Test of Pearl 101 — Operating systems and computer networks

Pearls of Computer Science (201300070) / Introduction to BIT (201300073)
Bachelor module 1.1, Technical Computer Science, EWI

14 October 2016, 13:45–14:45

Module coordinator: Maurice van Keulen
Instructor: Pieter-Tjerk de Boer

Answers

---

1. Operating systems

7 pt (a)

D

9 pt (b) B (memory management) because this hardware feature allows implementation of e.g. virtual memory, i.e., letting the OS fetch data from hard disk only when the program actually tries to access that part of memory.
E (protection) because the OS can use this hardware feature to detect that a program is trying to access memory that it is not allowed to access, e.g., because it is allocated to another program.

B, E

7 pt (c) The process goes through the state diagram as follows:
New – Waiting – Executing – Blocked (until keypress) –
Waiting (after the keypress has happened) – Executing –
Waiting (after being interrupted by a timeout) – Executing –
Waiting (after being interrupted by a timeout) – Executing –
Waiting (after being interrupted by a timeout) – Executing –
Finished.

1

7 pt (d) See above; the "dispatch" is the transition from Waiting to Executing.

5

Continued on next page...

2. Networks – protocols

7 pt (a) **C**

7 pt (b) **B**

9 pt (c) **Packet 5, because it acknowledges data which has not yet been sent.**
(It acknowledges bytes up to (and including) nr. 2400, but they have only been sent up to nr. 2200, namely see packet 2.)

Alternative answer, worth **5** *points:* **packet 3, because it should have Seq=230**.
Indeed, there's a gap in the data being received from host 130.89.144.74 (the "server"), but that does not mean that this packet is really incorrect. It could well be that a packet containing the missing data bytes has been lost on its way from the server to us; see the next question.

The `Ack=2001` in packet 3 is also not really wrong. You could expect this packet to have `Ack=2201`, thus acknowledging the contents of packet 2, but it might well be that packet 3 was sent by the server *before* it received packet 2 from us: the packets may have crossed each other.

9 pt (d)

`source IP =` **130.89.144.74**

`destination IP =` **130.89.13.213**

`Seq =` **230**

`Ack =` **2001**

`Len =` **20**

Note that this packet fills the sequence number range between packets 1 and 3.

Alternative answer, worth **5** *points:*

`source IP =` **130.89.13.213**

`destination IP =` **130.89.144.74**

`Seq =` **2201**

`Ack =` **230**

`Len =` **200**

The idea behind this answer is that it explains why packet 5 acknowledges data up to sequence number 2400. However, this answer is not really correct because Wireshark is running on host 130.89.13.213, so packets *from* this host can't be lost in the trace.

3. Networks – delay

15 pt (a)

Transmission delay on link A–B:	Propagation delay on link A–B:
$\frac{4000\ \textbf{bits}}{4\cdot 10^6\ \textbf{bits/s}} = 0.001\ \textbf{s} = 1\ \textbf{ms}$	$\frac{100\ \textbf{km}}{2\cdot 10^5\ \textbf{km/s}} = 0.0005\ \textbf{s} = 0.5\ \textbf{ms}$
Transmission delay on link B–C:	Propagation delay on link B–C:
similar calculation: 2 ms	**negligible because of small geographical distance**
Transmission delay on link C–D:	Propagation delay on link C–D:
similar calculation: 4 ms	**negligible because of small geographical distance**

Quite a few students made arithmetic errors or got the units confused (mega/kilo/milli etc.). Only a few points were subtracted for such errors, but your future employer probably won't be so tolerant if your calculations are off by a factor of 1000!

7 pt (b) There's no queueing at node A, since packets are generated with more time between them than the time needed to send them.
Potentially, there could be queueing at both nodes B and C, since both have a faster incoming than outgoing link. However, since it is given that the packets are generated at 3 ms intervals, and node B needs only 2 ms to transmit one of them, no queue will build up there. The only queue will be at **C**, since there packets arrive at 3 ms intervals, while it takes 4 ms to transmit one of them.
If you answered "B and C", you still got 4 points.

C

16 pt (c) **First packet takes 1 + 0.5 + 2 + 4 = 7.5 ms (adding all transmission and propagation times) to arrive at D.**
The other 2 packets queue behind it at node C, so they arrive at D with 4 ms (= transmission delay on link C–D) intervals, completing the transfer in 7.5 + 2 × 4 = 15.5 ms.

The above is a very concise answer, building on the argument made at question b about where the queue is.

There are many other ways of answering this question, e.g. by making a table of where every packet is in 0.5 ms steps, drawing a timeline diagram showing what happens when to each packet, or writing down a more verbose text ("At t=3, the second packet is generated at A and its transmission can start immediately, finishing at t=4; after the propagation delay, its last bit arrives at B at t=4.5. It finds no queue at B, so ...." and so on).