Device Management

Madiba Hudson-Quansah

Contents

Chapter 1

Types of Devices

- Devices are grouped into three categories:
  1. Dedicated Devices - Assigned to one job at a time, i.e. must be allocated for the entire duration of the job's execution.
  2. Shared Devices - Assigned to multiple jobs at a time, i.e. requires device manager supervision to allocate the device to different jobs at different times.
  3. Virtual Devices - Combination of dedicated and shared devices. Dedicated devices can be turned into shared devices through the use of spooling.

1.0.1 Dedicated Devices, Shared Devices and Virtual Devices

Dedicated devices are assigned to only one job at a time which they serve for the entire time it's active or until realised, e.g. monitors, mice, keyboards, printers. The disadvantage of this scheme is that devices must be allocated to a single user for the duration of a job's execution, which can be inefficient potentially leading to low device utilization.

Shared devices can be allocated to several processes at the same time, for example, a disk or any other direct access storage device (DASD), can be shared by interleaving their requests. The device manager supervises the allocation of the device to different jobs at different times. The advantage of this scheme is that it increases device utilization

Virtual devices are combination of both, i.e. dedicated devices that have been transformed into shared devices usually through the use of spooling.

1.0.2 USB Controller

The Universal Serial Bus (USB) Controller acts as an interface between the operation system, devices drivers and applications and the devices that are attached via the USB host. One USB host controller can accommodate as many as 127 devices through the use of hubs. Each device is uniquely identified by the USB host controller with an identification number, which allows many devices to exchange data with the commuter using the same USB connection. Th controller does this by assigning bandwidth do each device depending on its priority

Highest Priority - Assigned to real-time exchanges where no interruption in the data flow is allowed, i.e. video and audio streams.

Medium Priority - Assigned to devices that can allow occasional interrupts without affecting the use of the device, e.g. keyboards and mice.

Lowest Priority - Assigned to bulk transfers or exchanges that can accommodate slower data flow, for example file updates.

1.1 Management of I/O Requests

- The I/O traffic controller:

- Watches the status of devices, control units, and channels
- Determines if path available, Chooses a path out of several available paths, Waits for a path if all paths are busy.
- Maintains a database containing each unit's status and connections.

- The I/O scheduler:
  - Functions similarly to the Process scheduler.
  - Allocates devices, control units, and channels to I/O requests.
  - Decides which requests to allocate based on different criteria
  - I/O requests are usually not preemptible in most systems.
- The I/O device handler:
  - Performs data transfer handling, i.e. processing device interrupts, handling error conditions, and providing scheduling algorithms for servicing I/O requests.
  - Each device handler is device dependent, and each I/O device type has its own handler algorithm
- I/O requests use control blocks to store information:
  - Channel Control Block (CCB) - Contains information on channels
  - Control Unit Control Block (CUCB) - Contains information on control units
  - Device Control Block (DCB) - Contains information on devices

The Device Manager devices I/O requests into three parts with each one handled by a specific component of the device management subsystem.

I/O traffic controller - Watches the status of all devices, control units and channels.

I/O scheduler - Implements the policies that determine the allocation of, and access to the devices, control units and channels.

I/O device handler - Performs the actual transfer of data and processes the device interrupts.

1.1.1 I/O Traffic Controller

The I/O traffic controller monitors the status of every device, control unit and channel, with the complexity of the controller depending on the number of units in the I/O subsystem, and the number of possible paths between these units. The traffic controller has three main tasks to perform for each I/O request:

- Determine if there's at least one path to a device of the requests type available.
- If there are several paths available, choose which one to use.
- If all paths are busy it must determiner when one will become available.

To do this the traffic controller maintains a database containing the status and connections for each unit in the I/O subsystem grouped into Channel Control Blocks (CCB), Control Unit Control Blocks (CUCB) and Device Control Blocks (DCB).

Definition 1.1.1: Channel Control Block (CCB)

Contains information about each channel in the I/O subsystem, including:
- Channel ID
- Channel status (busy/free)
- List of control units connected to the channel
- List of processes waiting for the channel

Definition 1.1.2: Control Unit Control Block (CUCB)

Contains information about each control unit in the I/O subsystem, including:

- Control Unit Id
- Control Unit status (busy/free)
- List of channels connected to the control unit
- List of devices connected to the control unit
- List of processes waiting for the control unit

Definition 1.1.3: Device Control Block (DCB)

Contains information about each device in the I/O subsystem, including:

- Device ID
- Device Status (busy/free)
- List of control units connected to the device
- List of processes waiting for the device

1.1.2 I/O Scheduler

The I/O scheduler performs a similar function to the process scheduler, i.e allocating the devices, control units and channels to I/O requests. Under heavy loads, when the number of requests is greater than the number of available paths, the I/O scheduler must decide which request to satisfy first, with many of the criteria used by the process scheduler being applicable to I/O scheduling as well. However, unlike processes, I/O requests are usually not preemptible in most systems.

1.1.3 I/O Device Handler

The I/O device handler processes the I/O interrupts, handles error conditions, and provides detailed scheduling algorithms, which are device dependent. Each type of I/O device has its own device handler algorithm.

1.2 Sequential Access Storage Media

- Records are stored serially, one after another, with the earliest type being the magnetic tape, having:
  - Record length determined by program
  - Records are identified by position on the tape
  - Records are accessed by the read/write head moving across the tape to find the desired record.
  - Access time is long due to the need to wind through other records.
- Blocking groups records into blocks to reduce the amount of I/O operations needed, improving efficiency.

One of the earliest types of secondary storage media is magnetic tape, which writes and reads records in sequence from the beginning of a reel of tape to the end.

The length of each sequential record is usually determined by the job, and each record can be identified by its position on the tape. Data is recorded on eight of the nine parallel tracks that run the length of the tape with the ninth track holding a parity bit that is used for routine error checking.

The number of characters that can recorded per inch of tape is determined by the density of the tape. For example if you had records of 160 characters each, and where storing them on a tape with a density of 1,600 bytes per inch (bpi), then you would be able to store 10 records per inch of tape.

The tape needs time and space to stop, so a gap is inserted between each record, called the inter-record gap (IRG). The size of the IRG is about $\frac{1}{2}$ inch long regardless of the sizes of the records it separates. Therefore if 10 records are stored individually, there will be nine IRGs, between each record. This would result in about 5.5 inches of tape required to store 1 inch of data, which is inefficient.

Definition 1.2.1: Blocking

Grouping several records into blocks before recording them on tape to reduce the amount of I/O operations needed, improving efficiency.

Definition 1.2.2: Transfer Rate of a Tape Drive

The speed at which data can be read from or written to the tape, usually measured in inches per second (ips) or bytes per second (Bps), i.e.

$$\text{Transfer Rate} = \text{Density} \times \text{Transport Speed}$$

An alternative to storing records individually is to block them, when writing and unblock then when reading. The number of records in a block is usually determined by the job, with it being set to take advantage of the transfer rate of the tape drive.

Blocking requires that the entire block be read into a buffer before the records are processed so the size of the buffer must be at least as large as the block size.

Blocking has two main advantages:

- Reduces the number of I/O operations needed as a single READ command can move an entire block into main memory, i.e. a single physical record contains multiple logical records.
- Less storage space is wasted because the size of the physical record exceeds the size of the IRG.

And two main disadvantages:

- Overhead and software routines are needed for blocking, unblocking and recordkeeping.
- Buffer space may be wasted if you only need logical record but must read an entire block.

1.3 Direct Access Storage Devices (DASD)

- Can directly read/write specific disk areas without going through the whole medium, allowing random access.
- There are three categories:
  - Magnetic Disks
  - Optical Disks
  - Solid State (Flash) Memory
- Access time varies based on the location of the record, but usually faster than sequential access.

Direct Access Storage Devices (DASD), include all devices that can directly read/write to an arbitrary location on the storage medium without having to go through the entire medium. DASDs can be grouped into three categories:

1. Magnetic Disks
2. Optical Disc
3. Solid State (Flash) Memory

The variance in DASD access times is largely dependent on the location of the specific record being accessed, but is usually much faster than sequential access.

Chapter 2

Magnetic Disk Storage

- Single / Stack of magnetic platter
- Two recording surfaces (top and bottom)
- Each side when formatted is made up concentric tracks numbered from 0 on the outside to the highest track number in the centre.
- Read/write heads move in unison
- To access a record the system needs three things:
  - Cylinder number
  - Surface number
  - Sector number
- Access time is influenced by:
  - Seek time - Time to position read/write head on track, does not apply to fixed read/write head devices, usually the slowest part of the accessing process.
  - Search time - Time to rotate DASD, as the track rotates until the desired record is under the read/write head, influenced by rotational delay.
  - Transfer time - Time to transfer data from secondary storage to primary storage, usually the fastest part of the accessing process.

Magnetic disk drives feature the following characteristics:

Disk Arm - Moves the read/write heads, in unison, between each pair of surfaces, i.e. one for the top and one for the bottom of each platter.

Read/Write Head / Head - Positioned floating over each surface of each disk.

Spindle - Central shaft that holds and spins the platters.

Platter - Circular magnetic disk coated with magnetic material on which data is recorded. A stack of platters is called a disk pack, with several platters stacked on a common spindle. Each platter has two surfaces for recording data (top and bottom), with each surface formatted with a specific number of concentric tracks.

Track - Concentric circles on the surface of a platter, numbered from 0 on the outside to the highest track number in the centre. This is where data is recorded.

It is slower to fill a disk pack surface-by-surface than it is to fill it up track-by-track. If wee fill track 0 of all the surfaces, we have a virtual cylinder of data. There can be as many cylinders as there are tracks on a surface, and cylinders are as tall as there are surfaces in the disk pack. Therefore data is written and read on a cylinder basis.

Definition 2.0.1: Disk Sector

A subdivision of a track on a magnetic disk or optical disc, usually the smallest unit that can be read or written. On a magnetic disk the sectors are of different sizes, i.e bigger at the rim and smaller towards the centre. The disk spins at a constant angular velocity (CAV) to compensate for this.

To access any given record, the system needs three things:

Cylinder Number - So that the arm can move the read/write heads to the correct track.

Surface Number - So that the proper read/write head is activated.

Secot r number - So that the system can wait for the proper sector to rotate under the read/write head.

2.1 Access Times

There can be as many as three factors that contribute to the time required to access a file, depending on whether a disk has fixed or movable read/write heads. These are:

Seek Time - The time required to position the head on the proper track.

Search Time/ Rotational Delay - The time it takes to rotate the disk until the requested record is moved under the head.

Transfer Time - The time required to transfer the data from the disk to main memory.

Usually the seek time is the slowest part of the access process, followed by the search time, with the transfer time being the fastest.

2.2 Fixed-Head Magnetic Drives

- Record access only requires search time and transfer time, as the head is fixed to each track.
- Therefore the total access time is given as:

$$T_{\text{access}} = T_{\text{search}} + T_{\text{transfer}}$$

- DASDs rotate continuously with three basic positions needed to access a record in relation to the read/write head position.
- Little access variance as there is no seek time.
- Blocking minimizes access time.

Fixed-head disk drives have a separate head for each track on the disk, therefore seek time does not apply. The total access time is given as:

$$T_{\text{access}} = T_{\text{search}} + T_{\text{transfer}}$$

Because the disk rotates continuously, there are three basic positions for the requested record in relation to the position of the head:

About to Pass - The record is next to the head when the I/O command is executed, this gives a rotational delay of nearly zero. This is the best case scenario.

Opposite Side - The record is directly opposite the head when the I/O command is executed, resulting in a rotational delay of half a revolution, i.e $\frac{t}{2}$, where t is the time for a full revolution. This is the average case scenario.

Just Passed - The record has just passed the head when the I/O command is executed. This gives a rotational delay of nearly a full revolution, i.e t. This is the worst case scenario.

2.3 Movable-Head Magnetic Drives

Movable head disk drives add the computation of the seek time to the equation, i.e.

$$T_{\text{access}} = T_{\text{seek}} + T_{\text{search}} + T_{\text{transfer}}$$

The calculation for search time and transfer time is the same as that of fixed-head disk drives, with the seek time being dependent on the distance the head must move to reach the desired track. The maxium seek time which is the time taken to move the arm can be 10ms or less.

The variance in access time is much higher than that of fixed-head drives due to the seek time, with the average seek time being about one-third of the maximum seek time. Blocking can be used to minimize access time again as it reduces the number of I/O operations needed.

2.4 Device Handler Seek Strategies

- The device handler determines the device processing order, with the goal of minimizing seek time.
- The goals of a seek strategy are to:
  - Minimize arm movement
  - Minimize mean response time
  - Minimize variance in response time
- Common strategies include:
  - First-Come, First-Served (FCFS) - On average does not meet the goals.
  - Shortest Seek Time First (SSTF) -
    * Groups requests by proximity to current head position, i.e the current request being served.
    * Minimizes overall seek time, and reduces erratic arm movement.
  - SCAN variants - Uses a directional bit which indicates if the arm is moving towards/away from the disk centre. Then moves the arm form the outer to inner tracks servicing requests along the way, when the innermost/outermost track is reached the direction bit is flipped and the arm moves in the opposite direction servicing requests along the way.
    * LOOK (Elevator Algorithm) - Arm does not go to the innermost/outermost track if there are no requests in that direction. This reduces indefinite postponement of requests as the arm only moves in the direction of pending requests.
    * N-Step SCAN - Divides the request queue into sub-queues of size N. The arm services all requests in the current sub-queue using SCAN, while new requests are added to other sub-queues. This limits the maximum wait time for a request.
    * C-SCAN (Circular SCAN) - Arm only services requests on the inward sweep, when the innermost track is reached the arm jumps to the outermost track and continues servicing requests inwards. This provides a more uniform wait time compared to SCAN.
    * C-LOOK - Like C-SCAN but inward sweep stops at the last high-numbered track request, preventing the arm from going all the way to the centre if no requests exist there.
- The strat is:
  - FCFS for light loads
  - SSTF for moderate loads
  - Most scan variants for light to moderate loads
  - C-SCAN for moderate to heavy loads

A seek storage for the I/O device handler is the predetermined policy that the device handler uses to allocate access to the device among many processes that may be waiting for it. It determines the order in which the processes git the device with the goal of minimizing seek time. The goals of a seek strategy are to:

- Minimize arm movement
- Minimize mean response time
- Minimize the variance in response time

The broad generalization for selecting a seek strategy is:

FCFS - Works well with light loads

SSTF - Works well with moderate loads

SCAN and LOOK - Works well with light to moderate loads

C-SCAN - Works well with moderate to heavy loads

2.4.1 First-Come, First-Served (FCFS)

FCFS is the simplest device-scheduling algorithm, however on overt it doesn't meet any of the goals of a seek strategy. This is because it serves requests in the order they arrive without considering their location on the disk, which can lead to long seek times, high variance in response times, and extreme arm movement.

2.4.2 Shortest Seek Time First (SSTF)

SSTF works similarly to the Shortest Job Next (SJN) process scheduling algorithm, by selecting the request with the track closest to the one being served, i.e. the one with the shortest distance to travel (seek time).

This strategy minimizes overall seek time and reduces erratic arm movement. However, it can lead to starvation of requests that are far from the current head position if there are always closer requests to serve.

2.4.3 SCAN Variants

SCAN uses a direction bit to indicate whether the arm is moving towards the centre track or the outer track. The algorithm moves the arm methodically form the outer to the inner track and back again, servicing every request in its path. When the innermost or outermost track is reached, the direction bit is flipped and the arm moves in the opposite direction servicing requests along the way. It does not matter if there re no requests at the extreme tracks, the arm still moves from edge-to-edge with each sweep anyway.

2.4.3.1 LOOK (Elevator Algorithm)

The LOOK algorithm is a variant of SCAN where the arm does not go to all the way to either edge unless there are requests in that direction. Therefore it looks ahead for a request, using a wait queue before going to service it.

This eliminates the possibility of indefinite postponement of requests in out-of-the-way places, i.e. at the extreme tracks.

2.4.3.2 N-Step SCAN

N-Step SCAN is another variant of SCAN that holds all new requests until the arm starts on its way back, dividing the request queue into n sized sub-queues. The arm services all requests in the current sub-queue using SCAN, while new requests are added to other sub-queues. This limits the maximum wait time for a request.

2.4.3.3 C-SCAN (Circular SCAN)

C-Scan is another variant of SCAN where the arm picks up requests only on its path during the inward sweep. When the innermost track is reached, the arm jumps to the outermost track and starts servicing requests that came in during its inward sweep.

This way the system can provide quicker service to those requests that accumulated for the low number tracks while the arm was moving inward. The guiding concept behind this is that by the time the arm reaches the highest-number tracks, there are few requests immediately behind it. However there are many requests waiting at the low-number tracks, and these have been waiting the longest, i.e. C-SCAN provides a more uniform wait time compared to SCAN.

2.4.3.4 C-LOOK

An optimization of C-SCAN where the inward seep stops at the last high-numbered track request, so the arm doesn't move all the way to the last track unless its required to do so. Also the arm doesn't jump return to the outermost track, but rather to the first request in the queue.

2.4.4 Search Strategies: Rotational Ordering

- Rotational ordering optimizes search times by ordering requests based on their position on the disk surface.
- Read/write head movement time is hardware dependent and usually constant.
- Reduces time wasted due to rotational delay by grouping requests on the same track together.

Rotational ordering is a search time optimization technique that reorders pending requests once the heads have been positioned based on their position on the disk surface. Since the time taken for the

Chapter 3

Optical Disk Storage

- Single Spiralling track
- Constant sized sectors from centre to disc rim
- Spins at a constant linear velocity (CLV)
- More sectors and disc data than magnetic disk
- Performance measures:
  - Sustained data-transfer rate - Speed to read massive data mounts from disc, megabytes per second (Mbps), important for sequential access
  - Average access time - Average time to move head to specific disc location, milliseconds (ms)

3.1 CD and DVD Technology

- CD (Compact Disc)
  - Data recorded as zeros and ones, Pits (indentations/0), Lands (flat/1)
  - Reads with low-power laser where light that strikes land is reflected to the photodetector, and light that strikes a pit is scattered and absorbed. The photodetector then converts light intensity into a digital signal
- CD-R (Compact Disc-Recordable)
  - Contains several layers including a gold reflective layer and dye layer
  - Requires expensive disc controller
  - Records data using write-once technique:
    * Data is recorded using a high-power laser
    * High-power laser makes permanent marks on the dye layer
  - Data cannot be erased or modified
- CD-RW (Compact Disc Re-Writeable) / DVD-RW (Digital Versatile Disc Re-Writeable)
  - Data can be recorded, erased, and modified
  - Uses phase change technology, using amorphous and crystalline states to represent pits and lands respectively.
    * High-power laser heats up disc to change from crystalline (land) to amorphous state (pit), recording data
    * Low-energy beam heats up pits to allow recrystallization, erasing data
- DVD (Digital Versatile Disc)

- Higher storage capacity than CD (8.6 GB vs 700 MB)
  - Different laser wavelength (Red laser smaller pits, tighter spiral vs Infrared laser)
- Blu-Ray Disc
  - Higher storage capacity than DVD (25 GB vs 8.6 GB)
  - Smaller pits and tighter spiral than DVD
  - Uses blue-violet laser with shorter wavelength than DVD, allowing multiple layers (up to 4) to be stacked on top of each other.
    - BD-ROM (Blu-Ray Disc Read-Only Memory) - Pre-recorded discs
    - BD-R (Blu-Ray Disc Recordable) - Write-once discs
    - BD-RE (Blu-Ray Disc Re-Writeable) - Rewritable

Chapter 4

Solid State Storage

- Implements Fowler-Nordheim tunnelling to store data
  - Stores elections in a floating gate transistor
  - Electrons remain even when power is off
- Fast but expensive
- No moving parts, more durable than magnetic/optical storage
- Disadvantages:
  - Catastrophic failure - When one cell fails, it can cause neighbouring cells to fail, and no warning signs.
  - Data transfer rate degradation over time

4.1 Flash Memory Storage

- Electrically Erasable Programmable Read-Only Memory (EEPROM)
  - Non-volatile and removable
  - Emulates random access
- Writes data by sending electric charge to the floating gate
- Erases data by applying a strong electric field to remove electrons from the floating gate

Chapter 5

Components of the I/O Subsystem

- I/O Channels
-

5.1 I/O Channels

- Programmable units positioned between CPU and control unit
- Used to synchronize data transfer between CPU and I/O devices, as the CPU is faster than most I/O devices.
- Used to manage concurrent I/O operations
- Allows CPU and I/O devices to operate simultaneously
- I/O channel program
  - Specifies actions to be performed by the I/O channel
  - Controls data transmission between main memory and control units
- I/O control unit - Receives and interprets signals
- Disk controller (Disk Drive Interface) - Links disk drive and system bus
- I/O subsystem configuration - Multiple paths increase flexibility and reliability

Chapter 6

Communication among Devices

- Problems to resolve:
  - Know which components are busy/free - Structure interaction between units
  - Accommodate requests during heavy I/O traffic - Buffering records and queuing requests
  - Accommodate speed disparity between CPU and I/O devices - Buffering records and queuing requests
- I/O subsystem units finish independently of others
- CPU processes data while I/O operations are in progress
- Success requires device completion knowledge
  - Hardware flag tested by CPU - Channel Status Word (CSW) is a three bit flag to represent I/O subsystem components status (channel, control unit, device), one meaning busy, zero meaning free.
  - The flag is tested with polling and interrupts, with interrupts being more efficient.
- Direct Memory Access (DMA)
  - Allows control unit to access main memory directly
  - Transfers data without CPU intervention
  - Used for high-speed I/O devices (e.g., disk drives)
- Buffers
  - Temporary storage areas in main memory, channels, control units
  - Improves data movement synchronization between slow I/O devices and fast CPU/memory
  - Double buffering - Record processing by CPU while another is read or written by channel

Chapter 7

RAID

- Physical disk drive set viewed as a single logical unit, this is preferable to a few large-capacity disk drives.
- Improved I/O performance
- Improved data recovery in the event of a disk failure.
- Introduces redundancy which helps with hardware failure recovery.
- Significant factors in RAID level selection
  - Cost
  - Speed
  - The applications of the system
- Increases hardware cost due to the need for multiple disks.

7.1 Level Zero

- Uses data striping without redundancy (not considered true RAID)
- No parity and error corrections, i.e. no redundancy
- Devices appear as one logical unit improving performance
- Best for large quantifies of non-critical data
- Minimum of two disks required

7.2 Level One

- Uses data stripping with mirroring, this is considered true RAID as duplicate sets of data are stored.
- Provides redundancy and fault tolerance

7.3 Level Two

- Uses small strips and error-correcting codes (ECC) for redundancy, for example Hamming code
- Expensive and complex, as size of strips determines number of disks needed

7.4 Level Three

- An improvement on Level Two, using a single parity disk for redundancy instead of ECC.

7.5 Level Four

- Uses data striping with a dedicated parity disk for redundancy
- Computes parity for each strip
- Stores parities in corresponding strip on the parity disk

7.6 Level Five

- Improvement on Level Four that eliminates the dedicated parity disk distributing parity strips across all disks.
- Avoids Level Four bottleneck by spreading parity information across all disks.
- Complicated to regenerate data from a failed device as parity information is spread across all disks.

7.7 Level Six

- Provides an extra degree of error protection/correction
- Two different parity schemes are used (double parity)
- Parities are stored on separate disk across array

7.8 Nested RAID Levels

- Combines two or more standard RAID levels to leverage advantages of each level.
  - RAID 01 (0+1) - Level 1 system composed of Level 0 arrays
  - RAID 10 (1+0) - Level 0 system composed of Level 1 arrays
  - RAID 03(0+3) - Level 3 system composed of Level 0 arrays
  - RAID 30 (3+0) - Level 0 system composed of Level 3 arrays
  - RAID 50 (5+0) - Level 0 system composed of Level 5 arrays
  - RAID 60 (6+0) - Level 0 system composed of Level 6 arrays