



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Examination Answer Books

12 Pages

Instructions to Candidates

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
3. The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.

Complete, fold and seal
(see note 2 below)

SURNAME:

Thomson

FORENAME(S):

Nathan James

1	Student Number	P141062
2	Title of Examination Paper	Network Multiplayer Games
3	Module Code	M6201105
4	Date and Time of Examination	~~M6201105~~ 13th May 2019, 12 PM
5	Examination Venue	Swansea Campus, 310
6	Seat Number	G054
7	Number of Answer Books handed in	

List the numbers of the questions you have attempted	Office Use
1a	16
1b	
2a	19
2b	
3a	16
3b	
4a	19
4b	
4c	
4d	
4e	
Total	70%

GT 24/5/19

1a)

firstly the transport protocol will establish a connection. this is due to the protocol being connection-orientated. This connection will ensure that data will not be lost and will ensure that the server and client will be synchronized. Transport protocol utilizes a concept named three-way handshake. The host will send segment 'X' to the client, upon receiving segment 'X' the client will send an acknowledgement of 'X+1' back to the host and it's own segment off. This means that this is informing the host that it has correctly received seg bytes up to 'X' and is expecting to receive 'X+1' next.

These steps allow the protocol to order the bytes correctly if they arrived in the incorrect order as the protocol will know the correct order.

the protocol will also be able to recover data that is lost via transmission by re-transmitting ~~protocol~~ ~~segment~~ 'X' if it was not received correctly.

the protocol uses a concept named expectational acknowledgements in which the client and server ~~are~~ know the segments that are to be sent and received and this if an issue arises, can rectify it.

10

Difficult to read!

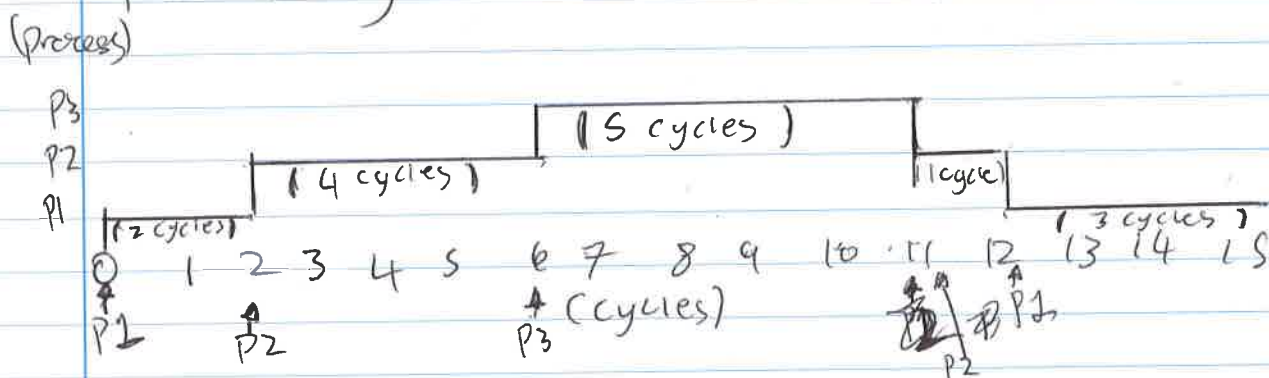
- 1b) a pre-emptive kernel is a kernel in which all threads can run concurrently. Non-pre-emptive kernels defer to threads running at different times, based on their priority.

the following

for example for processes P1, P2, P3 in which

process	arrival time (cycles)	execution cycles	priority
P1	0	5	3 (lowest)
P2	2	5	2
P3	6	5	1 (highest)

the resulting execution would look as follows



Non-pre-emptive does this to aim to ensure no deadlock, pre-emptive can encounter deadlock as multiple threads running could require resources held by another process that is running.

A non-pre-emptive kernel inherently has a priority system, as shown in the diagram above. Pre-emptive kernels could encounter deadlock, so a priority system would need to be implemented for a task queue system could be implemented to resolve this.

The pre-emptive kernel may also require revisiting the priority of low priority processes if deadlock occurs to a higher one in order to resolve deadlock.

Pre-emptive kernels have the benefit of being fast due to running concurrently but can encounter deadlock, non-pre-emptive kernels are slower due to their priority system, but lessen the risk of deadlock.

2a) a non-deterministic MAC has no guarantee of reaching the destination in a timely manner. This is due to collisions. The Carrier Sense Access Multiple Access - collision detector (~~CSMA-CD~~) will check to see if there is a collision going to be had when transmitting data, if there is the CSMA-CD will wait protocol will wait a random interval of time before attempting to re-transmit the data.

3

If an ~~online~~ online first person shooter game (FPS) this could have big draw backs. As FPS games are real-time any data not sent and received ~~it~~ would have an effect on gameplay. For example, player movement data could arrive late and this cause the player to jump back to a position prior when ~~it~~ eventually ~~correctly~~ successfully sends ~~and~~ the data, and the client successfully receives the data.

2

A solution could be to send data with time stamps and synchronize network nodes with the server using protocol data units (PDU).

1

address. Class C.

2B)

199.7.10.0

Number of subnets, $n = 9$ Maximum hosts, $Mh = 12$

Firstly finding the amount of bits that must be borrowed

$$\text{bits} = 2^n - 2 \geq n$$

try $2^1 - 2 \geq 9$, $2 - 2 \geq 9$. No
 $0 \geq 9$. No, good.

try $2^2 - 2 \geq 9$, $4 - 2 \geq 9$. ~~$2 \geq 9$~~
 $2 \geq 9$. No good.

try $2^3 - 2 \geq 9$, $8 - 2 \geq 9$.
 $6 \geq 9$. No good.

try $2^4 - 2 \geq 9$, $16 - 2 \geq 9$. ✓
 $14 \geq 9$. Good.

therefore 4 bits need to be borrowed for Subnetting.

as the ip address 199.7.10.0 is a Class C address, the resulting and we borrowed 4 bits for Subnetting. result mask is 255.255.255.240.

next if we can proceed must be determined, this is done by ensuring we have enough bits remaining

as the ip address is 199.7.10.0 meaning it is a Class C IP address and we have borrowed four bits for subnetting, the resulting Subnet mask is 255.255.255.240. ✓

the next step is to determine if we have enough bits remaining for the host addresses ~~ranges~~.

an important consideration is that in every host address range all '0's will be reserved for the network address and all '1's will be reserved for the broadcast domain.

therefore to calculate if we have enough bits is as followed.

$$h = 8 - \text{number of borrowed bits}$$

$$h = 8 - 4$$

$$h = 4$$

~~we have~~ F. there is four bits left for host addresses. next we must calculate how many hosts each subnet will be able to support this is done as followed.

$$1h = 2^h - 2 \geq 1h$$

$$1h = 2^4 - 2 \geq 12$$

$$1h = 16 - 2 \geq 12$$

$$1h = 14 \geq 12$$

we have enough bits. therefore we can continue.

knowing that the subnets cannot be all 0's the first three subnets are as follows.

$$0001 \quad 0000 = 199.7.10.16$$

$$0010 \quad 0000 = 199.7.10.32$$

$$0011 \quad 0000 = 199.7.10.48$$

Knowing that hosts cannot use the all 0's or all 1's the first three subnets, host ranges are as follows

Subnet one host address ranges

0001 0001 to 0001 1110
 199.7.10.17 to ~~199.7.10.30~~ 199.7.10.30

Subnet two host address ranges

~~to~~ 0010 0001 to 0010 1110
~~199.7.10.33~~ to
 199.7.10.33 ✓ to 199.7.10.47 ✓

Subnet three host address ranges

0011 0001 to 0011 1110
 199.7.10.49 ✓ to 199.7.10.63 x

the hosts

13

3 a) deadlock refers to when a program ceases to function ~~due~~ due to starvation, starvation refers to when processes cannot finish their tasks as it requires resources it does not have the possibility of dying.

deadlock can only occur if all of the four following conditions are present. These conditions are Mutual exclusion, hold and wait, non-preemptive and circular wait.

Mutual exclusion refers to when a resource can only be accessed by one process at any given time.

hold and wait refers to a ^{resource} process that is being held by a process, but this process requires an additional resource that is being held by another process in order to finish its task.

Non-preemptive refers to when a process can only give up its resources it is holding when it has completed its task.

~~Circular~~ Circular wait refers to when a chain of two or more processes require a process that is being held by the next member in the chain.

an example of deadlock is when two processes are holding a resource and each process requires the resource the other process is holding.

In order to fix this timeouts could be placed on

the Semaphores or a queue system
can be implemented.

14

3 b)

Semaphores and mutexes are variables
that are used in multi-threading.

~~Sem mutexes~~

Multi-threading refers to having multiple
processes running concurrently on
different threads.

Mutexes can only run one at a time, however
multiple semaphores can be used concurrently.

should

mutexes ~~eg~~ be used in applications where
it is paramount data is transmitted, for example.
turn based games such as chess, or any cost
in for mutexes will put the ~~next~~ player at a big disadvantage

Semaphores should be used in games in which speed
performance is most important, real-time games such
as pong, a semaphore can be used for each
board and the ball to ensure no object is
waiting for another, each semaphore (or
process) ~~has~~ movement of its objects.

4a) Consistency refers to ~~head~~ every node having the same data, this requires tight coupling and low latency in order to maintain Synchronization

3

Responsiveness refers to offloading work to client machines in order to free-up server load, this requires loose coupling and Protocol data units to be sent via the server to ensure clients do not diverge too far away from the correct data.

1.5

4b) For app in order to reduce the amount of ~~data that~~ game data that needs to be sent the ~~game~~ game could use a few approaches.

Some approaches could be to only send world data for a section of the world the player is in and not the entire world, to only send user data and to use compression techniques such as Huffman technique.

2

4c) this would be dead-reckoning.

Dead reckoning is a methodology in which the nodes on the network and not the server would calculate algorithms. this would be performed client side and ~~this~~ as a result, free of server load.

Some algorithms it can be used for one ~~calatua~~ calculating acceleration and velocity, if velocity is known client-side, acceleration can be updated client side.

The client may also create entities of objects, for example a vehicle and store the objects

kinematic information. This kinematic information could include velocity, acceleration and orientation, damage and also store any dead-reckoning algorithms.

~~The client would monitor its own kinematic information comparing it with the server for the entity.~~

~~upon passing a pre-set agreed threshold of divergence,~~

there will be a pre-set acceptable threshold of divergence, referring to how far the amount in which the client's entities are incorrect by in their kinematic information when comparing it to the server values for the same properties.

upon surpassing the acceptable threshold of divergence, the server would send time stamped ~~and~~ protocol data units updates containing the correct information to every node on the network in order to synchronize the network.

5.5

4D. Interpolation is when the current point and the end point are known at all times and it steps incrementally across the path from start ~~point~~ end point. Gives a smooth ^{look} ~~result~~ as a result.

Extrapolation is when it does not know the entire information and estimates the next point by using the acceleration, velocity ~~vectors~~. Can cause incorrect values and as a result Jerry motions from when the server sends out ~~corrected~~ protocol data units.

4

4e) This is due to extrapolation. In order to resolve this, the game could use interpolation in conjunction with dead-reckoning.

Upon sending protocol data unit updates from the server to correct and synchronize entities on clients with the correct values, instead of instantly jumping to the correct position, the ~~differe~~ entity could interpolate from the ~~show~~ incorrect ~~to~~ values to the correct values, as this would make it look like a smoother transition.

Furthermore, the threshold of divergence from entities and server could be lowered in order to allow for corrected values and synchronizations more frequently, allowing for less severe jumps in position. This however is a trade-off as more updates means higher server load.

3.