Q1.

1. I/O control methods can be classified as (1)_Programmed I/O_(2)_Direct Memory Access__.
2. Each physical record on the disk has a unique address that consists of three parts: (1)__head identifier__(2)_track_identifier_(3)_sector identifier_.
3. Data READ/WRITE time = (1)_seek time_ + (2) _rotational latency_ +(3)_transfer time__.
4. The metric for measuring I/O performance are (1)_response time__,(2)__throughput_.

Q2. **What are the work steps of the DMA controller? Please answer it and briefly describe the process of each step.**

DMA is Direct Map Access which gives controller access to memory bus and asks it to transfer data blocks to/from memory directly.

5. Device driver is told to transfer disk data to buffer at address X.
6. Device driver tells disk controller to transfer C bytes form disk to buffer at address X.
7. Disk controller initiates DMA transfer.
8. Disk controller sends each byte to DMA controller.
9. DMA controller transfers bytes to buffer X, increasing memory address and decreasing C until C = 0.
10. When C = 0, DMA interrupts CPU to signal transfer completion.

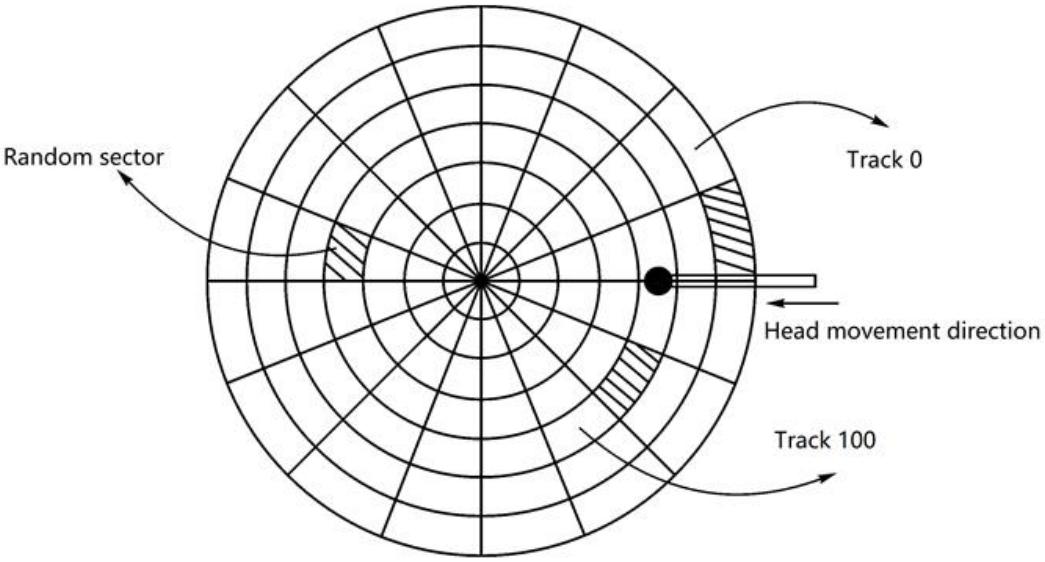
Brief explanation: Firstly, CPU tells device driver and device driver tells disk controller. Then when disk controller starts to initiate DMA transfer and sends bytes to DMA controller while DMA controller transfers bytes to buffer X, CPU is able to focus on other things. After DMA interrupts CPU, CPU is again back to this job and can read data from address X.

Q3. **Under what circumstance, the SSTF algorithm will cause starvation?**

SSTF refers to shortest seek time first. It selects the request with the minimum seek time from the current head position.

It will cause starvation when there is a request which is far away from the current head position while lots of requests with shorter seek time from the current head position are there or are created continually. The request with long distance can not be replied in this way and it will cause starvation.

Q4. **Suppose the computer system uses the disk as the following:**



Let a single-sided disk rotation speed be 12000r/min, each track has 100 sectors, 200 tracks in total, and the average movement time between adjacent tracks is 1 ms.

If at some point, the head is located at track 100 and moves in the direction in which the track number increases, the track number request queue is 70, 90, 30, 120, 20, 60. For each track in the request queue, a randomly distributed sector is read.

a) If the C-SCAN algorithm is used to read the six sectors,

(1) Write the track access sequence (5 points)

The request queue after sort is 20, 30, 60, 70, 90, 120. The range for the track number is from 0 to 199. The default moving direction is from small track number to large number. So under C-SCAN, the sequence is 100, 120, 199, 0, 20, 30, 60, 70, 90.

(2) How much time is required in total? The calculation process is required. (15 points).

418ms.

Data READ/WRITE time = seek time + rotational latency + transfer time.

Seek time is used to position the head/arm over the proper track. The number of tracks we need to go across is |199 - 100| + |0 - 199| + |90 - 0| = 388. Since the average movement time between adjacent tracks is 1 ms, the seek time is 388ms.

Rotational latency is that caused by waiting for desired sector to rotate under r/w head. Since a randomly distributed sector is read for each track, the whole track should be traveled to guarantee that we can get the correct sector. Since the rotation speed is 12000r/min which corresponds to 5ms per rotation. So the rotational latency is 5ms*6=30ms.

Transfer time is the cost for transferring a block of bits(sector) under r/w head. In this situation, it is ignored since we already counted the cost when we consider the whole track rotational latency for each request.

So the total time is 388ms + 30ms = 418ms.

修改： 388 + 15 ms

b) If using SSD, which scheduling algorithm do you think should be used, and explain why? (10 points)

FIFO should be used since the seek time and rotational latency can both be eliminated if using SSD.(Latency for SSD is queuing time + controller time +xfer time.) So there is no necessity to use other algorithms for optimizing while FIFO is the simplest.