I/O Management and Disk Scheduling

Chapter 11

Categories of I/O Devices

- Human readable– Printers, Display, Keyboard etc.
  - Used to communicate with the user
- Machine readable– Disk and tape drives, Sensors, Controllers, Actuators
  - Used to communicate with electronic equipment
- Communication– Digital line drivers
  - Used to communicate with remote devices

Differences in I/O Devices

- Data rate (several orders of magnitude)
- Application (file management sw, virtual mem. management sw and hw, different priorities--terminal)
- Complexity of control
- Unit of transfer (byte steam for terminal vs large blocks for disk)
- Data representation– different encoding schemes
- Error conditions – respond differently to errors

Performing I/O: Types

- Programmed I/O-- Process is busy-waiting for the operation to complete
- Interrupt-driven I/O
  - I/O command is issued
  - Processor continues executing instructions
  - I/O module sends an interrupt when done
- Direct Memory Access (DMA)
  - DMA controls exchange of data between main memory and the I/O device
  - Processor interrupted only after entire block has been transferred

Evolution of the I/O Function

- Processor directly controls a peripheral device
- Controller or I/O module is added
  - Processor uses programmed I/O without interrupts
  - Processor does not need to handle details of external devices

Evolution of the I/O Function

- Controller or I/O module with interrupts
  - Processor does not spend time waiting for an I/O operation to be performed
- Direct Memory Access
  - Blocks of data are moved into memory without involving the processor
  - Processor involved at beginning and end only

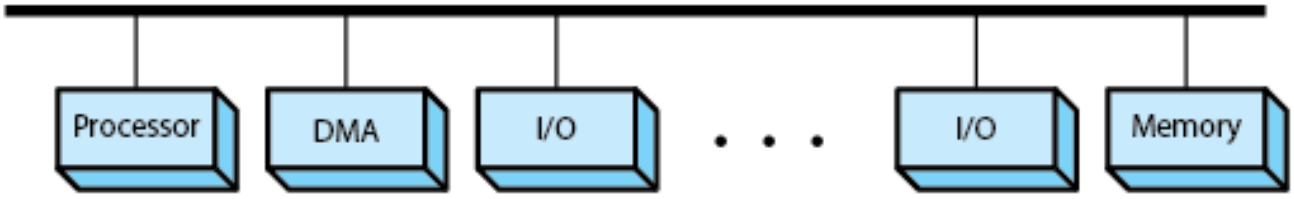
Evolution of the I/O Function

- I/O module is a separate processor
- I/O processor
  - I/O module has its own local memory
  - Its a computer in its own right

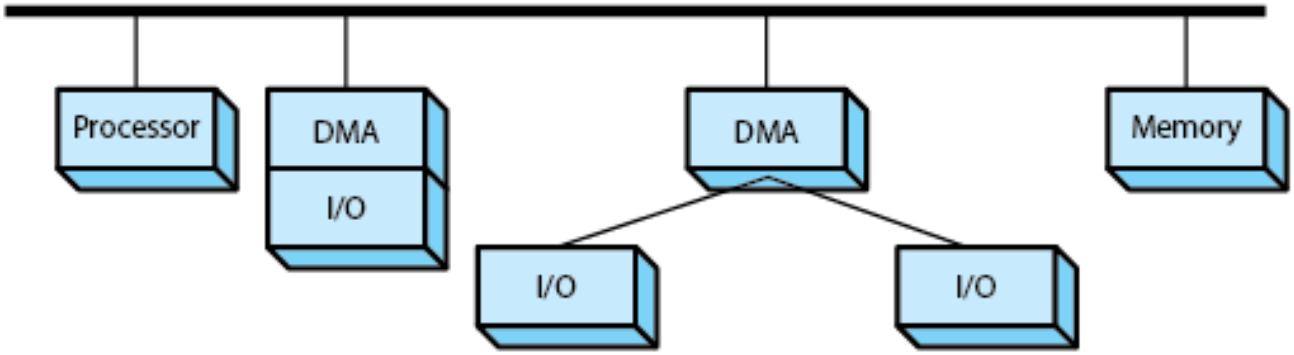
Direct Memory Access

- Processor delegates I/O operation to the DMA module
- DMA module transfers data directly to or form memory
- When complete DMA module sends an interrupt signal to the processor

DMA Configurations

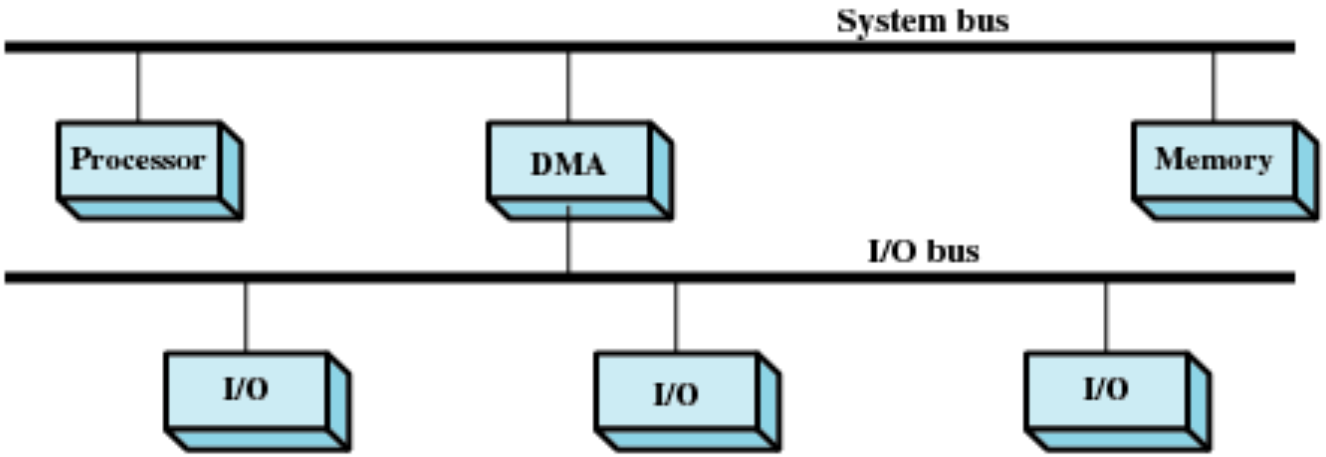


(a) Single-bus, detached DMA



(b) Single-bus, Integrated DMA-I/O

DMA Configurations



(c) I/O bus

Figure 11.3 Alternative DMA Configurations

Operating System Design Issues

- Efficiency
  - Most I/O devices extremely slow compared to main memory
  - Use of multiprogramming allows for some processes to be waiting on I/O while another process executes
  - I/O cannot keep up with processor speed
  - Swapping is used to bring in additional Ready processes which is an I/O operation

I/O Buffering

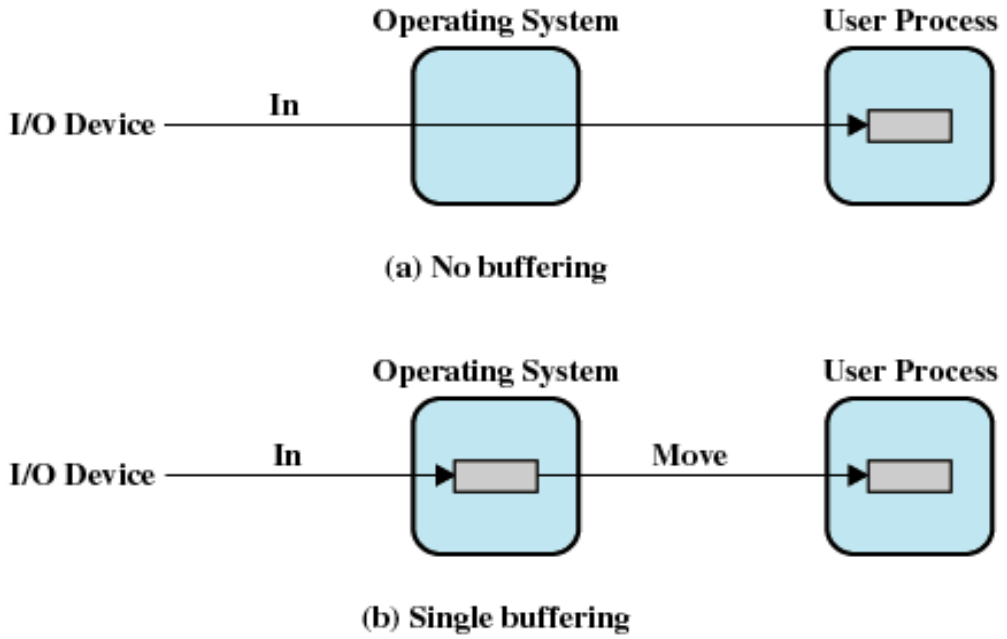
- Reasons for buffering
  - Processes must wait for I/O to complete before proceeding
  - Process cannot be entirely swapped out– corresponding pages must remain in main memory during I/O
  - Risk of single process deadlock– process swapped out before I/O operation starts

I/O Buffering

- Block-oriented
  - Information is stored in fixed sized blocks
  - Transfers are made a block at a time
  - Used for disks and tapes
- Stream-oriented
  - Transfer 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

Single Buffer

- OS assigns a buffer in MM for an I/O request
- Block-oriented
  - Input transfers made to buffer
  - Block moved to user space when needed
  - Another block is moved into the buffer (read ahead)



Single Buffer

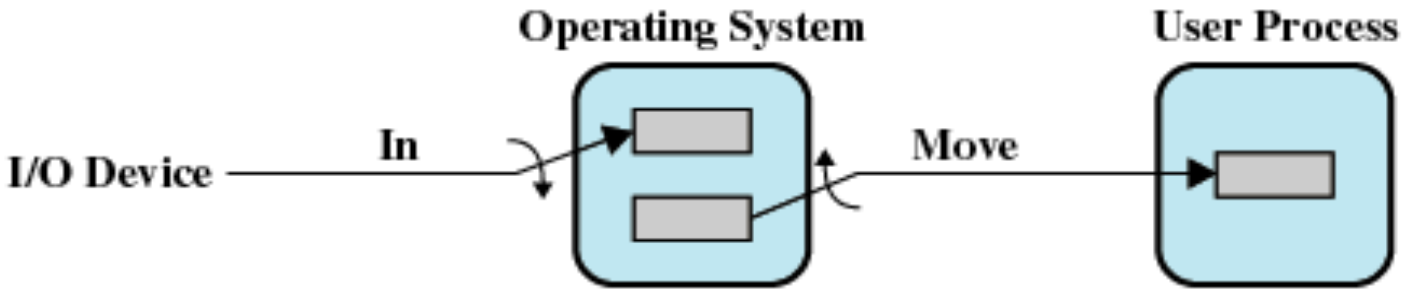
- Block-oriented
  - User process can process one block of data while next block is read in
  - Swapping can occur since input is taking place in system memory, not user memory
  - Operating system keeps track of assignment of system buffers to user processes

Single Buffer

- Stream-oriented
  - Used a line-at-a-time, e.g. line printer (or byte-at-a-time, e.g., form-mode terminals)
  - Output to the terminal is one line at a time

Double Buffer

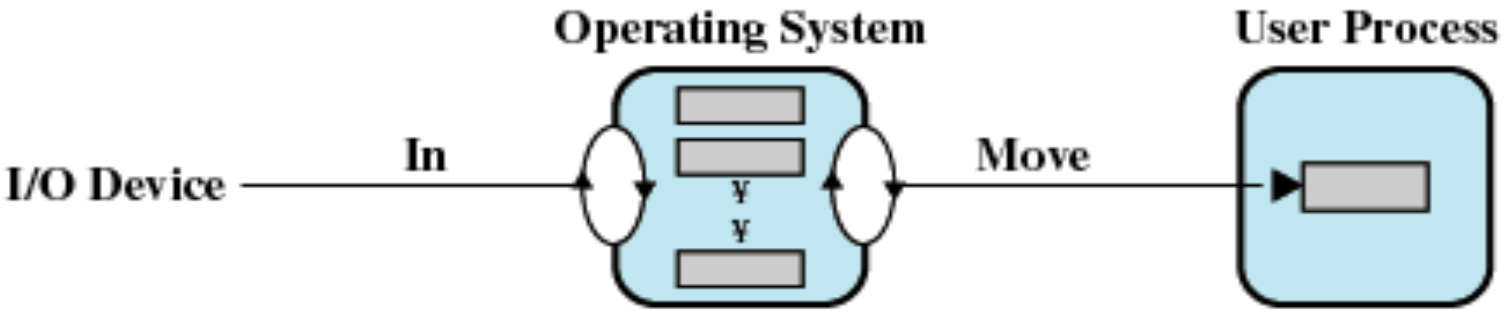
- Use two system buffers instead of one
- A process can transfer data to or from one buffer while the operating system empties or fills the other buffer



(c) Double buffering

Circular Buffer

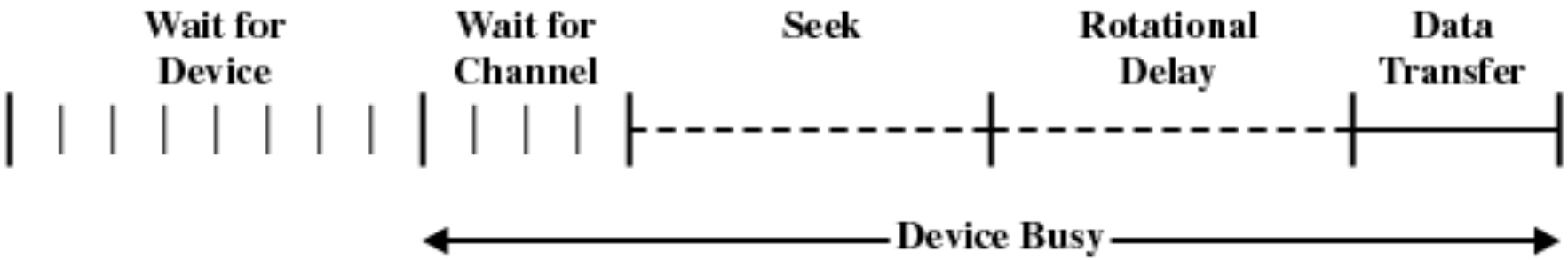
- More than two buffers are used
- Each individual buffer is one unit in a circular buffer
- Used when I/O operation must keep up with process



(d) Circular buffering

Disk Performance Parameters

- R/W: find track and sector and position the head
- Seek time
  - Time it takes to position the head at the desired track
- Rotational delay or rotational latency
  - Time its takes for the beginning of the sector to reach the head



Disk Performance Parameters

- Access time
  - Sum of seek time and rotational delay
  - The time it takes to get in position to read or write
- Data transfer occurs as the sector moves under the head

Disk Scheduling Policies

- Seek time is the reason for differences in performance
- For a single disk there will be a number of I/O requests
- If requests are selected randomly, we will have poor performance

Disk Scheduling Policies

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

Disk Scheduling Policies

- Priority
  - Goal is not to optimize disk use but to meet other objectives
  - Short batch jobs may have higher priority
  - Provide good interactive response time

Disk Scheduling Policies

- Last-in, first-out
  - Good for transaction 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

Disk Scheduling Policies

- Shortest Service Time First
  - 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
  - Problem: Always coming in nearby requests

Disk Scheduling Policies

- SCAN (elevator)
  - Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction
  - Direction is reversed
  - Max latency: 2t
- C-SCAN
  - Restricts scanning to one direction only
  - 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
  - Max latency: t + seek_time
  - Arm stickiness!

Disk Scheduling Policies

- N-step-SCAN
  - Disk requests divided into subqueues of length N
  - Subqueues are processed one at a time, using SCAN
  - New requests added to other queue when queue is processed
  - Big N →SCAN; N=1 → FIFO

- FSCAN
  - Two queues
  - One queue is empty for new requests

Disk Cache

- Buffer in main memory for disk sectors
- Contains a copy of some of the sectors on the disk

Disk Cache: Design issues

1) Deliver the data from cache to process
   a) Transfer the data from cache to process
   b) Use shared memory capability and pass pointers (readers/writers model)
2) Replacement strategy: Which block to replace?

Least Recently Used

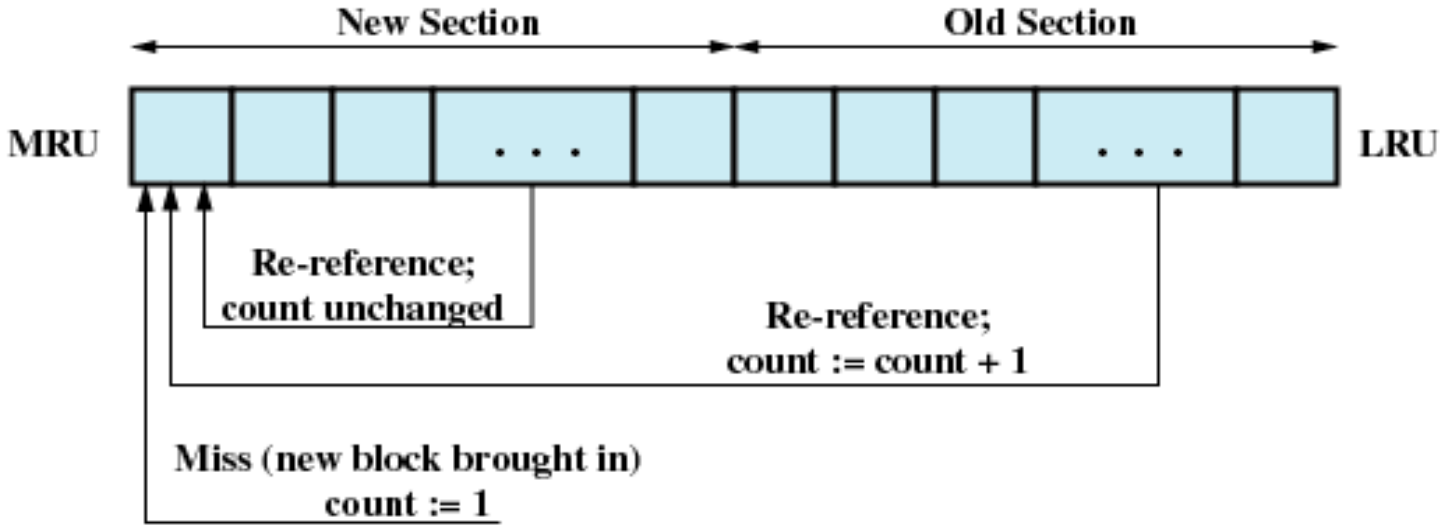
- Cache = Stack of blocks
- When a block is referenced or brought into the cache, it is placed on the top of the stack
- Replace the last block in the cache
- Blocks are not moved around:
  - Stack of pointers

Least Frequently Used

- A counter is associated with each block
- Counter is incremented each time block accessed
- Replace block with smallest count (fewest references)
- Counter misleading: blocks referenced many times in short periods

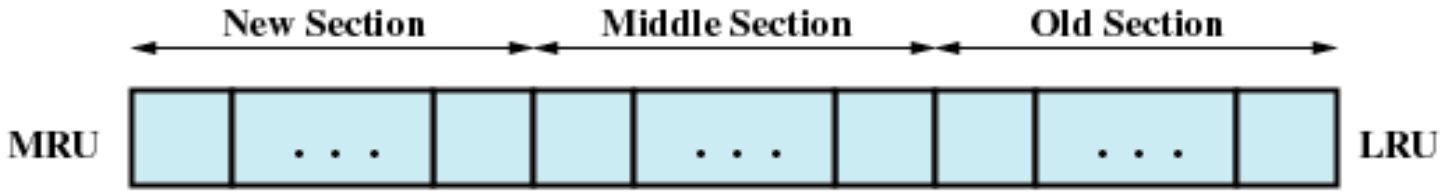
Frequency Based Replacement

- Two-section stack-like organization
- A counter is associated with each block
- The counter is incremented only when block goes from old section to new one
- Replace the old-section block with smallest counter
- Problem: New Section blocks do not have enough time to increment counter



Frequency Based Replacement

- **Three**-section stack-like organization
- A counter is associated with each block
- Counter changes for middle-section and old-section blocks
- Replace the old-section block with smallest counter
- The middle-section allows blocks to increment their counter before falling down to the old section



(b) Use of three sections

Replacement

- On-demand
  - Replace sector only when need the slot
  - Does not cluster writes
- Pre-planned
  - Release a number of slots each time
  - Good for clustering writes