

CSCE 351 Operating System Kernels

Fall 2001
Steve Goddard

Homework 2, November 15

Due: 12:30pm November 27

- 1) (5 points) If a disk controller writes the bytes it receives from the disk to memory as fast as it receives them, with no internal buffering, is interleaving conceivably useful? Explain.

- 2) (10 points) Consider a disk that is doubled interleaved, as in Fig 3-4(c) of your text. It has eight sectors of 512 bytes per track and a rotation rate of 300 rpm.

How long does it take to read all the sectors of a track in order, assuming the arm is already correctly positioned, and $\frac{1}{2}$ rotation is needed to get sector 0 under the head?

What is the data transfer rate from the disk (to the controller)?

- 3) (10 points) Now consider a non-interleaved disk with the same characteristics described in the previous problem (except the interleaving, of course).

How long does it take to read all the sectors of a track in order, assuming the arm is already correctly positioned, and $\frac{1}{2}$ rotation is needed to get sector 0 under the head?

What is the data transfer rate from the disk (to the controller)?

- 4) (5 points) How much does the transfer data rate degrade due to interleaving?

Given the performance degradation, why is disk interleaving commonly used?

- 5) (10 points) The DM-11 terminal multiplexer, which was used on the (now ancient) PDP-11, sampled each (half-duplex) terminal line at seven times the baud rate to see if the incoming bit was a 0 or a 1. Sampling the line took 5.7 microseconds. How many 1200-baud lines could the DM-11 support?

- 6) (10 points) A local area network is used as follows. The user issues a system call to write data packets to the network. The OS then copies the data to a kernel buffer. Then it copies the data to the network controller board. When all the bytes are safely inside the controller, they are sent over the network at a rate of 10 megabits/sec. The receiving network controller stores each bit a microsecond after it is sent. When the last bit arrives, the destination CPU is interrupted, and the kernel copies the newly arrived packet to a kernel buffer to inspect it. Once it has figured out which user the packet is for, the kernel copies the data to the user's space.

Assume that each interrupt and its associated processing takes 1 millisecond, that packets are 1024 bytes (ignore headers), and that copying a byte takes 1 microsecond. Further, assume that the send is blocked until the work is finished at the receiving side and an acknowledgement comes back. For simplicity, however, assume that the time to get the acknowledgement back is so small it can be ignored.

What is the maximum rate at which one process can pump data to another?

- 7) (10 points) The message format of Figure 3-15 in your text is used for sending request messages to block device drivers. Which fields, if any, could be omitted for messages to character device drivers? Explain.
- 8) (10 points) Disk requests come in to the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 5 msec per cylinder moved. How much seek time is needed for
 - A) First-Come, First Served (FCFS).
 - B) Shortest Seek First (SSF).
 - C) Elevator algorithm (initially moving upward).
- 9) (10 points) The file system task is the only process that is supposed to send a message to block device drivers. As such, lines 9165 to 9175 of the code listing in the text check to make sure that this restriction holds.

What happens if a user process accidentally sends a message to a block device driver (task)?

Describe a solution to this problem (i.e., how could the code be changed so that a mistaken message is handled more gracefully).

- 10) (10 points) Consider the indirect routine `*dr_schedule`.

What does this routine do for a RAM disk?

What does this routine do for a normal hard disk?

- 11) (5 points) What does the routine `clock_mess( )` do. Why is it used?
- 12) (5 points) What does a call to `enable_iop( )` do? Why is it used?