

BSS Practicum Week 4 Exercises

Andre Nanninga

26-may-2014

Chapter 11

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

The advantage is that you can access a block in the middle of the file without going through each block from the start until you find it in the linked list. The FAT is usually kept in memory meaning that a request for a block in the middle of a file is even faster.

Chapter 12

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?

No, a RAID 0 organization must use every disk in its array for read requests and will therefore have a higher rate of reading. A RAID 1 organization will only use one of its array for reading. The only increase in speed a RAID 1 could have is in choosing which disk to read from depending on the distance between the reader head and the block.

Chapter 13

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

An advantage is that no special I/O instructions are needed and also doesn't need protection rules that prevent users from executing these I/O instructions.

This flexibility must however be handled with care, without these protection rules any program *could* access the device control registers and would therefor be a security threat.

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?

Blocking I/O should be used when a process is waiting for a single specific shortterm event, for example keyboard input, mouse movement or touch gestures. For programmers it's also easier to use blocking I/O instead of refactoring code to make use of asynchronous methods.

Nonblocking I/O should be used when a process needs to handle other business while waiting for an I/O event. One example would be a webserver which should still serve existing connections while waiting for new connection to arrive via a I/O event.

Busy-wait is nearly always more expensive in terms of CPU usage than using nonblocking I/O and should therefore be avoided.

Chapter 16

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?

Increasing the speed of an Ethernet segment can result in a higher amount of collisions which means that packages will be dropped and must be resent. The resending of the packages is costly and can slow down your overall connection.

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?

UDP would not be a good alternative to TCP. UDP is inherently unreliable and will not resend dropped packages. This makes it ideal for gaming or video streaming where a single dropped package isn't an issue because newer packages will override it anyway. For HTTP however we want every package to be delivered reliably, afterall a article with the second paragraph

missing isn't very user friendly. Therefor using TCP for HTTP is the best option.