Date:_____

Quiz for Chapter 6 Storage and Other I/O Topics

Not all questions are of equal difficulty. Please review the entire quiz first and then budget your time carefully.

Name:_____

Course:_____

Solutions in Red

1. [6 points] Give a concise answer to each of the following questions. Limit your answers to 20-30 words.

(a) What is memory mapped I/O?

An I/O scheme in which portions of address space are assigned to I/O devices and reads and writes to those addresses are interpreted as commands to the I/O device.

(b) Why is DMA an improvement over CPU programmed I/O?

DMA is a mechanism that provides a device controller the ability to transfer data directly to or from the memory without involving the processor. This allows the CPU to perform arithmetic and other instructions while the DMA is going on in parallel

(c) When would DMA transfer be a poor choice?

DMA is not useful when the amount of data to transferred between memory and the I/O device is very less. In this case the overhead of setting up the DMA transfer would outweigh the benefits of direct data transfer without the interference of the processor.

2. [6 points] Mention two advantages and disadvantages for using a single bus as a shared communication link between memory, processor and I/O devices.

Advantages:
(1) *Versatility* – single connection scheme for easy add-ons (2) *Low Cost* – single set of wires shared in multiple ways

Disadvantages:
(1) Communication bottleneck, bandwidth limits the maximum I/O throughput. (2) Devices will not always be able to use the bus when they need to.

Name:_____

3. [6 points] 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
= 12 + (0.5*60*10^3/3600) + (512/(3.5*2^{20}))*1000 + 5.5
= 25.97 ms

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

Only the transfer time gets changed.

Disk Access Time = seek time + rotational delay + transfer time + controller overhead
= 12 + (0.5*60*10^3/3600) + (8*1024/(3.5*2^{20}))*1000 + 5.5
= 28.07 ms

(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?

Since 4 logically consecutive sectors can be read at once, we can read off 2 KB at once. To read 32 KB, we need to read 16 sectors on each disk. So, the time taken is the same as in (b)

4. [6 points] Answer the following questions:

(a) 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.

Disk Access Time = seek time + rotational delay + transfer time + controller overhead
= 8 + (0.5*60*1000/7200) + (512/20*2^{20})*1000 + 2
= 14.17 ms

(b) 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?

Disk Read Time for a 4KB block
= seek time + rotational delay + transfer time + controller overhead
= 8 + (0.5*60*1000/7200) + (4*1024/20*2^{20})*1000 + 2
= 14.17 ms

Processing Time = 20 * 10^6 * (1/(400*10^6))
= 1/20 = 0.05 s = 50 ms

Disk Write Time for 4 KB block = 14.17 ms

Total time to completely process a 4 KB block = 2*14.17 + 50 = 78.34 ms

Number of blocks processed per second = 1000/78.34 = 12.76

5. [6 points] What is the bottleneck in the following system setup, the CPU, memory bus, or the disk set?

- The user program continuously performs reads of 64KB blocks, and requires 2 million cycles to process each block.
- The operating system requires 1 million cycles of overhead for each I/O operation.
- The clock rate is 3GHz.
- The maximum sustained transfer rate of the memory bus is 640MB/sec
- The read/write bandwidth of the disk controller and the disk drives is 64MB/sec, disk average seek plus rotational latency is 9ms.
- There are 20 disks attached to the bus each with its own controller. (Assume that each disk can be controlled independently and ignore disk conflicts.)

Amount of time CPU takes to process each 64 KB block
= ((2 * 10^6)/ (3 * 10^9)) * 10^3
= 0.67 ms

Amount of time spent in memory transfer for 64 KB block
= ((64 * 2^{10})/(640*2^{20})) * 1000 = 0.097 ms

Amount of time spent in I/O transfer for a 64 KB block
= seek time + rotational delay + transfer time + controller overhead
= 9 + ((64*2^{10})/(64*2^{20}))*10^3 + (((10^6 * 3)/10^9)*10^3
= 12.97 ms

The main bottleneck is I/O in the above system.

6. [6 points] Discuss why RAID 3 is not suited for transaction processing applications. What kind of applications is it suitable for and why?

RAID 3 is unsuited to transactional processing because each read involves activity at all disks. In RAID 4 and 5 reads only involve activity at one disk. The disadvantages of RAID 3 are mitigated when long sequential reads are common, but performance never exceeds RAID 5. For this reason, RAID 3 has been all but abandoned commercially.

Name:_____

7. [7 points] Suppose we have two different I/O system A and B. A has data transfer rate: 5KB/s and has access delay: 5 sec. While B has data transfer rate: 3 KB/s and has access delay: 4 sec. Now we have a 3M I/O request, taking performance into consideration, which I/O system will you use? What about for a 3KB request?

3M request
Case 1:
t (in sec) = 5 + (3 * 1024 * 1024) / (5 * 1024) = 5 + 614.4 = 619.4
Case 2:
t (in sec) = 4 + (3 * 1024 * 1024) / (3 * 1024) = 4 + 1024 = 1028

So system 1 will be chosen.

3K request
Case 1:
t (in sec) = 5 + (3 * 1024) / (5 * 1024) = 5.6
Case 2:
t (in sec) = 4 + (3 * 1024) / (3 * 1024) = 5

So system 2 will be chosen.

8. [7 points] If a system contains 1,000 disk drives, and each of them has a 800,000 hour MTBF, how often a drive failure will occur in that disk system? Could you give some idea to improve that? And why will your idea work?

MTBF (array) = MTBF (one disk) / Number of disks in array

So, answer is 800,000 / 1000 = 800 hrs

Set them up in a RAID 5 configuration which involves distributing the data and parity bits across the disk drives. This way even when one drive fails, the parity information enables the disk array to continue operation and rebuild the failed drive online once it has been replaced. Now, the likelihood that a second drive will fail before the failed drive is restored is pretty low. Hence the reliability can be shown to increase dramatically. To further improve the reliability, one can have a spare disk which is hot-swapped with the failed disk in order to reduce the restoration time.

9. [6 points] What is the average time to read a 512 byte sector for Seagate ST31000340NS in Figure 6.5? What is the minimum time? Assume that the controller overhead is 0.2 ms, and the disk is idle so that there is no waiting time.

Average =
Average seek time + Average rotational delay + Transfer time + controller overhead
= 8.5 + 0.5 rotations/7200RPM + 0.5KB/105MBps + 0.2
= 8.5+4.17+0.005+0.2
= 12.875ms

Minimum:
Minimum seek time + Minimum rotational delay + Transfer time + controller overhead
= 0.8 + 0 rotations + 0.5KB/105MBps + 0.2
= 0.8+0+0.005+0.2
= 1.005ms

Name:_____

10. [4 points] How many times can you store a 4MB song at your 1GB NOR flash memory in Figure 6.7 before the first wear out if wear leveling working ideally?

100,000 * (1GB/4MB) = 25,000,000
So, you can rewrite a 4MB song about 25,000,000 times ideally

11. [6 points] In Figure 6.8 which fields are correlated with each other? Why do these correlations exist?

Correlation 1: Each external bus supports multiple devices per channel whereas each internal device supports only a single device per channel. The reason for this correlation is that internal devices are rarely added by end users and can be only few in number due to space constraints. By contrast many external devices may be added although a computer can only have a few external ports. Outside of a computer there is enough room for additional hub devices.

Correlation 2: External devices have longer maximum bus length.

12. [6 points] In Figure 6.9, PCI-E connections are available from both the north bridge and the south bridge. What are the advantages and disadvantages to attaching devices to the PCI-E connections on the north and south bridges?

The advantages of connecting PCI-E to the north bridge is that the interface is 'closer' to the processor and memory. That is to say, fewer hops from chip-to-chip must take place for data exchange, and as a result less competition for the data. This is illustrated in figure 6.9---PCI-E on north bridge has 4GB/sec bandwidth, though PCI-E on south bridge has 1GB/sec. This is also a disadvantage of the south bridge.

The advantage of connecting PCI-E to the south bridge is that, by lowering the priority of the PCI-E ports, we have raised the priority of the memory-to-processor connection. In many circumstances, memory throughput is of higher priority than I/O devices.

13. [6 points] Section 6.7 focuses on transactional processing as an example of a disk IO intensive application. Give another example of a disk IO intensive application compare and contrast the performance requirements and consider how different disk implementations (magnetic media, flash memory, or MEMS device) can be more or less appropriate for different applications.

Streaming media is an example of an IO intensive application. Magnetic media would be most suitable since the accesses are expected to be linear and magnetic media can take advantage of spatial locality. MEMS devices benefit very little from spatial locality, and flash memory does not benefit at all.

14. [6 points] Imagine that you are proposing a new disk IO benchmark for transaction processing, what sort of experiments would you perform to show that your benchmark's results are meaningful. Imagine that you are reviewing a paper introducing a new disk IO benchmark for transaction processing. What sort of subtle flaws would you search for?

You should verify that the disk access trace for the benchmark strongly resembles the traces for real applications. In particular, you should make sure that not only average metrics are good matches, but also check that the extremes are good matches, and that burstiness in the real applications load is accounted for. Important factors to test for are spatial and temporal locality, frequency of transaction requests, and the amount of data read or written per transaction. Failure to take any of these

considerations into account can lead to a flawed benchmark. For this reason, some industry benchmarks are real applications processing real input sets.

15. [6 points] Which of the following would be an acceptable transport medium for real-time transmission of human voice data? Which would be "overkill"?

- 56.5Kbps modem
- 100 Base-T Ethernet connection
- 802.11b wireless connection.

A 56.5Kbps modem is adequate, the others are overkill. Perhaps the question could be made more challenging by having the reader estimate the number of simultaneous phone conversations that could be had over each of the media. Alternatively, the question could be reworded to highlight the differences between bandwidth and latency.

16. [10 points] A given computer system includes a hard disk with direct memory access (DMA).

(a) Suppose a user application needs to change a single byte within a disk block. Sketch, in order, all communications that must take place between the processor and the hard drive to complete this operation.

Both a read and a write must occur. (1) The os tells the disk to perform a read by providing the disk address and memory address to a mutually readable location and issuing an interrupt to the disk. (2) The disk performs the read. (3) The disk writes the result (via DMA) to the appropriate memory location. (4) The disk interrupts the processor to signal completion. (5) The application/os modifies the in-memory copy of the disk block. (6) The os tells the disk to perform a write by providing the disk address and memory address to a mutually readable location and issuing an interrupt to the disk. (7) The disk reads the addresses, and then transfers the block from memory via DMA. (8) The performs the write. (9) The disk interrupts the processor to signal completion.

(b) Assume the total time required to perform a read of n blocks from a hard disk is T(n) = 250ms + 100ms * n. Further, assume that for any read or write to a hard disk block a, there is a probability p = 0.75 that hard disk block a+1 will be read soon afterwards. Given that an application has requested a read of a single disk block, the OS can expect the application to read subsequent blocks later. If the OS will pursue a strategy of reading n blocks at a time, analyze how the OS can choose this n in order to minimize the expected read time.

There is a setup cost associated with each read, thus it is beneficial to batch reads. First, let's analyze the probabilities of different read lengths. The probability that at least n blocks will be read can be figured inductively; i.e. P(at least n blocks| at least
n-1 blocks) = p, and P(at least 1 block) = 1. Thus, P(at least n blocks) = P(at least n blocks | at least n-1 blocks) * ... * P(at least 2 blocks| at least 1 block) * P(at least 1 block), or P(at least n blocks) = p**(n-1).

Similarly the probability that we will read exactly n blocks is,

P(exactly n blocks | at least n blocks) = 1-p
P(exactly n blocks) = P(exactly n blocks | at least n blocks) * P(at
least n blocks) = (1-p)p**(n-1)

Name:_____

Now, for the second half, we compare the time to perform the read given our predictions are correct to the time required to perform the read. Consider that any number of reads m is possible, independent of our n-block read strategy. The total time required to perform an m-block read as a number of n-block reads, a la

T_n(m) = ceil(m/n) * T(n)

And our expected read time is:

E(n) = SUM from m=1 to inf of T_n(m) * P(exactly m blocks)

Now, we choose n to minimize E(n). The optimal read strategy is 4.