POKHARA UNIVERSITY

Time Bound Open Book Hybrid Examination

Level: Bachelor	Semester: Spring, 2020	Full Marks: 70
Programme: BE Software		Pass Marks: 31.5
Course: Network Programming		Time : 2 hrs.

Candidates are required to give their answers in their own words as far as practicable. The figures in the margin indicate full marks.
Attempt all the questions.

Group – A: (5×10)

1. Explain the need of different protocols to communicate between different processes in the same or different computers and their other features beside error and flow control. 10

2. What are the uses of generic socket address structure in network programming? How it can be implemented in different socket API call? Under what conditions does a descriptor become ready to (i) read and (ii) write? 2.5+2.5+ 2.5+ 2.5

OR

In an Internet Datagram client, you don't need to "bind" the socket to the client's address. But in a Unix datagram client, you need to bind the socket to the client's address. Why? Explain with suitable example code and show the mechanism of passing length of socket address structure in different functions. 5+5

3. Give an outline of simple TCP server that can handle multiple clients using Berkeley Socket. Briefly explain in what condition the zombie is created and how they can be handle? 6+2+2

4. In which approach of communication, windows socket provides better functionalities than Unix socket? Discuss about synchronous and asynchronous communications in network programming. On the basis of your assumption, list out the nature of applications which should use the synchronous socket and asynchronous socket. 10

5. Why is it necessary to call WSAStartup() and WSACleanup() functions during development of winsock client-server applications? Can you make a half-duplex from a full duplex socket? If yes, explain how? If no, explain why not. 10

Group – B: (1 × 20)

6. Assume that you are appointed as team lead in a small software company. There are 9 software programmers working under you. You need to communicate with your subordinates frequently. Also, all of your subordinates are connected to the LAN network. In order to communicate with your subordinates, you do not want a third party application or any fancy graphics application (since developing the required application by yourself leads to quick promotion). Also you cannot install any new software without concerning the system department. You also don't want to reveal your communication to other colleagues except your team members. 20

   Design and implement a simple CUI based chat/messaging application using appropriate socket/Winsock APIs to communicate with your subordinates. Your application should fulfill the given requirements.

1. Your goal is to create cross-platform Chat Server (i.e. it may run either in Unix/Linux or windows environment)
2. Your subordinates always initiate connections with unique code names. (use nickname).
3. Team leader keeps record of all of his/her subordinates in buffer.
4. Team leader should store subordinates member id(you can also use socket descriptor for id as well) , ip address, nick name , connection request time, message to team leader, message to other member (should keep member id and message)- append some digit at the start of message to distinguished the type of message (i.e. message to team leader or message to other member)
5. If member 1 wants to send a message to member 2 , it should pass the message via team leader. (if he/she only wants to communicate with team leader- message to other member field should be null or empty string and id value is negative number)
6. If 50% or more than 50% members want to quit the application, your team's leader application and all remaining members' applications also quit the application. (you can use voting to quit application)
7. Send a bye message to all of your subordinates during exiting the application.

Develop necessary protocols required to complete the application.

Mark Distribution for Application

Evaluation: 60%- for designing cross-platform system, protocol, communication mechanism, buffer handling and data structures.

40%- for implementation (Code part) – minor syntactical mistakes will be ignored. Concern more about function, flow, user defined functions etc.