

CSE 120, Answers to HW4

Chapter 5

Q5

Let's first see how many memory words need to be transferred to memory per interrupt:

this is 34 total words if we assume each register holds exactly one memory word.

At a speed of 1/10 words/nsec, it takes $34 \cdot 10 = 340$ nsec per interrupt to save state. We can therefore have at most $10^9/340 = 2.94 \cdot 10^6$ interrupts handled per second.

Q10

(a) Device drivers

Note: On a lot of systems a big part of this work is done by the device controller, but the only part of the OS aware of this work is the device driver.

(b) Device drivers

(c) Device-independent OS software

(d) User-level I/O software

Q12

Here we are interested in finding the total time elapsed from when a process issues a system call to send a packet, to when the acknowledgment comes back and it is ready to send another packet.

First the packet has to be copied twice: $2 \cdot 1024$ usec.

It has to be sent over: $1024 \cdot 8 / 10^7 = 8192 \cdot 10^{-7}$ sec.

There is a delay of 1 msec per bit for storage on the receiver's NIC : 8192 msec.

The receiver's NIC then issues an interrupt : 1 msec.

The packet is copied onto the receiving process's space: 1024 usec.

The ACK is sent back to the sender, an interrupt is generated : 1 msec.

So it takes a total of $3 \cdot 1.024 + 0.8192 + 8194$ msec ≈ 8197.89 msec so the transfer rate is about 8192bits/8197.89 msec ≈ 1 Kilobit/sec .

Q13

By having a spooling directory access to the printer can be controlled centrally in order to ensure a fair access to the resource for all processes. This would in effect keep any single process from holding the printer indefinitely.

Q17

This is the probability that 2 or more drives have failed within a given hour. Assuming these individual drive failures are mutually independent, the probability we're looking for is:

$$P(2 \text{ or more drives failing}) = \sum_{x=2}^k \binom{k}{x} p^x (1-p)^{k-x} = 1 - (1-p)^k - np(1-p)^{k-1}.$$

Note:

The variable x above simply represents the number of drives that have failed, and $\binom{k}{x}$ is the number of possible ways x drives could have failed.

Q24

(a) **First-come, first served**

Service Order	20	10	22	20	2	40	6	38
# of cylinders moved	Initial	10	12	2	18	38	34	32

Total seek time: $146 * 6 \text{ msec} = 876 \text{ msec}$.

(b) **Closest cylinder next**

Service Order	20	20	22	10	6	2	38	40
# of cylinders moved	Initial	0	2	12	4	4	36	2

Total seek time: $60 * 6 \text{ msec} = 360 \text{ msec}$.

(a) **Elevator algorithm** (moving upward)

Service Order	20	20	22	38	40	10	6	2
# of cylinders moved	Initial	0	2	16	2	30	4	4

Total seek time: $58 * 6 \text{ msec} = 358 \text{ msec}$.

Q27

The clock ticks 60 times per second, each time requiring 2 msec of the CPU time, so the fraction of the CPU devoted to the clock is $(60 * 2 \text{ msec}) / (1 \text{ sec}) = 0.12$.

Q45

Reducing the voltage by a factor of n , would reduce the CPU clock speed (and thus the CPU speed) by a factor of n .

If a user types at 1 char/sec it is then acceptable to reduce the CPU speed to processing 1 char every second. At the current voltage the CPU is able to process 1 char every 100 msec, so reduce the voltage by $n = 10$ is acceptable. This corresponds to a 99% energy savings.

Chapter 8

Q3

At 200 MIPS (200 million instructions/sec) per CPU, 2 CPU's will saturate a 400MHz (400 million cycles/sec) bus if every memory reference requires a bus cycle.

If instead caching is used and each CPU only miss in cache 10% of the time, only 10% of 200 million instructions would require the bus in a given second for each CPU. We could therefore have as many as $400 / 20 = 20$ CPUs in the system.

To generalize this, let x be the cache hit rate, then every CPU requires $(1 - x)200$ million instructions to go over the bus in a given second, so the system could have as many as $400/(1 - x)200$ CPUs, that is $2/(1 - x)$. The last question is then answered by the equation $(1 - x)32 = 2$ or $x = 30/32 = 15/16$.

Q18

If we assume no copying we are only interested in the network transfer time:

$$64 \cdot 8 \cdot 10^{-9} \text{ s} = 512 \text{ usec.}$$

The time for 1 copy is: $(64 \cdot 8 \cdot 20)/32 \text{ nsec} = 320 \text{ nsec}$. Therefore the total delay not ignoring copying time is $512 + 3 \cdot 320 \text{ nsec} = 1472 \text{ nsec}$.

If instead 4 additional copies are required, then it takes $512 + 7 \cdot 320 \text{ nsec} = 2752 \text{ nsec}$.

Q26

One obvious reason for a limit on cable length is signal strength. But this is a minor reason as the technology would still allow for fairly long cables (longer than the current limits) so there must be another reason.

This has to do with collision detection. When two computers at either ends of a cable send data around the same time, there is a collision, i.e. the signal received on both sides is garbled. Both NIC cards would detect this and need to resend all the packets that collided. To determine how many packets need to be resent, one has to first determine how long it takes for a single packet to travel across the cable, in other words one needs to know the cable length. Therefore to make things simpler, the cable length is made short enough that every NIC knows at most one packet on each side could have been involved in the collision.

Q28

(a) VOD definitely requires a connection. Without a connection, VOD would require every video to be sent to all subscribers all the time and only allow those who pay for it to view it. This can be done for PPV but the add-on of VOD is that the subscriber can start viewing from any frame s/he chooses as soon as the purchase is completed, and this would require an enormous bandwidth to stream all frames of all videos at all time. With a connection oriented protocol the required frames would only be sent to the interested parties. Also note that it is not possible to broadcast over the internet.

Also an unreliable connection will be appropriate. A reliable connection would increase the delay between video frames, while the final quality of the video would not suffer much from a few frames being lost in the course of the transfer.

(b) This definitely requires a connection, and the page should not have missing data when displayed to the client. A reliable byte stream would be appropriate.