Sheet10
IO MANAGEMENT AND DISK SCHEDULING

1) List and briefly define three techniques for performing I/O.
   - **Programmed I/O:** The processor issues an I/O command, on behalf of a process, to an I/O module; that process then busy-waits for the operation to be completed before proceeding.
   - **Interrupt-driven I/O:** The processor issues an I/O command on behalf of a process, continues to execute subsequent instructions, and is interrupted by the I/O module when the latter has completed its work. The subsequent instructions may be in the same process, if it is not necessary for that process to wait for the completion of the I/O. Otherwise, the process is suspended pending the interrupt and other work is performed.
   - **Direct memory access (DMA):** A DMA module controls the exchange of data between main memory and an I/O module. The processor sends a request for the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred.
2) What is the difference between logical I/O and device I/O?
   - **Logical I/O:** The logical I/O module deals with the device as a logical resource and is not concerned with the details of actually controlling the device. The logical I/O module is concerned with managing general I/O functions on behalf of user processes, allowing them to deal with the device in terms of a device identifier and simple commands such as open, close, read, write.
   - **Device I/O:** The requested operations and data (buffered characters, records, etc.) are converted into appropriate sequences of I/O instructions, channel commands, and controller orders. Buffering techniques may be used to improve utilization.
3) What is the difference between block-oriented devices and stream-oriented devices? Give a few examples of each.
   - **Block-oriented devices** stores information in blocks that are usually of fixed size, and transfers are made one block at a time. Generally, it is possible to reference data by its block number. Disks and tapes are examples of block-oriented devices.
   - **Stream-oriented devices** transfer data in and out as a stream of bytes, with no block structure. Terminals, printers, communications ports, mouse and other pointing devices, and most other devices that are not secondary storage are stream oriented.
4) Why would you expect improved performance using a double buffer rather than a single buffer for I/O?

   Double buffering allows two operations to proceed in parallel rather than in sequence. Specifically, a process can transfer data to (or from) one buffer while the operating system empties (or fills) the other.
5) What delay elements are involved in a disk read or write?

Seek time, rotational delay, access time.

6) Briefly define the disk scheduling policies illustrated in Figure 11.7.
   - **FIFO:** Items are processed from the queue in sequential first-come-first-served order.
   - **SSTF:** Select the disk I/O request that requires the least movement of the disk arm from its current position.
   - **SCAN:** The disk arm moves in one direction only, satisfying all outstanding requests en route, until it reaches the last track in that direction or until there are no more requests in that direction. The service direction is then reversed and the scan proceeds in the opposite direction, again picking up all requests in order.
   - **C-SCAN:** Similar to SCAN, but restricts scanning to one direction only. Thus, 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.
7) Briefly define the seven RAID levels.
   - **Level 0:** Non-redundant
   - **Level 1:** Mirrored; every disk has a mirror disk containing the same data.
   - **Level 2:** Redundant via Hamming code; an error-correcting code is calculated across corresponding bits on each data disk, and the bits of the code are stored in the corresponding bit positions on multiple parity disks.
   - **Level 3:** Bit-interleaved parity; similar to level 2 but instead of an error-correcting code, a simple parity bit is computed for the set of individual bits in the same position on all of the data disks.
   - **Level 4:** Block-interleaved parity; a bit-by-bit parity strip is calculated across corresponding strips on each data disk, and the parity bits are stored in the corresponding strip on the parity disk.
   - **Level 5:** Block-interleaved distributed parity; similar to level 4 but distributes the parity strips across all disks.
   - **Level 6:** Block-interleaved dual distributed parity; two different parity calculations are carried out and stored in separate blocks on different disks.

8) Considering Table 11.2

a) Perform the same type of analysis as that of Table 11.2 for the following sequence of disk track requests: 27, 129, 110, 186, 147, 41, 10, 64, 120. Assume that the disk head is initially positioned over track 100 and is moving in the direction of decreasing track number.

Disk head is initially moving in the direction of decreasing track number:

FIFO		SSTF		SCAN		C-SCAN	
Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed
27	73	110	10	64	36	64	36
129	102	120	10	41	23	41	23
110	19	129	9	27	14	27	14
186	76	147	18	10	17	10	17
147	39	186	39	110	100	186	176
41	106	64	122	120	10	147	39
10	31	41	23	129	9	129	18
64	54	27	14	147	18	120	9
120	56	10	17	186	39	110	10
Average	61.8	Average	29.1	Average	29.6	Average	38

b) Do the same analysis, but now assume that the disk head is moving in the direction of increasing track number.

If the disk head is initially moving in the direction of increasing track number, only the SCAN and C-SCAN results change:

SCAN		C-SCAN	
Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed
110	10	110	10
120	10	120	10
129	9	129	9
147	18	147	18
186	39	186	39
64	122	10	176
41	23	27	17
27	14	41	14
10	17	64	23
Average	29.1	Average	35.1

9) Calculate how much disk space (in sectors, tracks, and surfaces) will be required to store 300,000 120-byte logical records if the disk is fixed sector with 512 bytes/sector, with 96 sectors/track, 110 tracks per surface, and 8 usable surfaces. Ignore any file header record(s) and track indexes, and assume that records cannot span two sectors.

Each sector can hold 4 logical records. The required number of sectors is 300,000/4 = 75,000 sectors. This requires 75,000/96 = 782 tracks, which in turn requires 782/110 = 8 surfaces.

10) Consider the disk system described in the previous problem, and assume that the disk rotates at 360 rpm. A processor reads one sector from the disk using interrupt-driven I/O, with one interrupt per byte. If it takes 2.5 µs to process each interrupt, what percentage of the time will the processor spend handling I/O (disregard seek time)?

There are 512 bytes/sector. Since each byte generates an interrupt, there are 512 interrupts. Total interrupt processing time = 2.5 x 512 = 1280 µs. The time to read one sector is:

((60 sec/min) / (360 rev/min)) / (96 sectors/track)

= 0.001736 sec = 1736 µs

Percentage of time processor spends handling I/O:

(100) x (1280/1736) = 74%

11) Repeat the preceding problem using DMA, and assume one interrupt per sector.

With DMA, there is only one interrupt of 2.5 µs. Therefore, the percentage of time the processor spends handling I/O is

(100) x (2.5/1736) = 0.14%

12) It should be clear that disk striping can improve the data transfer rate when the strip size is small compared to the I/O request size. It should also be clear that RAID 0 provides improved performance relative to a single large disk, because multiple I/O requests can be handled in parallel. However, in this latter case, is disk striping necessary? That is, does disk striping improve I/O request rate performance compared to a comparable disk array without striping?

It depends on the nature of the I/O request pattern. On one extreme, if only a single process is doing I/O and is only doing one large I/O at a time, then disk striping improves performance. If there are many processes making many small I/O requests, then a nonstriped array of disks should give comparable performance to RAID 0.

13) Consider a 4-drive, 200 GB-per-drive RAID array. What is the available data storage capacity for each of the RAID levels, 0, 1, 3, 4, 5, and 6?

RAID 0: 800 GB	RAID 4: 600 GB
RAID 1: 400 GB	RAID 5: 600 GB
RAID 3: 600 GB	RAID 6: 400 GB