# 2013 - 2014 BSS Practicum

## Solutions Week 5

## Chapter 11

**Exercise 1.1 Linked Allocation**
What are the advantages of the variation of linked allocation that uses a FAT to chain together the blocks of a file?

**Answer 1.1, see 11.2**
The advantage is that while accessing a block that is stored at the middle of a file, its location can be determined by chasing the pointers stored in the FAT as opposed to accessing all of the individual blocks of the file in a sequential manner to find the pointer to the target block. Typically, most of the FAT can be cached in memory and therefore the pointers can be determined with just memory accesses instead of having to access the disk blocks.

## Chapter 12

**Exercise 1.2**
Could a RAID Level 1 organization achieve better performance for read requests than a RAID Level 0 organization (with nonredundant striping of data)? If so, how?

**Answer 1.2**
Yes, a RAID Level 1 organization could achieve better performance for read requests. When a read operation is performed, a RAID Level 1 system can decide which of the two copies of the block should be accessed to satisfy the request. This choice could be based on the current location of the disk head and could therefore result in performance optimizations by choosing a disk head that is closer to the target data.

## Chapter 13

**Exercise 1.3**
1.2 What are the advantages and disadvantages of supporting memory-mapped I/O to device-control registers?

**Answer 1.3**
The advantage of supporting memory-mapped I/O to device-control registers is that it eliminates the need for special I/ O instructions from the instruction set and therefore also does not require the enforcement of protection rules that prevent user programs from executing these I/O instructions. The disadvantage is that the resulting flexibility needs to be handled with care; the memory translation units need to ensure that the memory addresses associated with the device control registers are not accessible by user programs in order to ensure protection.

**Exercise 1.4**
Describe three circumstances under which blocking I/O should be used. Describe three circumstances under which nonblocking I/O should be used. Why not just implement nonblocking I/O and have processes busy-wait until their device is ready?

**Answer 1.4**
Generally, blocking I/O is appropriate when the process will be waiting only for one specific event. Examples include a disk, tape, or keyboard read by an application program. Non-blocking I/O is useful when I/O may come from more than one source and the order of the I/O arrival is not predetermined. Examples include network daemons listening to more than one network socket, window managers that accept mouse movement as well as keyboard input, and I/O-management programs, such as a copy command that copies data between I/O devices. In the last case, the program could optimize its performance by buffering the input and output and using non-blocking I/O to keep both devices fully occupied.
Non-blocking I/O is more complicated for programmers, because of the asynchronous rendezvous that is needed when an I/O occurs. Also, busy waiting is less efficient than interrupt-driven I/O so the overall system performance would decrease.

## Chapter 16

**Exercise 1.5**
Explain why doubling the speed of the systems on an Ethernet segment may result in decreased network performance. What changes could help solve this problem?

**Answer 1.5**
Faster systems may be able to send more packets in a shorter time. The network would then have more packets traveling on it, resulting in more collisions, and therefore less throughput relative to the number of packets being sent. More networks can be used, with fewer systems per network, to reduce the number of collisions.

**Exercise 1.6**
The original HTTP protocol used TCP/IP as the underlying network protocol. For each page, graphic, or applet, a separate TCP session was constructed, used, and torn down. Because of the overhead of building and destroying TCP/IP connections, performance problems resulted from this implementation method. Would using UDP rather than TCP be a good alternative? What other changes could you make to improve HTTP performance?

**Answer 1.6**
Despite the connectionless nature of UDP, it is not a serious alternative to TCP for the HTTP. The problem with UDP is that it is unreliable, documents delivered via the web must be delivered reliably. (This is easy to illustrate - a single packet missing from an image downloaded from the web makes the image unreadable.)
One possibility is to modify how TCP connections are used. Rather than setting up - and breaking down - a TCP connection for every web resource, allow persistent connections where a single TCP connection stays open and is used to deliver multiple web resources.