SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)
(Deemed to be University, u/s 3 of UGC Act 1956)

Networks Laboratory Lesson Plan — Autumn' 2023 (5th Semester)

Discipline: CSE

Course name and Code: Networks Laboratory (IT 3095) (L-T-P-Cr:0-0-2-1)

Instructor Name: Dr. Manas Ranjan Lenka, Email: manaslenkafcs@kiit.ac.in

Instructor Chamber: F-101, 1st Floor, Block-C, Campus-14

Free Time : Tuesday (2.00 – 3.00PM), Thursday (2.00 – 3.00PM), Friday (2.00 – 3.00PM)

Mobile Number : 91-9861077824

Technical Assistants Details:

1. Mr. Prasanta Kumar Panda, Email: prasant.panda@kiit.ac.in, Mob: 9861275798
2. Mrs. Sasmita Pradhan, Email: 2281076@kiit.ac.in, Mob: 9438413245

Course Contents:

List of Experiments (Day wise):

Day1

- *Aim of the experiment:*

1. Discuss what is networking and it's significance in computer network. Discuss the components (i.e. h/w and s/w) required for data communication in a Computer Network. (Show the h/w components like Network Interface Card (NIC), Network Cable, RG-45 Connector,Hub, Switch, Router etc.)

2. Highlight the importance of socket programming as a s/w for data communication and the basic fundamentals required for doing socket programming using C.
3. Review of function, pointer, structure, structure with in a structure, pointer to structure, and command line argument concept using C programming Language.
4. What is little endian and big endian. Discuss the significance of endianness in computer network.

Assignments

1. Write a C program to swap the content of 2 variables entered through the command line using function and pointer.

   Input: ./swap 10 20
   Output: After swapping value of num1 = 20, value of num2 = 10

2. Write a C program to assign values to each members of the following structure. Pass the populated structure to a function Using call-by-value and another function using call-by-address and print the value of each member of the structure. Then print the value of each member of the structure.

```
struct student_info{
        int roll_no;
        char name[50];
        float CGPA;
        struct dob age;
};
struct dob{
        int day;
        int month;
        int year;
};
```

   Input: Populate each data member of the structure
   Output: Print the value of each data member of the structure

3. Write a C program to extract each byte from a given number and store them in separate character variables and print the content of those variables.

Input/Output:

```
manas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_By_
e/Lab1$ ./a.out 258
The input number = 258
digit in the 1st byte=2
digit in the 2nd byte=1
digit in the 3rd byte=0
digit in the 4th byte=0
```

4. Write a C Program to enter a number and store the number across the following structure and print the content of each member of the structure. Then aggregate each member of the structure to form the original number and print the same.

```
struct pkt{
        char ch1;
        char ch2[2];
        char ch3;
};
```

Input/Output:

```
manas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_By_Me/Lab1$ ./a.out 258
The input number = 258
digit in the 1st byte=2
digit in the 2nd byte=1
digit in the 3rd byte=0
digit in the 4th byte=0
1st member of the structure=2
2nd member of the structure=1,0
3rd member of the structure=0
The regenerated number = 258
manas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_By_Me/Lab1$
```

5. Write a C program to check whether the Host machine is in Little Endian or Big Endian. Enter a number, print the content of each byte location and Convert the Endianness of the same i.e. Little to Big Endian and vice-versa.

Input/Output:

```
manas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_By_Me/Lab1$ ./a.out 258

extracted byte from the LSB of the Number=2

extracted byte from the LSB of the Number=1

Memory representation of the Number

---------------------------
Memory Address  ->      Value
---------------------------
1668590496      ->      2
1668590497      ->      1
1668590498      ->      0
1668590499      ->      0

The LSB of the number is stored at the lowest memory address
Hence, the host machine is in little Endian

The Number is converted to Big Endian now
Memory representation of the Number

---------------------------
Memory Address  ->      Value
---------------------------
1668590444      ->      0
1668590445      ->      0
1668590446      ->      1
1668590447      ->      2

The number in Big Endian Format is 33619968
manas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_Bymanas@manas-HP-ProBook-x360-440-
G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_Bymanas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/
manas@manas-HP-ProBook-x360-440-G1:~/Manas_Data/Study_Materials/Important_Materials/Computer_Networks/Network_lab/Lab_By_Me/Lab1$
```

Day2

➢ *Aim of the experiment:*

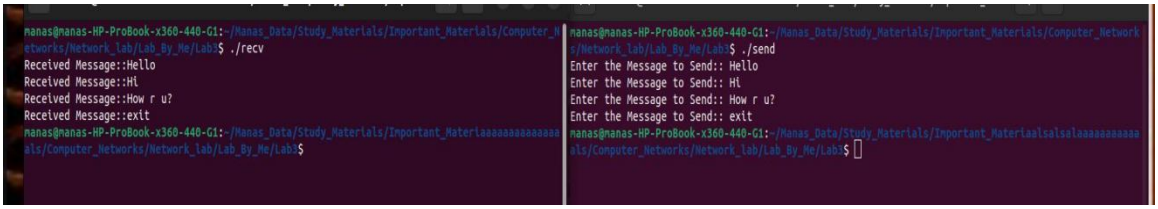
1. Basics of Socket Programming.
2. Details of Connection less Socket programming APIs for TCP/IP stack using C.

Assignment

1. Write a sender and receiver program in C by passing the IP address and the port number of each other through the command line arguments using connection less

socket. Both of them will exchange messages with each other continuously. If any one of them will receive the "exit" message from the other end then both of them will close the connection. (Assume both the client and server are running with in the same host)

Input/Output:



Day3

- *Aim of the experiment:*

1. Demonstrate the packet Analyzer tool (wireshark) to analyze the details of a packet which is captured during packet transmission in the network.

Assignments

1. Analyze the packets using wireshark, that are captured by running both client and server (connection less) with in the same host.
2. Analyze the packets using wireshark, that are captured by running the client in one host and the server in another host (connection less).

Day4

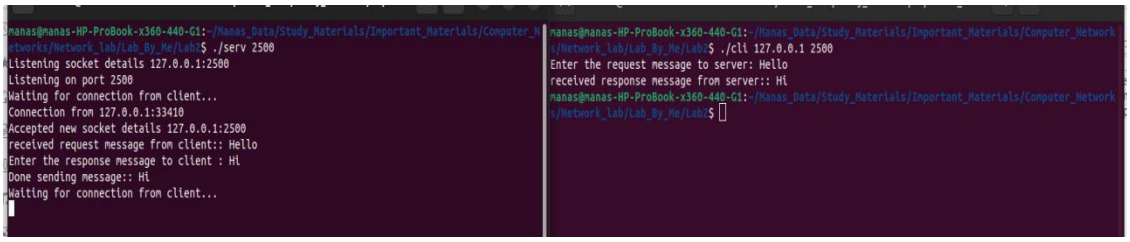
- *Aim of the experiment:*

1. Details of Connection Oriented Socket programming APIs for TCP/IP stack using C.

Assignment

1. Write a connection oriented client and server program in C using command line arguments. At the server side, pass the port number (to whom the server will bind to) in the command line. At the client side, pass the IP address and the port number of the server (to whom the client will connect to) as command line argument and carry out the following tasks.
   i. Print all the relevant messages during the connection establishment at both the ends.
   ii. After establishment of connection exchange messages. After message exchange is over then the client sends a "close" message to the server to tear down the connection.

Input/Output:



Day5

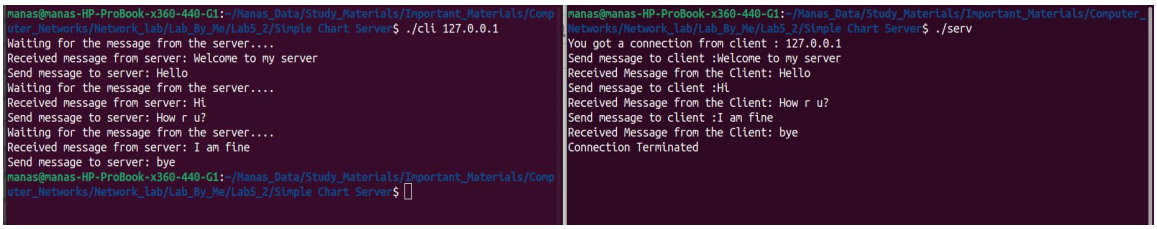
➢ *Aim of the experiment:*

1. Discuss how to design a Sequential Chart Server.

Assignment

1. Write a connection oriented client and server socket program using C where the server will behave as a chart server serving multiple chart clients but one at a time. When the chart server receives a "bye" message from a particular client then it terminate the respective connection with that client.

Input/Output:



Day6

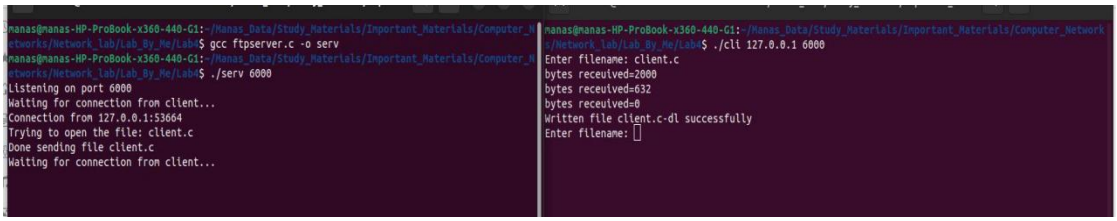
➢ *Aim of the experiment:*

2. Discuss the overview of file transfer over a computer network.

Assignment

1. Write a connection oriented client and server program in C using command line arguments. Do the file transfer from the server as follows.
   i. Server first sends the list of files present in the current directory at it's own end.
   ii. After receiving the same, client send the name of a file it wants to download from the server.
   iii. Finally, after receiving the same server uploads the file to the client.
   iv. After sending the file, server closes the client connection at its own end.

Input/Output:



Day7

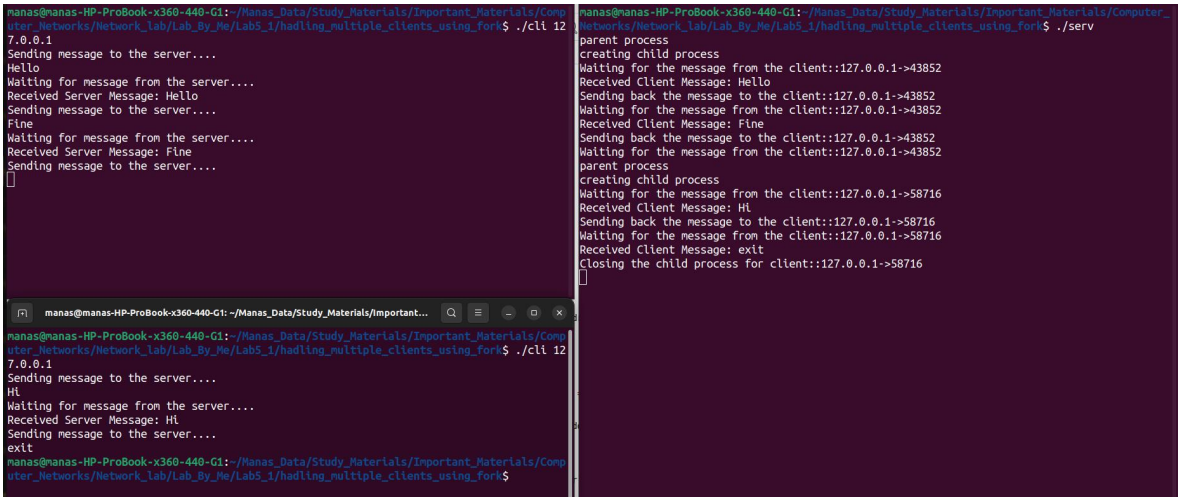
➢ *Aim of the experiment:*

3. What is I/O multiplexing and why it is required?
4. Discuss different types of I/O multiplexing.
5. Discuss how to design a concurrent chart server using fork().

Assignment

1. Design a connection oriented concurrent chart server using fork() in C where the server will serve multiple chart clients simultaneously. When the chart server receives a “exit” message from a particular client then it terminate the respective connection with that chart client.

Input/Output:



Day8

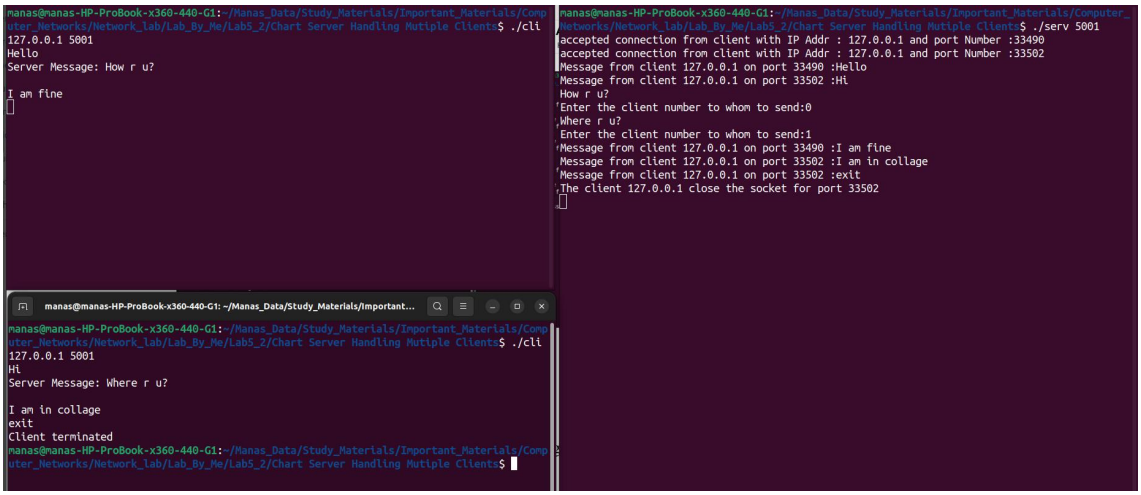
➢ *Aim of the experiment:*

1. Details of I/O multiplexing using select() API.
2. Discuss how to design a concurrent chart server using select().

Assignment

1. Design a connection oriented concurrent chart server using select() in C where the server will serve multiple chart clients simultaneously. When the chart server receives a “exit” message from a particular client then it terminate the respective connection with that chart client.

Input/Output:



Day9

➢ *Aim of the experiment:*

1. Introduction to network simulator tool (NS2/NS3/Packet Tracer) and its application in research work.
2. Demonstration of how routing works using NS2/NS3/Packet Tracer.

Assignments

1. Simulate routing of packets in a LAN with in the same subnet.
2. Simulate routing of packets in a LAN with different subnets.

Day10

➢ *Aim of the experiment:*

1. Comparison and analysis of some routing protocols using NS2/NS3/Packet Tracer.

Assignment

1. Compare and analyze DSR and AODV protocol using NS2/NS3/Packet Tracer.

Grading Policies:

- *Continuous Evaluation components :* Continuous evaluation 60 Marks (30 Marks evaluation before the mid-sem examination and the rest 30 Marks evaluation after the mid-sem examination) consists of the following components.
  - √ **Lab participation (10 Marks)**: Students' participation in the lab based on their attendance and engagement.
  - √ **Lab records (10 Marks):** Neatly written lab records based on the assignments to be evaluated.
  - √ **Continuous evaluation (based on Lab skills, 20 Marks)**: Students' lab skills will be assessed through hands-on activities and involvements in doing assignments during the lab hour.
  - √ **Quizzes/Viva-voice (20 Marks)**

- *End semester evaluation :* The sessional examination will be of 40 marks, consisting of the following two components.
  - √ Programming Skill test (30 Marks)
  - √ Quiz Test (10 Marks)

    OR
  - √ Research projects on Networking (40 Marks)

Practice Problem Sets:- Project on "**LAB Monitoring System**"

1. Design a client-server architecture, where the server will monitor all possible activities of each student in a LAB as follows.

   - First the instructor will open his/her laptop/desktop to start the server application. Then each student will open their laptop/desktop to start a client application that connects to the server running at instructor's laptop/desktop.

   - Each student can chat with the instructor as and when needed and vice-versa. But no two students can chat with each other.

   - The instructor can take the snapshot of a particular student's laptop/desktop as and when he needed.

   - A suitable pop-up message will be shown in the instructor's laptop/desktop in any of the following cases.

     i. In case a student's laptop/desktop remain inactive for a certain period (as configured by the instructor).

     ii. In case a student opens any browser for surfing the Internet.

iii. In case a student inserts a pen drive into his/her laptop/desktop.

iv. In case a student close the client application running in his/her laptop/desktop.

Reference Materials:-

1. Beej's Guide to Network Programming Using Internet Sockets (https://beej.us/guide/bgnet/pdf/bgnet_a4_c_2.pdf)
2. Unix Network Programming, Volume 1: The Sockets Networking API (Addison-Wesley Professional Computing Series) by W. Stevens, Bill Fenner, and Andrew Rudoff,
3. Wireshark Tutorial for Beginners - YouTube (https://www.youtube.com/watch?v=TkCSr30UojM)
4. Cisco Packet Tracer (https://www.netacad.com/courses/packet-tracer)