COSC 264

Assignment

User Codes:

Source Code :

Morgan:

1.

A deadlock occurs when two processes sharing a singular resource. When the use of this conflicts such that no process can access the resource it becomes a deadlock.  
The deadlock in our networking program occurs when…..

2.

3.  
To solve the problem with bit errors we added a header checksum to the packet type. This checksum is simply an addition of all other header fields. This meant that in the sender and receiver we had to add a check that the packet checksum was indeed the same as the sum of all other header fields for each packet. If not the packet must have had a bit error introduced and would be dropped.

4.  
The select function allows a program to wait on file descriptors or similar (such as sockets in our case) until they become ready. That is, when it becomes possible to perform I/O operations without blocking

5.  
To check that the file was copied exactly we looked at the properties window in the file view for the outfile and made sure it had the exact number of bytes as the infile. To complement this, we also had “START” and “END” in the files respectively and checked to make sure these were copied where they should be.  
Could we make the file more diverse and then have another program to check it??

6.

7.

Imas:

Questions:

1. The protocol between sender and receiver as described above has (at least) one

weakness: it has a deadlock. Please explain the notion of a deadlock in the

context of networking protocols and describe the particular deadlock situation in

our case. A guiding question is: what can go wrong and when in case certain

packets are lost?

Ans) A deadlock refers to the problem the occurs when two programs are using the same resource and therefore, prevent each other from using the resource as both are contesting it. In our case, the sender is sending programs from the same port that is is being read from. I our case, a deadlock that can occur is if enough packets are lost in quick succession such that both sender and receiver are waiting on packets and therefore do not send any packets themselves. This will cause both to time-out, resulting in both sending packets at similar times and therefore not sending the packet in the end. This problem can be resolved by staggering the delay between the sender and the receiver. However, this will not remove the chance for a deadlock to occur, but rather allow it be self-resolve provided that the packet isn’t completely dropped within this time frame.

What is the magicno field good for?

The magicno field is a random hexadecimal number that gives an easy way to ensure that the packet is what is being expected. Therefore, it is also useful in the checksum as it is a value that is expected to be constant throughout all packets. ??? – Not too sure but idk what else it is

Please explain what the select() function is doing and why it is useful for the

channel (and in another way for the sender).

The select function (select.select) in python mimics the select function in linux in that it allows for easy waiting of inputs and outputs. This is very useful for the channel as without it it becomes difficult to properly buffer any inputs and send them to the receiver. For sender this is useful as it makes sure that the packets will arrive in the same order sent assuming that they are buffered properly by the channel. This waiting on inputs is what makes the channel function properly as otherwise it would be difficult to guarantee in sequence delivery of the packets from sender

5. Please explain how you have checked whether or not the file was transferred cor-

rectly (i.e. the receivers copy is identical to the transmitters copy).

Contribution Form: