**IO Review**

1. What is memory mapped I/O?

Memory mapped I/O is a mechanism of performing I/O operations between Central Processing Unit(CPU) and other peripheral devices in a computer. In Memory mapped I/O, it uses same address bus for addressing both memory and Input/ Output devices. Hence when an address is being accessed by the CPU, it might be pointing to a part of physical RAM, but it may also point to memory of the Input/ Output device. Therefore, the CPU command used for accessing the memory can also be used to accessing devices.

1. Why is DMA an improvement over CPU programmed I/O?

DMA is an Improvement of CPU programmed I/O because Direct Memory Access engine doesn't holds the memory bus for the entire duration of transfer, instead it grabs it only when some specific data needs to be transferred. It means CPU access to memory is affected, but not complete shut off. In the meantime, CPU can also work with data in parallel using its cache.

1. When would DMA transfer be a poor choice?

DMA transfer can be bad option in case of Burst Mode data transfer because in this case, the CPU is relinquished inactive for quite long periods of time. In general, DMA requests have higher priority than all other bus exertion, including interrupts. Therefore, no interrupts can be respected during a DMA cycle.

1. Disk Technology. Suppose we have a magnetic disk (resembling an IBM Microdrive) with the following parameters:

Average seek time 12 ms

Rotation rate 3600 RPM

Transfer rate 3.5 MB/second

Sectors per track 64

Sector size 512 bytes

Controller overhead 5.5 ms

Answer the following questions. (Note: you may leave any answer as a fraction.)

(a) What is the average time to read a single sector?

Disk Access Time = seek time + rotational delay + transfer time + controller overhead

rotational delay = rotation rate / 2

Rotation rate = 3600, thus time for one rotation

= 1min \* 60 seconds / 3600 = 0.016second = 16.66ms per rotation

Rotational delay for one disk = 1/ 2 \* 16.66ms = 8.33 = 30.000/RPM = (1 / (RPM / 60)) \* 0.5 \* 1000

Transfer time = (512 / 3.5\*2^20)\*1000 = x ms

= 12 + 8.33 + 0.14 + 5.5 = 25.97ms

b) What is the average time to read 8 KB in 16 consecutive sectors in the same cylinder?

Transfer time = (8 \* 1024 / 3.5\*2^20)\*1000 = x ms

= 12 + 8.33 + 2.24 + 5.5 = 28.07ms

c) Now suppose we have an array of 4 of these disks. They are all synchronized such that the arms on all the disks are always on the same sector within the track. The data is striped across the 4 disks so that 4 logically consecutive sectors can be read in parallel. What is the average time to read 32 consecutive KB from the disk array?

4 disks -> each disk can read a sector at a time, total memory can read at a time = 4 \* 512 (size of sector) = 2KB.

To read 32KB in 4 disks, need to read 8k in each disk 8k = 8 \* 1024 / 512 = 16 sectors

5. What is the average time to read or write a 512-byte sector for a typical disk rotating at 7200 RPM? The advertised average seek time is 8ms, the transfer rate is 20MB/sec, and the controller overhead is 2ms. Assume that the disk is idle so that there is no waiting time.

Access Time = seek time + rotational delay + transfer time + controller overhead

8ms+(0.5/(7200/60))\*1000 ms+0.5KB/20.0MB/s+2ms=8+4.15+0.025+2=14.175ms.

1. A program repeatedly performs a three-step process: It reads in a 4-KB block of data from disk, does some processing on that data, and then writes out the result as another 4-KB block elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 7200RPM, has an average seek time of 8ms, and has a transfer rate of 20MB/sec. The controller overhead is 2ms. No other program is using the disk or processor, and there is no overlapping of disk operation with processing. The processing step takes 20 million clock cycles, and the clock rate is 400MHz. What is the overall speed of the system in blocks processed per second assuming no other overhead?

The average rotational delay: t=0.5/7200=4,167

Read time: 4kb/20mb=0,195ms

Total read time: 4,167+8+2+0,195=14,36 ms

Processing of Data = 20million clock cycle/400mHz = 50ms

Writing of data to disk = Readtime + writing time +processing

= (2x14,36) + 50

=78,72 ms. To process 1 block

-🡪 12.7 block per sec.

7. How much cylinder skew is needed for a 7200-rpm disk with a track-to-track seek time of 1 msec? The disk has 200 sectors of 512 bytes each on each track.

The disk rotates at 120 RPS, so 1 rotation takes 1000/120 msec.

With 200 sectors per rotation,

the sector time is 1/200 of this number or 5/120 = 1/24 msec.

Thus the cylinder skew should be 24.

8. Disk requests come in to the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 6 msec per cylinder moved. How much seek time is needed for

1. (a) First-Come, first served.
2. (b) Closest cylinder next.
3. (c) Elevator algorithm (initially moving upward).

In all cases, the arm is initially at cylinder 20.

(a).FCFS:

Total Head Movement: 10+12+20+38+34+32=142

Total seek tine:142\*6=276 (ms)

(b)Closest Cylinder next: 20->22->10->6->38->40.

Total Head Movement: ((22-20)+(22-10)+(10-6)+(6-4)+(38-6)+(40-38)) = 60

Total Seek Time: 60\*6 = 360(msec)

(C) elevator algorithm

Total Head Movement: 2+16+2++30+4+4=58

Total seek tine:58\*6= 348;