Secondary Storage Structure

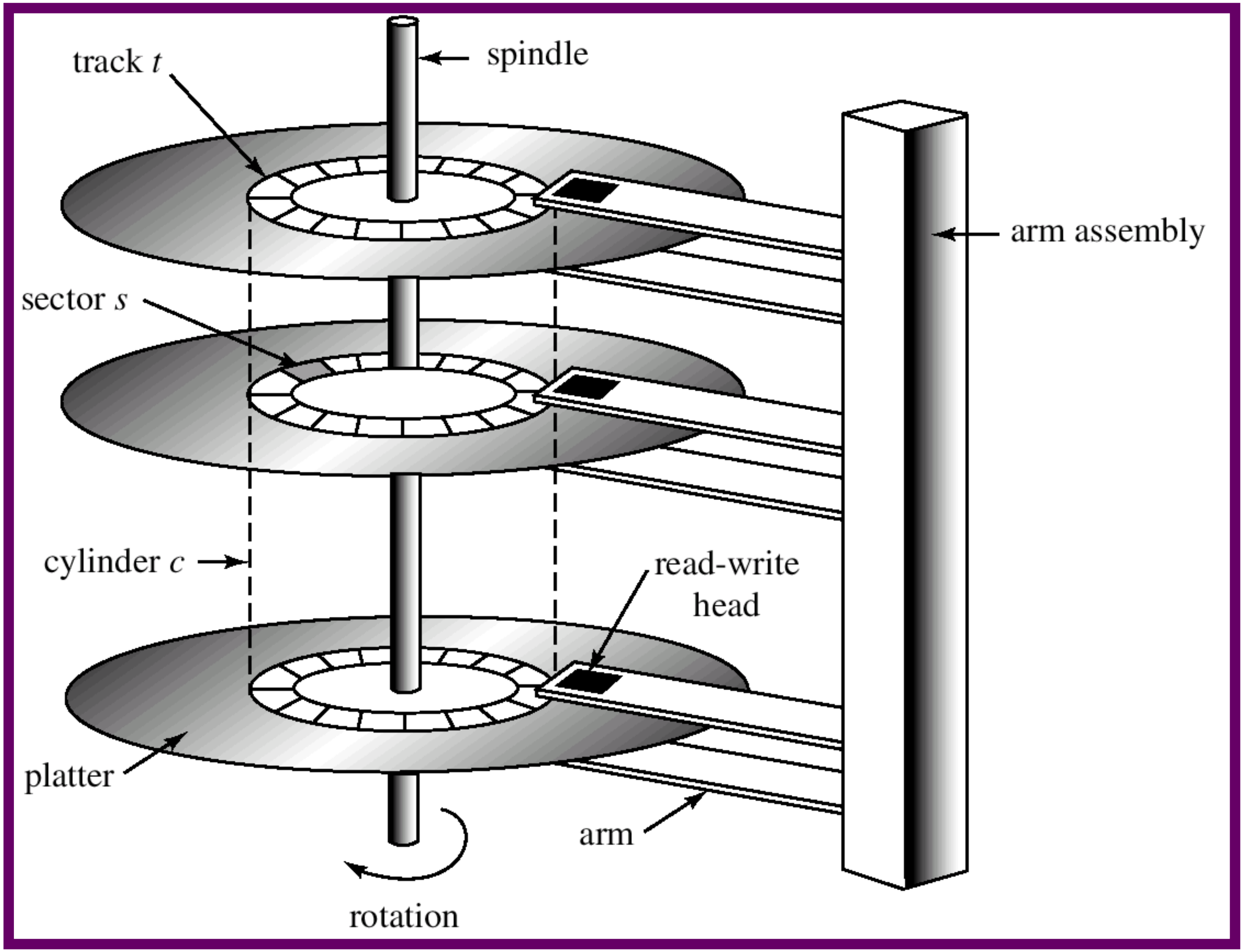
- Disk Structure
- Disk Scheduling
- Disk Management
- RAID Structure

Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer.

- 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.

Moving-Head Disk Mechanism



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 two major components
  - *Seek time* is the time for the disk are 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.)

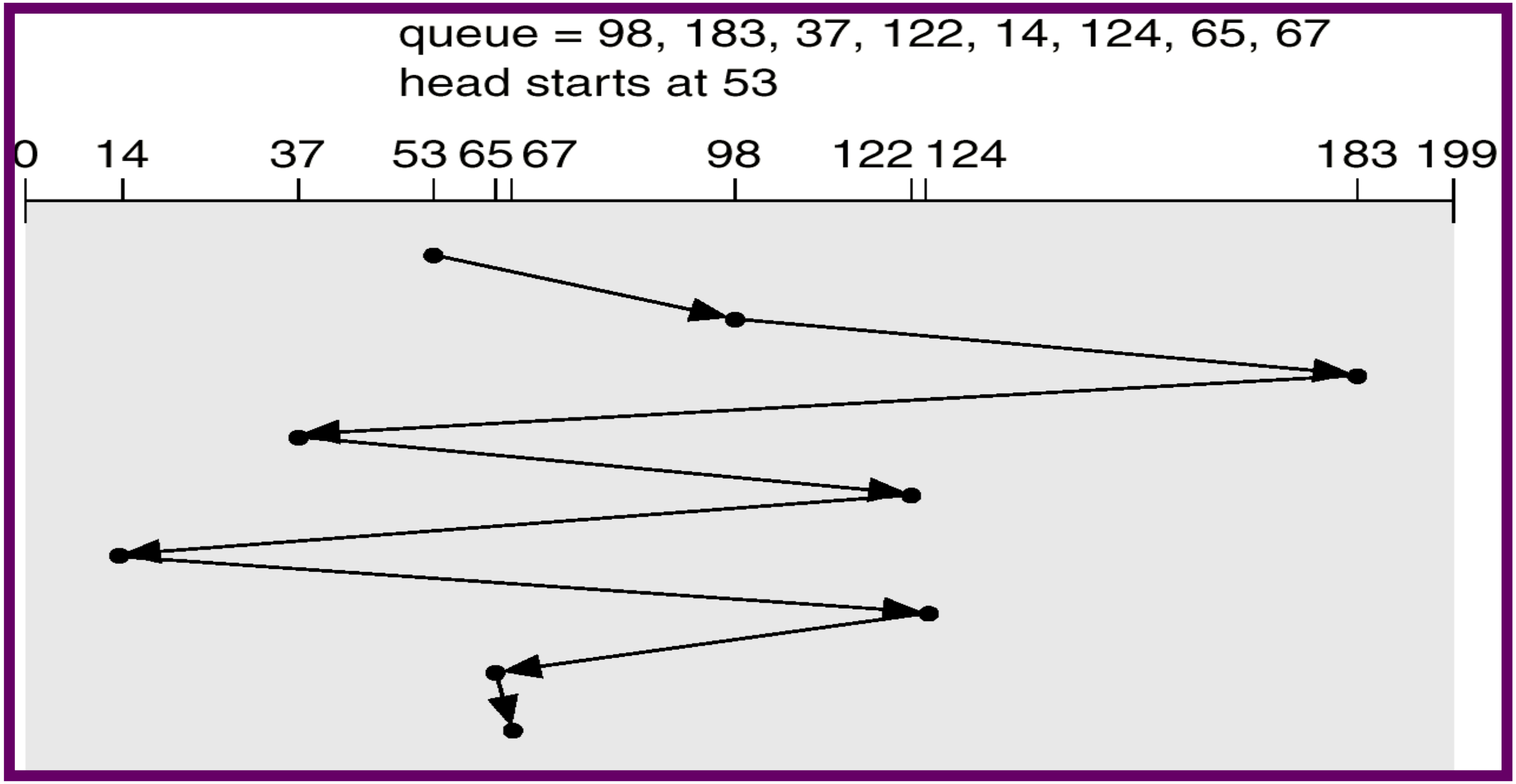
- Whenever a process needs I/O or from the disk, it issues a system call to the OS with the following information.
  - Whether the operation is input or output
  - What the disk address for the transfer is
  - What the memory address for the transfer is
  - What the number of bytes to be transferred is.
- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

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

Head pointer 53

FCFS

- FCFS
- +ve points: fair
- -ve points: slow
- Illustration shows total head movement of 640 cylinders.

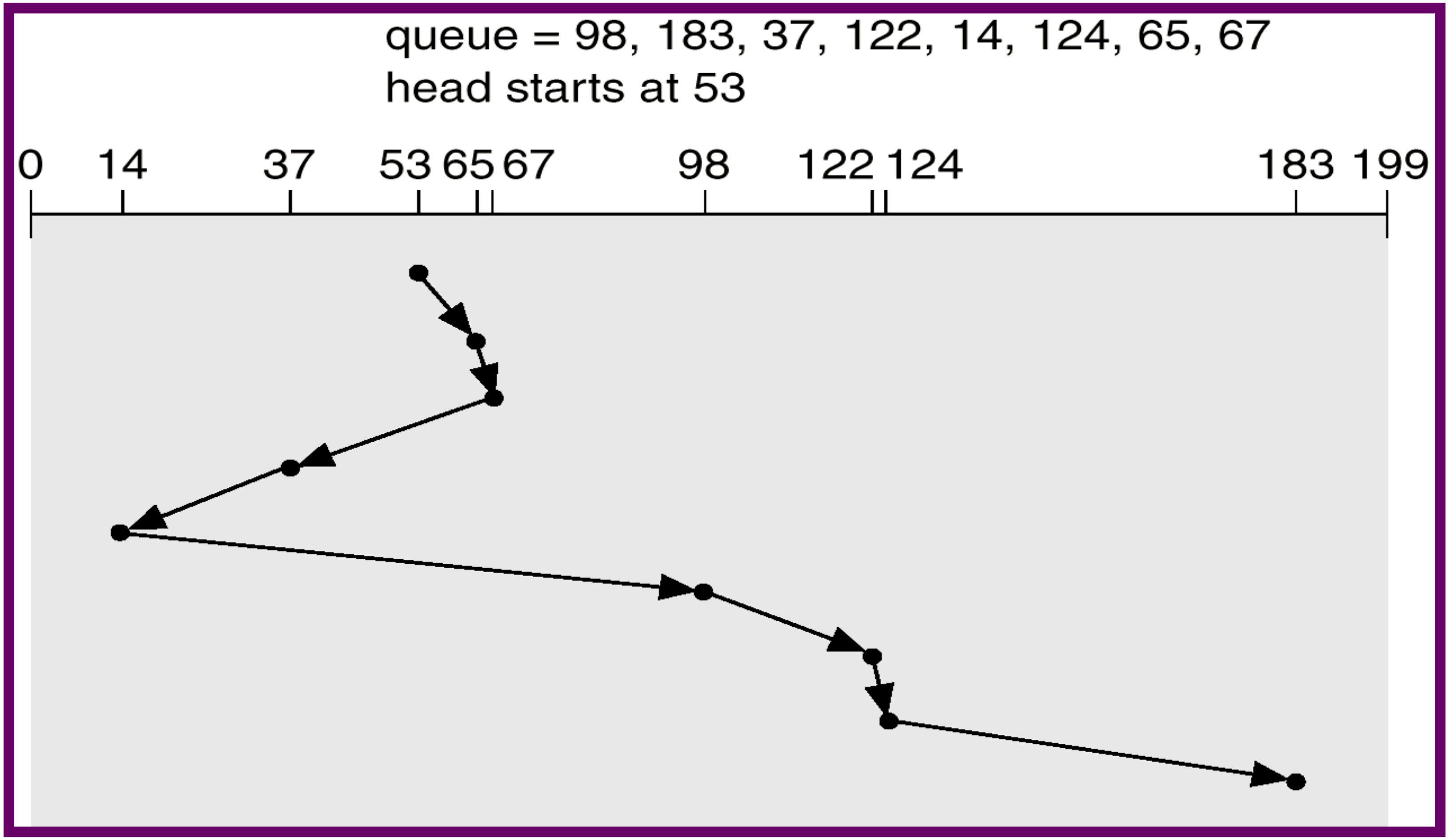


Shortest-seek-time-first (SSTF) algorithm

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.
- Illustration shows total head movement of 236 cylinders.

SSTF (Cont.)

Total head movement=236 cylinders

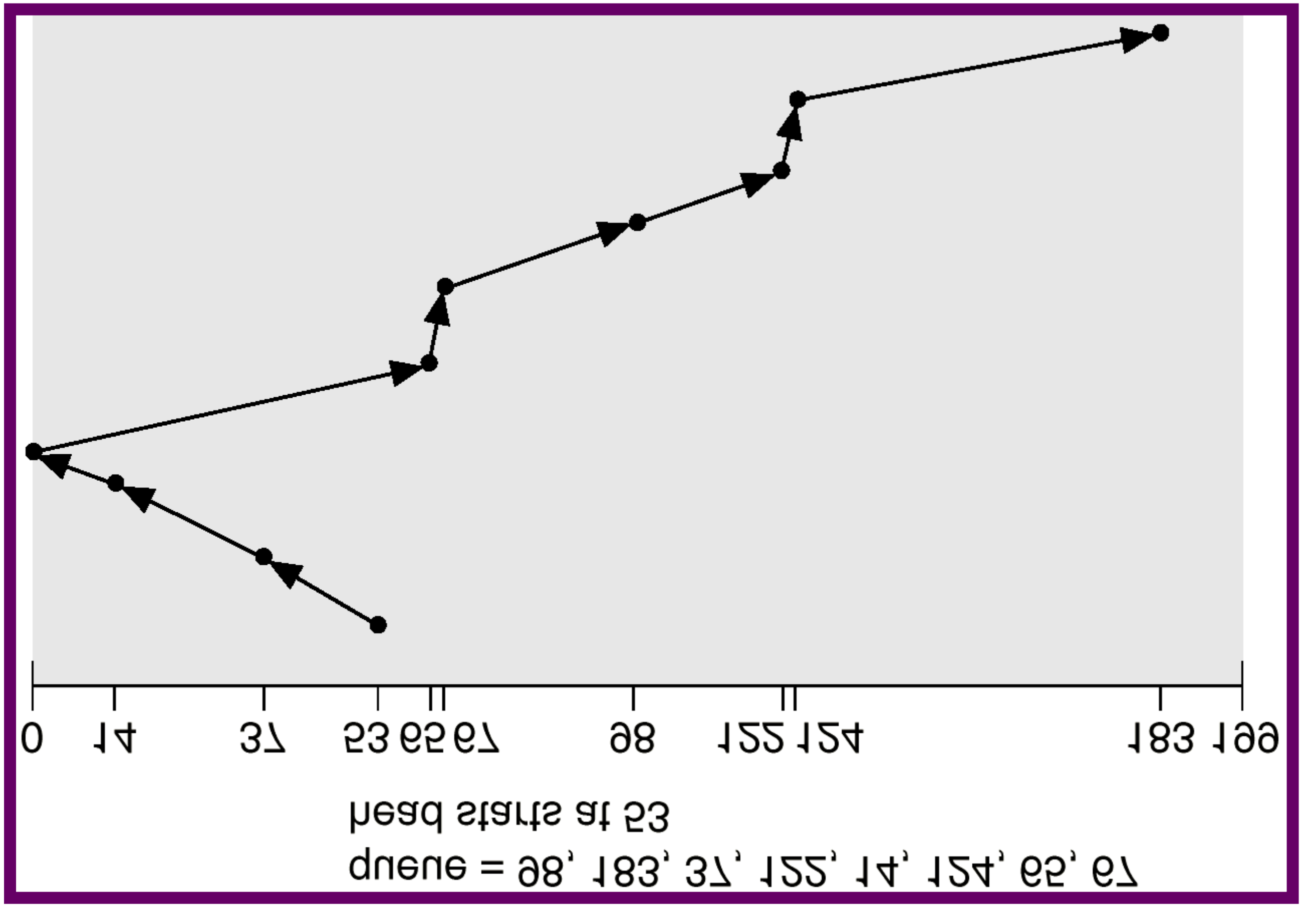


SCAN or Elevator algorithm

- 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*.
  - This algorithm does not give uniform wait time.
- Illustration shows total head movement of 208 cylinders.

SCAN (Cont.)

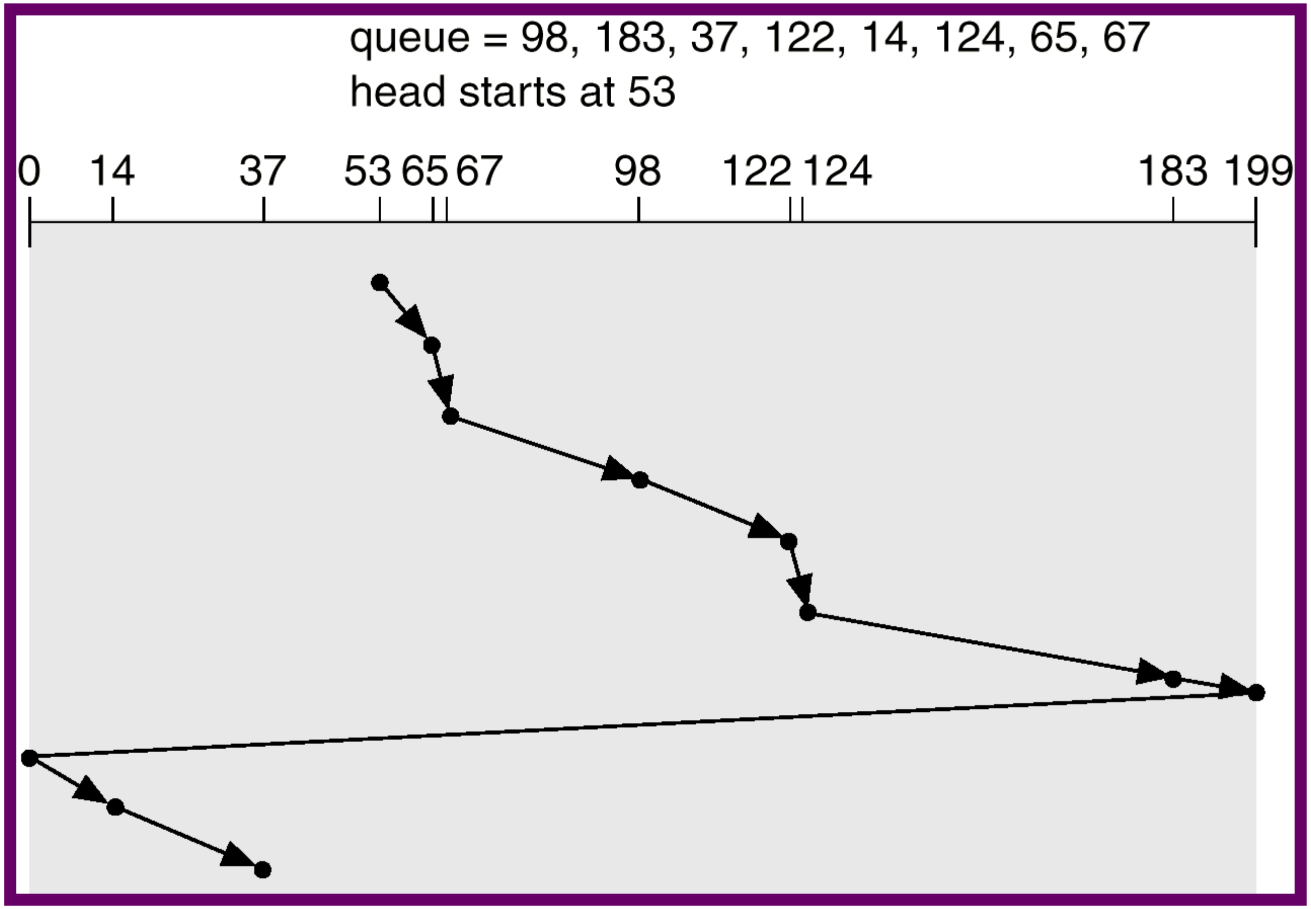
Total head movement=208 cylinders



C-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.)



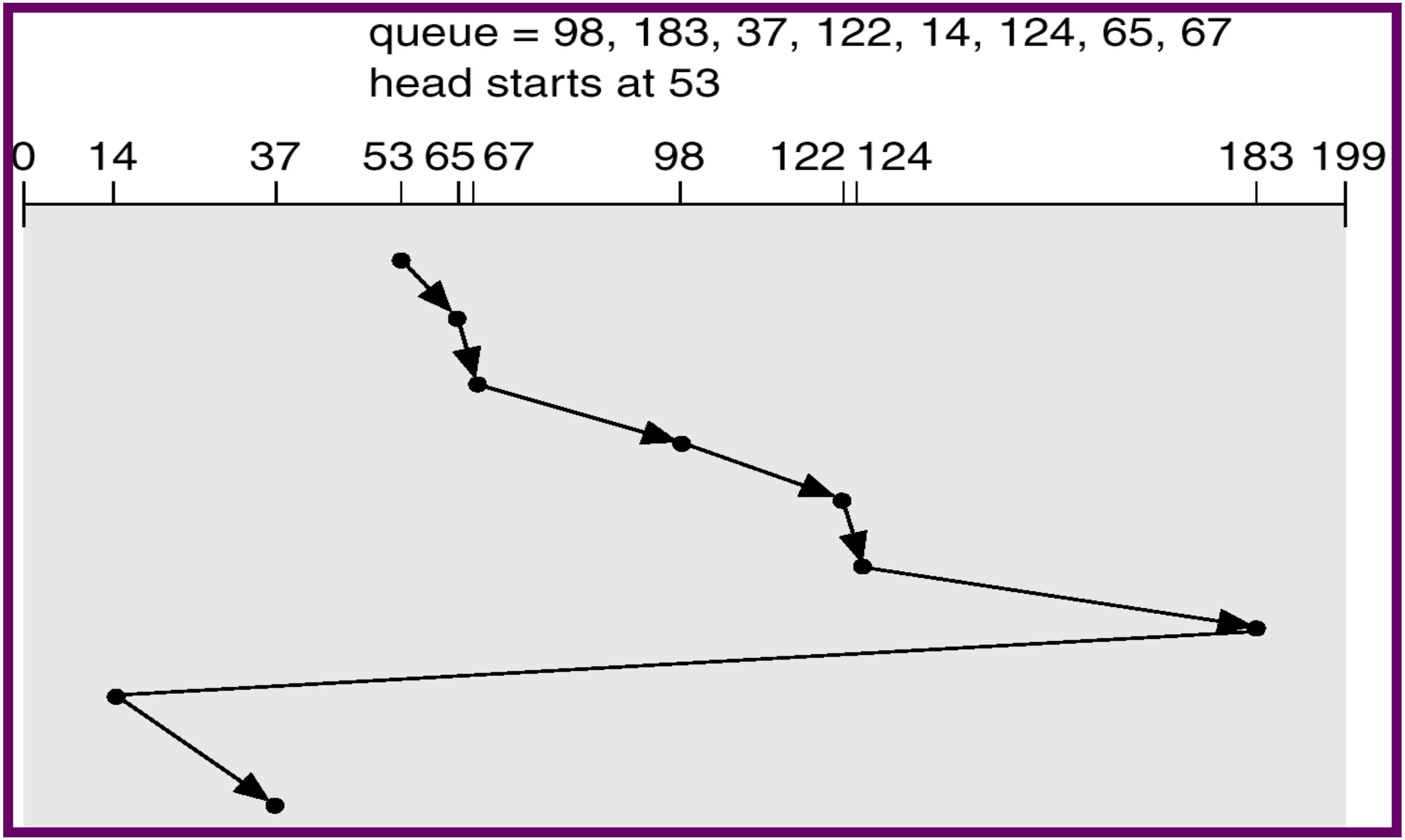
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

C-LOOK

- Version of 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.

C-LOOK (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

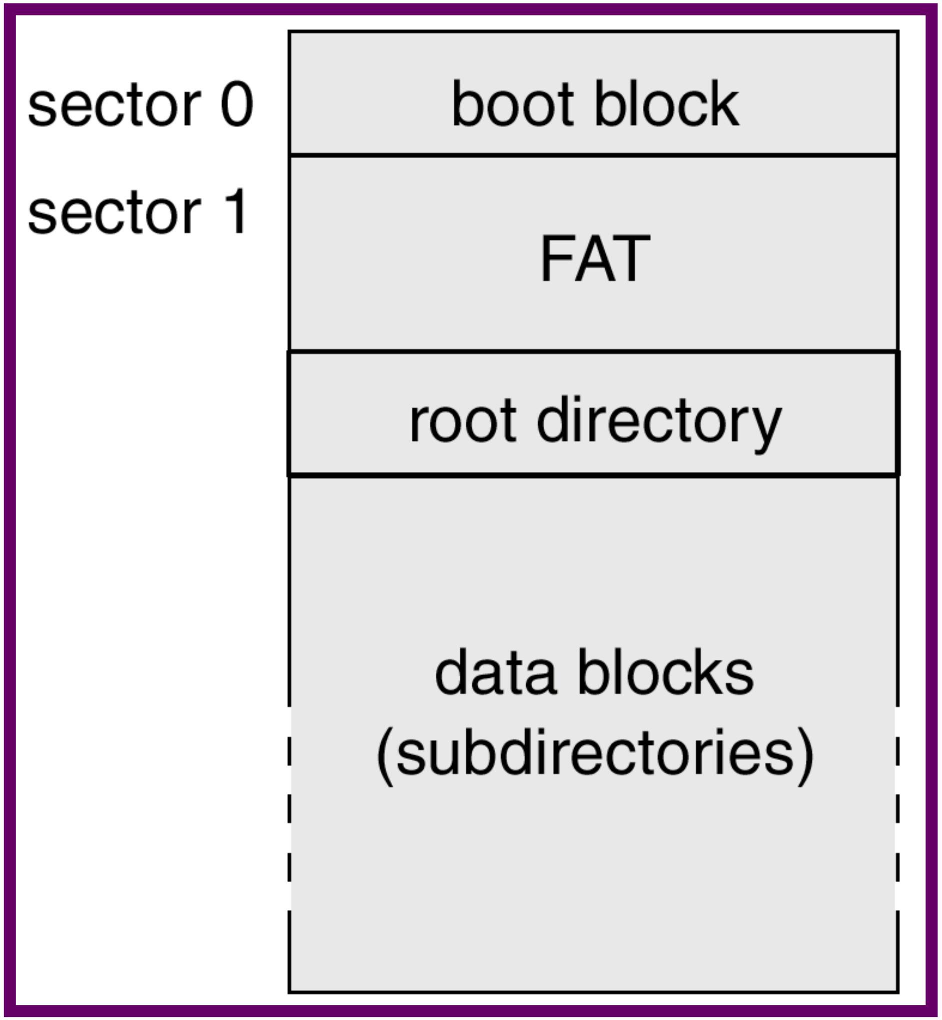
Selecting a Disk-Scheduling Algorithm

- 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.

Disk Management

- *Low-level formatting*, or *physical formatting* — Dividing a disk into sectors that the disk controller can read and write.
- To use a disk to hold files, the operating system still needs to record its own data structures on the disk.
  - *Partition* the disk into one or more groups of cylinders.
  - *Logical formatting* or “making a file system”.
- Boot block initializes system.
  - The bootstrap is stored in ROM.
  - *Bootstrap loader* program.
- Methods such as *sector sparing* used to handle bad blocks.

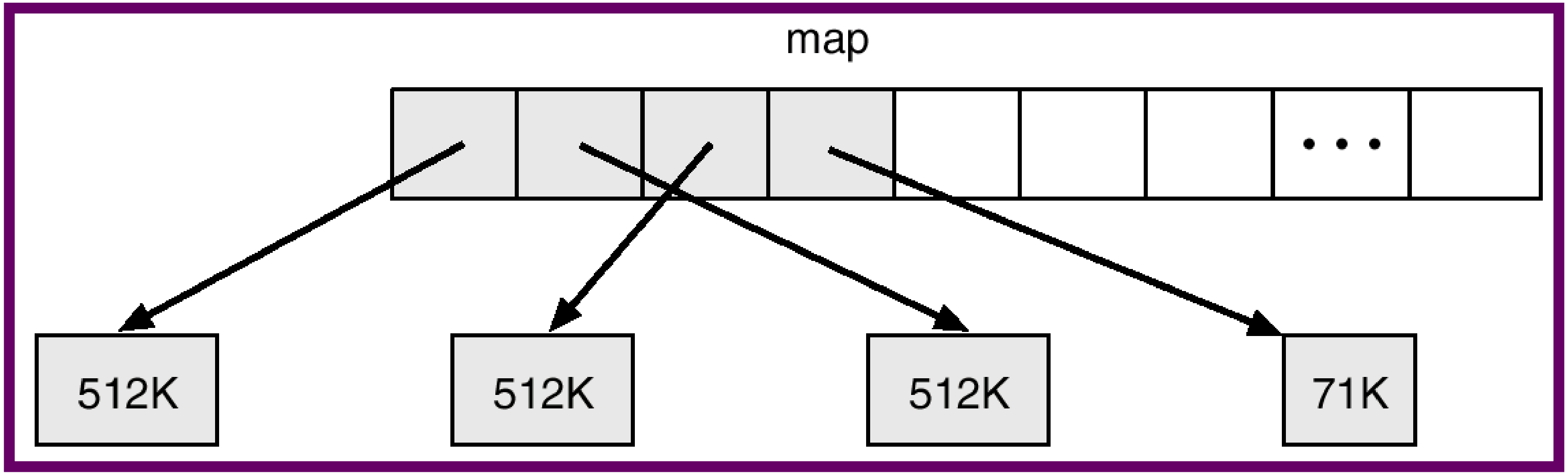
MS-DOS Disk Layout



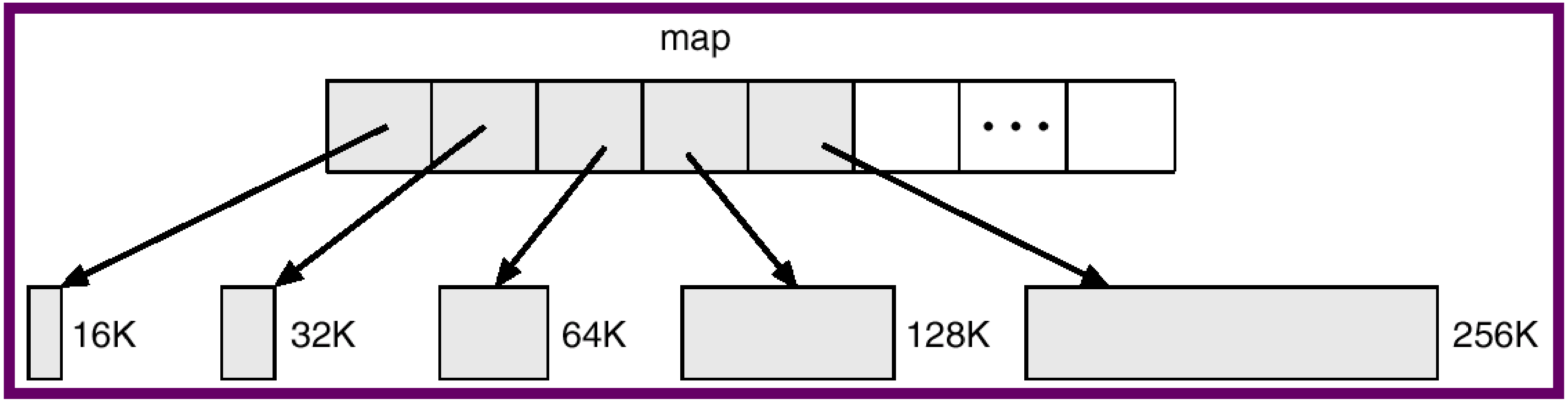
Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory.
- Swap-space can be carved out of the normal file system,or, more commonly, it can be in a separate disk partition.
- 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.

4.3 BSD Text-Segment Swap Map



4.3 BSD Data-Segment Swap Map



RAID Structure

- **Problem with disk**
  - **Data transfer rate is limited by serial access.**
  - **Reliability**
- **Solution to both problems: Redundant arrays of inexpensive disks (RAID)**
- In the past RAID (combination of cheap disks) is alternative for large and expensive disks.
- Today: RAID is used for their higher reliability and higher data-transfer rate.
- So the I in RAID stands for “independent” instead of ‘inexpensive”.
  - So RAID stands for **Redundant Arrays of Independent Disks.**
- RAID is arranged into six different levels.

RAID: Improvement of Reliability via Redundancy

- The chance that some disk out of N disk fail is much higher than single disk.
- Each failure of disk leads to loss of data
- Solution: Redundancy.
  - Store extra information which can be used in the event of failure.
- Simplest: Mirrored disks
  - Each logical disk consists of two physical disks.
  - Read from any disk
  - Write to both disks.

RAID: improvement in Performance via Parallelism.

- # of reads per unit time can be increased.
- With multiple disks we can improve the transfer rate with data stripping.
- Bit level Data stripping:
  - Splitting the bits of each byte across multiple disks.
  - If we have array of 8 disks, we write bit i of every byte to disk i.
  - The array of eight disks can be treated as single disk that are eight time normal size and eight times the access rate.
  - Every disk participates in the read.
  - But each access can read eight times as many data.
- Block level stripping: Blocks of files are stripped across multiple disks.
- Two goals
  - Increase the throughput of multiple small accesses.
  - Reduce the response time of large accesses.

Block-level stripping

- Block-level stripping is the logical extension of bit-level stripping.
- For example first block contains 1000 bytes. So it is a sequence of 7000 bits (ignoring 8th bit which is a parity bit). The first block is stored n Disk1. Similarly, another blocks are stored in Disk2, Disk3,..., Disk7 respectively. In Disk8, the parity of D1 to D7 blocks are stored.
- Disk1: 7000 bits of block1
- Disk2: 7000 bits of block2
- Disk3: 7000 bits of block3
- .
- .
- Disk7= 7000 bits of block7
- D8= 7000 bit parity (parity of 7 bits (combination of first bit of each block)).

- So if any of the disk fails, we can easily recover the failed data by accessing other 7 disks.

RAID levels

- RAID level 0: Disk arrays with stripping at the level of blocks, but without any redundancy (no mirroring and no parity)
  - Block level stripping
- RAID level1: Disk mirroring.
  - Block level striping
- RAID level2:
  - Error detection with parity bits.
  - Error correcting stores two or more parity bits.
  - Data can be stripped among disks and parity bits are stored in other disks.
  - Bit level stripping

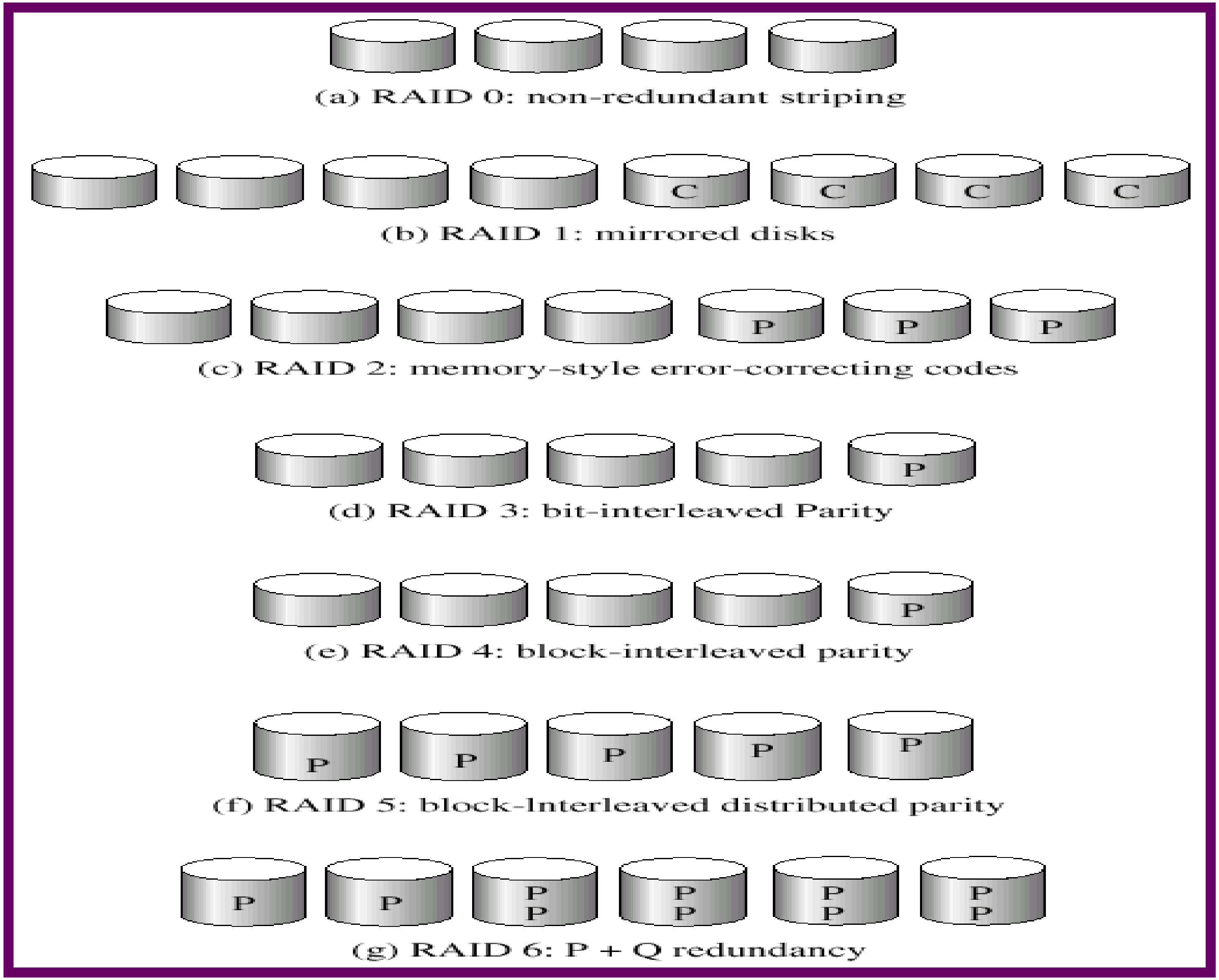
RAID levels

- RAID level 3: bit-interleaved parity organization
  - Disk controllers can detect whether a sector has been read correctly.
  - The bits of failed sector can be recovered by computing the parity of the remaining bits.
- RAID Level 3 is similar to RAID level 2 but less expensive
  - One disk overhead.
- RAID level 2 is not used in practice.
- Advantages of RAID level 3 over level 1 (mirroring)
  - One parity disk is needed; reducing the storage overhead
  - Transfer rate is same.
- RAID 3 performance problem
  - Computing and writing parity

RAID levels

- RAID level 4: block-interleaved parity organization
  - Uses block-level stripping
  - Keeps parity block on separate disk
  - If one disk fails the parity block and other blocks can be used to recover the failed disk.
- A block read accesses only one disk
  - Data transfer rate for each process is slower, However multiple read requests can be carried out in parallel.
  - Write results into two writes: block and corresponding parity.
- RAID level 5: Block interleaved distributed parity
  - Parity is distributed among all N+1 disks.
  - For each block one disk stores parity and others store data.
- RAID level 6:
  - Similar to RAID 5, but stores extra redundant information to guard against multiple disk failures.
- RAID 0+1 and RAID 1+0
  - Combination of RAID levels 0 and 1

RAID Levels



(a) RAID 0: non-redundant striping

(b) RAID 1: mirrored disks

(c) RAID 2: memory-style error-correcting codes

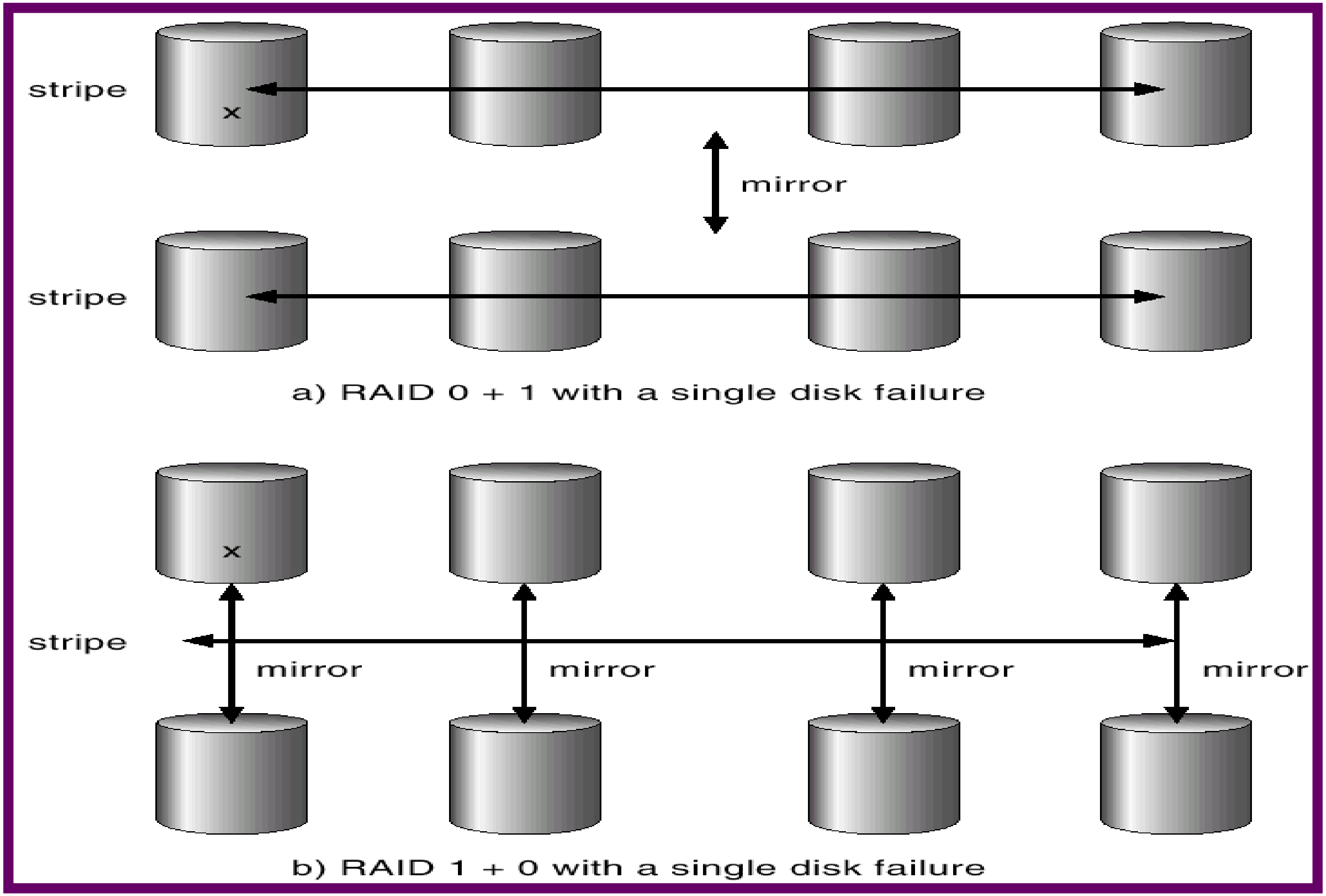
(d) RAID 3: bit-interleaved Parity

(e) RAID 4: block-interleaved parity

(f) RAID 5: block-Interleaved distributed parity

(g) RAID 6: P + Q redundancy

RAID (0 + 1) and (1 + 0)



a) RAID 0 + 1 with a single disk failure

b) RAID 1 + 0 with a single disk failure

Selecting a RAID level

- In case of a failure, the time to rebuild data vary with the RAID level.
- RAID level 0: used for high performance applications where data loss is not critical.
- RAID level 1: Popular for applications that require high reliability with fast recovery.
- RAID 0+1 and 1+0 are used where performance and reliability are important.
- RAID level 5 is used to store large volume of data.
- RAID level 6 is not supported.
- Hot spare disks can be used to reduce human intervention.

- The concepts of RAID can be generalized to other systems
  - Arrays of tapes
    - ✓ Recovery of damaged tape
  - Broadcast of data over wireless systems.
    - ✓ Receiver can reconstruct the information with parity information.