



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.

1	Student Number	P149706
2	Title of Examination Paper	network multiplayer games
3	Module Code	M6X01105
4	Date and Time of Examination	13/05/2019 12.00
5	Examination Venue	310
6	Seat Number	G055
7	Number of Answer Books handed in	

Complete, fold and seal
(see note 2 below)

SURNAME: Thomas

FORENAME(S): George Anthony

List the numbers of the questions you have attempted	Office Use
1.a	5
2.a	23
2.b	
3.a	10
3.b	
4.a	15
4.b	
4.c	
4.d	
4.e	
Total	53

2) a) CSMA-CD is a layer two protocol. When data is about to be sent, it will check the network to see if there will be a collision. If there is no collision then the data will be sent. If there is a collision then the transfer will be cancelled early. It will then wait a random amount of time before trying again. This will have issues when being used in real time multiplayer games because the delay in send and receive will lead to a disadvantage. Non-deterministic MAC allow for data transfers to be completed without any errors however no data to be lost however it takes longer for all data to be sent.

b) 199.7.10.0 N=9 subnets - Class C
Mh=12 hosts

Find the amount of bits needed to borrow using:

$$2^n - 2 \geq N \quad N = 9$$

$$n=1 \quad 2^1 - 2 = 0 \geq 9 \quad \text{no good}$$

$$n=2 \quad 2^2 - 2 = 2 \geq 9 \quad \text{no good}$$

$$n=3 \quad 2^3 - 2 = 6 \geq 9 \quad \text{no good}$$

$$n=4 \quad 2^4 - 2 = 14 \geq 9 \quad \text{OK}$$

n=4 4 bits needed from host
 18693216

Subnet mask = 1111 0000

= 240

Subnet mask = 199.7.10.240

is subnetting possible

host bits
remaining = h

$$2^h - 2 \geq Nh$$

$$2^4 - 2 \geq 12$$

$$16 - 2 \geq 12$$

$$\underline{14 \geq 12}$$

Subnetting is
possible

$$h = 8 - n$$

$$h = 8 - 4$$

$$\underline{n = 4}$$

host starts with 8
bits because class C

Subnets

$$1 = 00010000 = 16 = 199.7.10.16$$

$$2 = 00100000 = 32 = 199.7.10.32$$

$$3 = 00110000 = 48 = 199.7.10.48$$

$$4 = 00100000 = 64 = 199.7.10.64$$

Subnet range >

$$1 = 0001 - 0001 = 16 + 1 = 17$$

$$0001 - 1110 = 16 + 14 = 30$$

Subnet 1 ranges from 199.7.10.17
to 199.7.10.30

$$2 = 0010 - 0001 = 32 + 1 = 33$$

$$0010 - 1110 = 32 + 14 = 46$$

Subnet 2 ranges from 199.7.10.33
to 199.7.10.46

$$3 = 0011 - 0001 = 48 + 1 = 49$$

$$0011 - 1110 = 48 + 14 = 62$$

Subnet 3 ranges from 199.7.10.49
to 199.7.10.62

Subnet 4 - $0100-0001 = 64+1 = 65$

$0100-0001 = 64+14 = 78$

Subnet 4 ranges from 199.7.10.65
to 199.7.10.78

Subnets of all 0's and all 1's are ignored because all 0's are reserved for network and all 1's for domain.

3/a) Deadlock is when all processes are waiting for a resource which is being held by another process. This causes the system everything to be stuck waiting around and everything stops. Deadlock is caused by 4 conditions being true simultaneously. Firstly a chain link which is where 1 process is waiting for a resource which is being held by next process which is waiting for a resource from the next process and so on. The next condition is mutual exclusion.

Deadlock can be prevented by giving each process a timer so that it can only use a resource for a certain amount of time. Or processes can be ranked on importance, so that higher priority processes can use a resource rather than waiting for it from a lower priority process to finish.

3b) Mutexes and Semaphores ~~are~~ both ^{can} used when multithreading a system. They will allow for different threads to run simultaneously so that for example a game's world could move and update on one thread while another waits for user input. ~~This however~~ for this situation a mutex ~~is~~ would be used. This is because there are only two processes that need to run simultaneously. ~~Herein~~ This is the main difference between a mutex and a semaphore. Because there ~~can~~ ^{can} be ~~one~~ ^{multiple} mutex but ~~can~~ there can be multiple semaphores. However, semaphores are only needed if you want to run more than 2 threads simultaneously. Semaphores can be used when updating multiple users/information simultaneously. ✓

4)a) Consistency ensures that all nodes are up to date on all information so that there is no disadvantage to any user. Responsiveness is how quickly a node updates, ensuring synchronization. ✓

If all nodes are not up to date then this can cause synchronisation issues. ✓

4b) There are multiple approaches that can be taken to minimize data that needs to be sent. For instance, sending the entire world data is way too much data and is unnecessary. Sending only important data like player inputs will reduce data to be sent. Also making the clients & system do work for the system will reduce data such as is done with techniques like dead reckoning.

2

4c) Dead reckoning is a technique used to reduce improve responsiveness. There are many types of dead reckoning algorithms, such as not moving, constant velocity, acceleration and so on. ~~It sends PDU's which contain kin.~~ The player's PC will create an object for player's entity in the world. It will then send PDU's which contain kinematic information about the entity. Such as position, velocity, acceleration, orientation, damage and dead reckoning algorithm type. The algorithm creates a predicted path that the entity will take and will send PDU's every 1 in 5 seconds. Of course, players do not move in the way that is predicted. So the PC will monitor the player's movement and if the player's movement veered from the predicted path by an accepted amount then the PC will send out a PDU with the updated info and will restart dead reckoning. This can result in jerky movement.

6

4d) Interpolation is used to find a smooth transition between points. If you had two positions that were far away then the movement between them would be unpleasant for users. Interpolation fixes this. Extrapolation is predicting movement ~~that goes past~~ the next position or game state by using the current position or game state. If an object has a velocity then extrapolation can be done to find the next predicted position. If acceleration is known then velocity can be updated.

4e) Interpolation and dead reckoning can be used together. Because dead reckoning has a predicted path and when user strays from the result is jerky, interpolation can be used to remove the jerkiness from the movement.

1 a) ^{Assuming they mean they will use features of TCP} TCP requires a connection to be established in order to send data. A 3 way handshake can be used. This will send sequence number to identify which data is expected. Host 1 will send an initial message identifier of x . If received, host 2 will $+1$ to the identifier to say that it has received everything up until x and expects $x+1$ next. Host 2 will send $x+1$ back to host 1 along with its own identifier of y . This ensures that data is correctly ordered when being sent. Compression can prevent data loss.