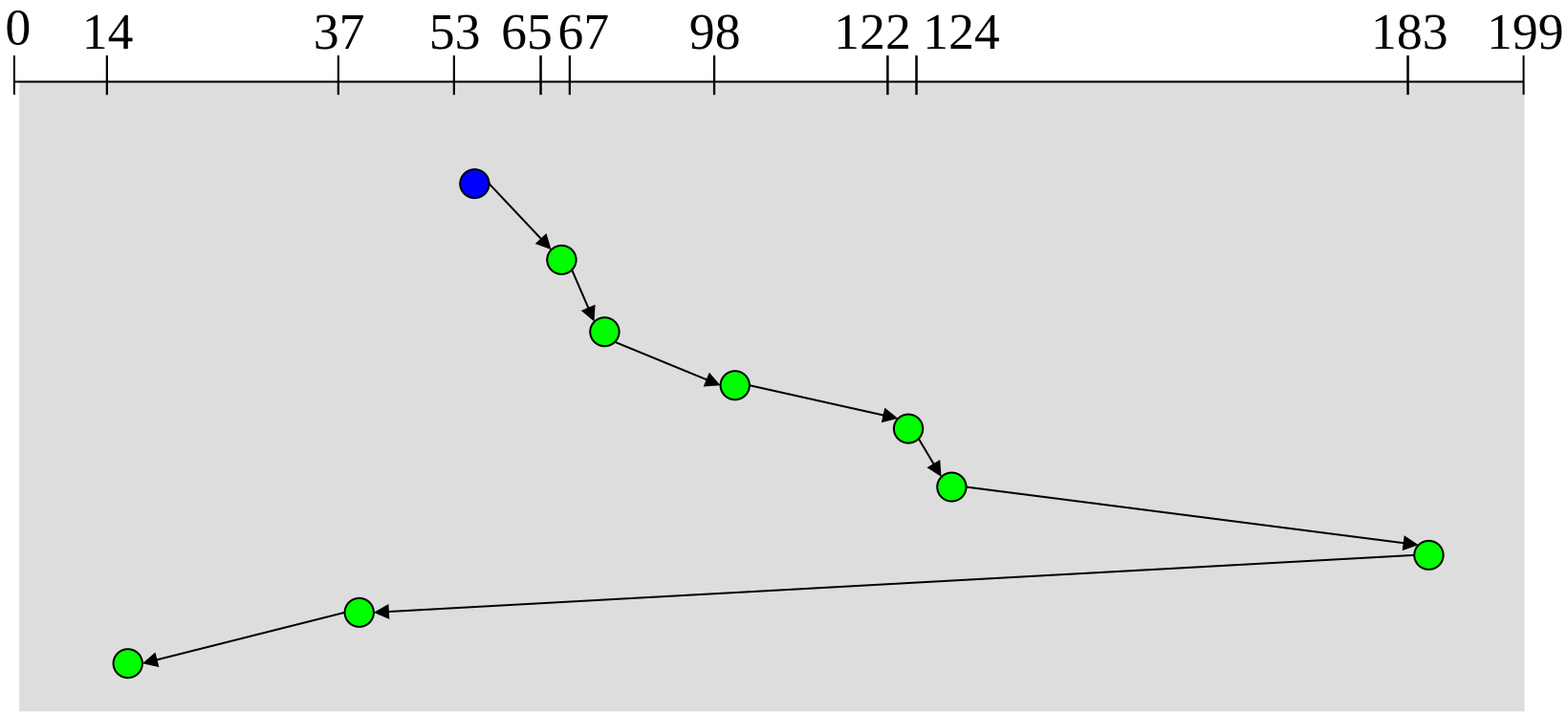
LOOK and C-LOOK Scheduling

- **Version of SCAN and C-SCAN**
- **Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.**

LOOK (Cont.)

公平/效率

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53, moving toward 199.

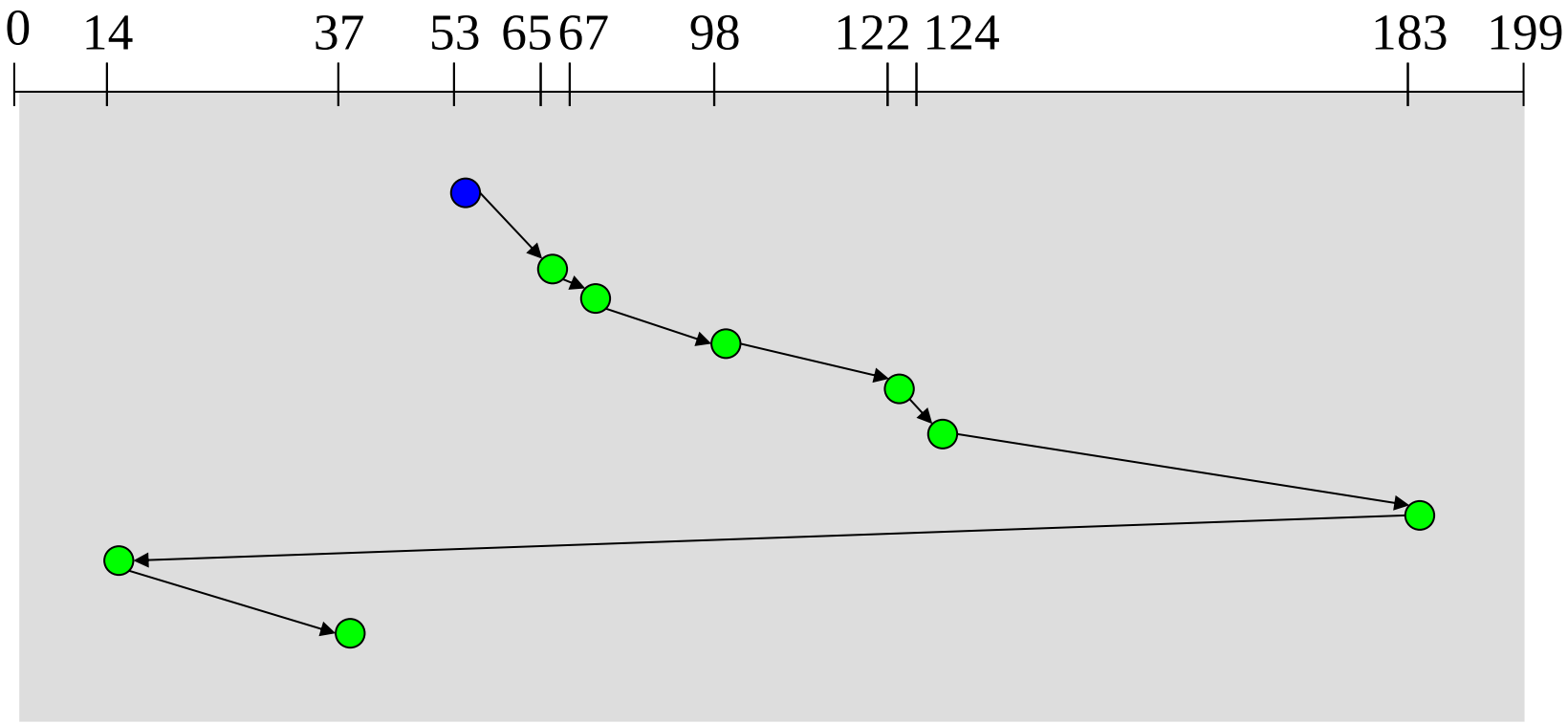


- **total head movement of 299 cylinders.**

C-LOOK (Cont.)

公平/效率

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53, moving toward 199



- **total head movement of 322 cylinders.**

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.
  - Less starvation
- 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.
- What about rotational latency?
  - Difficult for OS to calculate
- How does disk-based queueing effect OS queue ordering efforts?

12.5 Disk Management

- disk formatting
- booting from disk
- recovery from bad block

Disk formatting

- ***Low-level formatting*, or *physical formatting* — Dividing a disk into sectors that the disk controller can read and write.**
  - **Fills the disk with a special data structure for each sector.**
  - **The data structure typically consists of a header(sector #), a data area, and a trailer(ECC, error correction code).**
  - **Usually 512 bytes of data but can be selectable**
- **To use a disk to hold files, the OS still needs to record its own data structures on the disk.**
  - ***Partition* the disk into one or more groups of cylinders. each treated as a logical disk.**
  - ***Logical formatting* or "making a file system".**
  - **To increase efficiency most file systems group blocks into clusters**
    - **Disk I/O done in blocks**
    - **File I/O done in clusters**

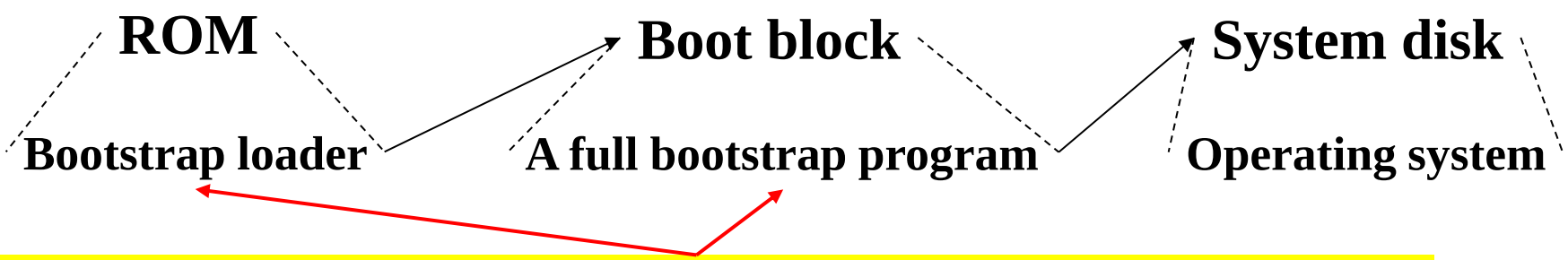
Disk Management (Cont.)

- **Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example).**

Boot block

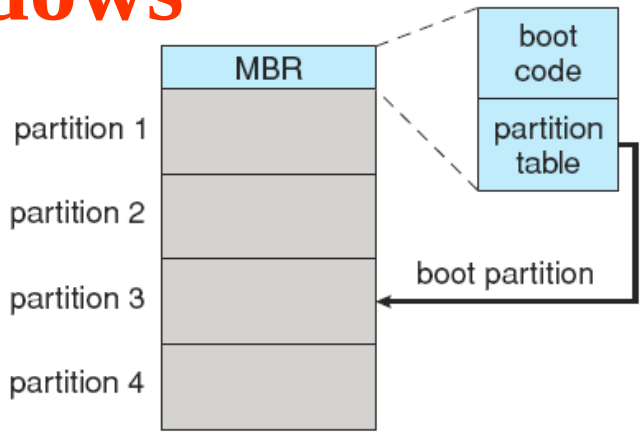
read only, needs no initialization, at a fixed location,
processor can start executing when powered up or reset.

- **Boot block initializes system**
  - **Initializes all aspects of the system, from CPU registers to device controllers and the contents of main memory, and then starts the operating system.**
  - **The bootstrap is stored in ROM.**
  - **The bootstrap program finds the OS kernel on disk, loads that kernel into memory, and jumps to an initial address to begin the operating-system execution.**
  - **Most systems store a tiny *bootstrap loader* program in the boot ROM.**
  - **The full bootstrap program is stored in a partition called the boot blocks, at a fixed location on the disk.**



Instructs disk controller to read the contents into memory.

Booting from a Disk in Windows



- Windows allows a hard disk to be divided into partitions
  - one partition, identified as the boot partition, contains the OS and device drivers.
- Boot code in the first sector on the hard disk, terms MBR (master boot record).
  - Besides boot code, the MBR contains a partition table listing the partitions for the hard disk and a flag indicating which partition the system is to be booted from.
- Booting begins by running code resident in the ROM.
  - Directs the system to read the boot code from the MBR.
  - Once the system identifies the boot partition, it reads the first sector (the boot sector) from that partition.
  - loading the various subsystems and system services.

Bad blocks

- Controller calculates the ECC and finds the sector is bad.
  - Reports this finding to the operating system.
- Methods used to handle bad blocks depends on the disk and controller in use.
- An unrecoverable hard error results in lost data.
- On simple disks, such as some disks with IDE controllers, bad blocks are handled manually.
  - *format* command, *chkdsk* command
- For more sophisticated disks
  - methods such as *sector sparing* (or *forwarding*) are used to handle bad blocks.
  - most disks are formatted to provide a few spare sectors in each cylinder and a spare cylinder as well.
  - Some controllers can be instructed to replace a bad block by *sector slipping*.

Exercise 1

- **Suppose that a disk drive has 5000 cylinders, number 0 to 4999. The drive is current serving a request at cylinder 143, and the previous request was at cylinder 94. The queue of pending requests, in FIFO order, is:**

  86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

- **Starting from current position. If the total distance (in cylinders) that the disk arm moves to satisfy all the pending request is 9769, so what disk-scheduling algorithm may the system use? And why?**

Answer to exercise 1

Exercise 2

- **There is a disk drive which has 200 cylinders 0 through 199. The SCAN algorithm and SSTF algorithms are used to schedule the request queue. Consider a disk queue with requests for I/O to blocks on cylinders,**
  98,182,37,122,14,65, 124,67.
  and the disk head is initially at cylinder 53.
  Suppose these two algorithms result in a same total head movement of cylinders.
- **Question:**
  Please describe the movement direction of the disk head when using the SCAN algorithms, which movement direction?

Answer to exercise 2

Exercise 3

- There is a disk drive which has 200 cylinders 0 through 199. The SCAN algorithm and SSTF algorithms are used to schedule the request queue. Consider a disk queue with requests for I/O to blocks on cylinders,
  98,182,37,122,14,65, 124,67.
  Staring from the current head position, the SCAN algorithm is used to schedule the request queue, no matter which direction the disk head moves, the schedule algorithm results in a same total distance (in cylinders).
- Which cylinder is the current head on? Why?

Answer to exercise 3

12.6 Swap-Space Management

- Swap-space, Virtual memory uses disk space as an extension of main memory.
  - Less common now due to memory capacity increases.
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw).

- Swap-space Use
- Swap-space Location
- Swap-space management

Swap-space Use

- Depending on the memory-management algorithms in use.
  - Use swap space to hold an entire process image.
  - Store pages that have been pushed out of memory.
- The amount of swap space needed depends on:
  - the amount of physical memory,
  - the amount of virtual memory it is backing,
  - the way in which the virtual memory is used.
- It may be safer to overestimate than to underestimate the amount of swap space required.
  - If a system runs out of swap space it may be forced to abort processes or may crash entirely.
  - Some systems recommend the amount to be set aside for swap space.
    - In the past, Linux, double the amount of physical memory.
- Some systems allow multiple swap spaces.

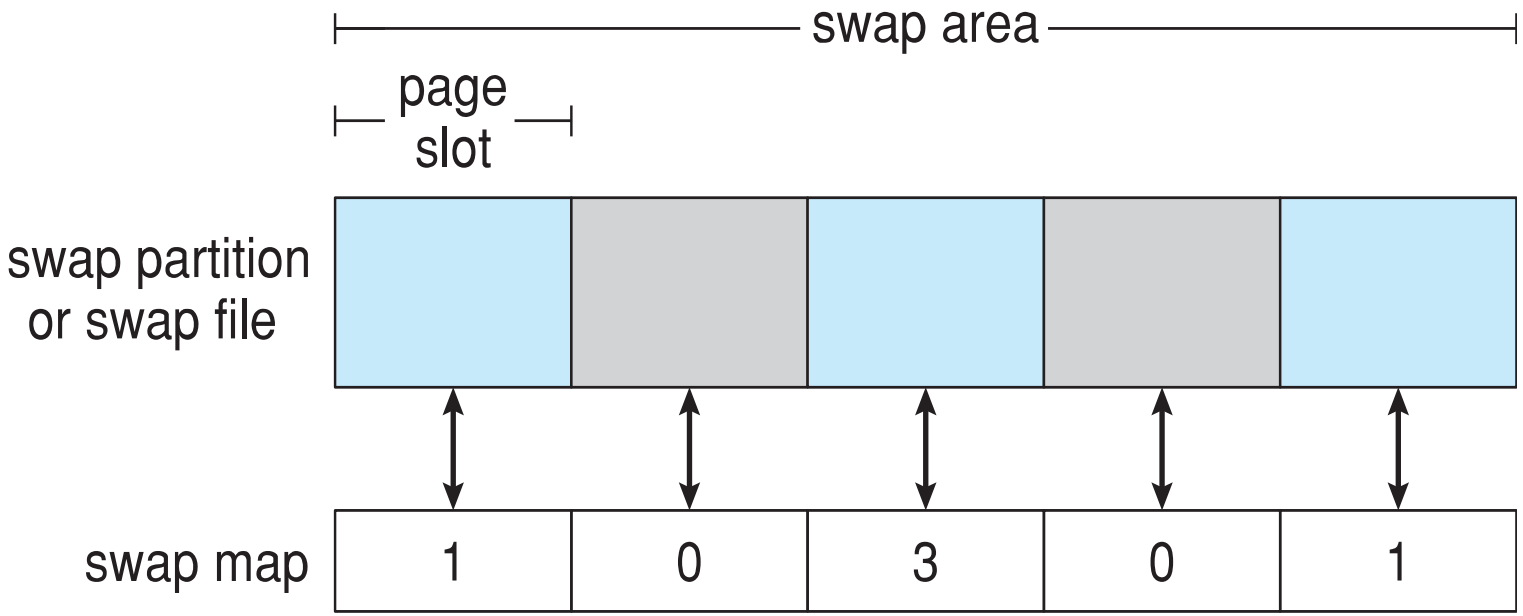
Swap-space Location

- A swap space can reside in one of two places:
  - Carved out of the normal file system, a large file
  - In a separate disk partition, raw partition
- A large file
  - Normal file-system routines can be used to create it, name it, and allocate its space.
  - Easy to implement, but inefficient.
- Raw partition
  - Swap-space storage manager, allocate and deallocate
  - A fixed amount of swap space is created during disk partitioning, adding more swap space requires repartitioning the disk.
  - Reinitialized at boot time.
- Some OSs are flexible and can swap both in raw partitions and in file-system space.
  - Linux: the policy and implementation are separate
    - allowing the machine's administrator to decide which type of swapping to use.

Swap-space management

- 4.3BSD allocates swap space when process starts; holds *text segment* (the program) and *data segment*.
- Kernel uses *swap maps* to track swap-space use.
- Solaris 2 allocates swap space only when a page is forced out of physical memory, not when the virtual memory page is first created.
  - File data written to swap space until write to file system requested.
  - Other dirty pages go to swap space due to no other home.
  - Text segment pages thrown out and reread from the file system as needed.

Data Structures for Swapping on Linux Systems



Homework

- **12.2**

Thinking:

12.1