**School of Computer Engineering**

**KIIT deemed to be University**

**Networks Laboratory Lesson Plan – Autumn’2023 (5th Semester)**

Discipline: CSE/IT/CSCE/CSSE

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

*Instructor Name:* Dr. Suneeta Mohanty, [Email: smohantyfcs@kiit.ac.in](mailto:Email:%20smohantyfcs@kiit.ac.in)

*Instructor Chamber:* F-08, Block-A, Campus-15

Technical Assistants Names: Manorama Choudhury, Email:manorama.choudhury@kiit.ac.in,

Sandipta Mangaraj, Email:sandipta.mangaraj@kiit.ac.in

**Course Contents:**

**List of Experiments (Day wise):**

**Day-1**

* *Aim of the experiment:*

1. Discuss what is networking and its 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.
2. Write a C program to assign values to each member 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.

struct student\_info{

int roll\_no;

char name[50];

float CGPA;

struct dob age;

};

1. 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:*

1. 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.

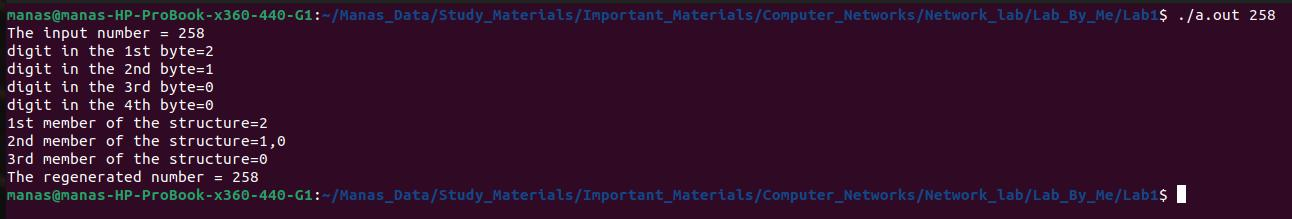
struct pkt{

char ch1;

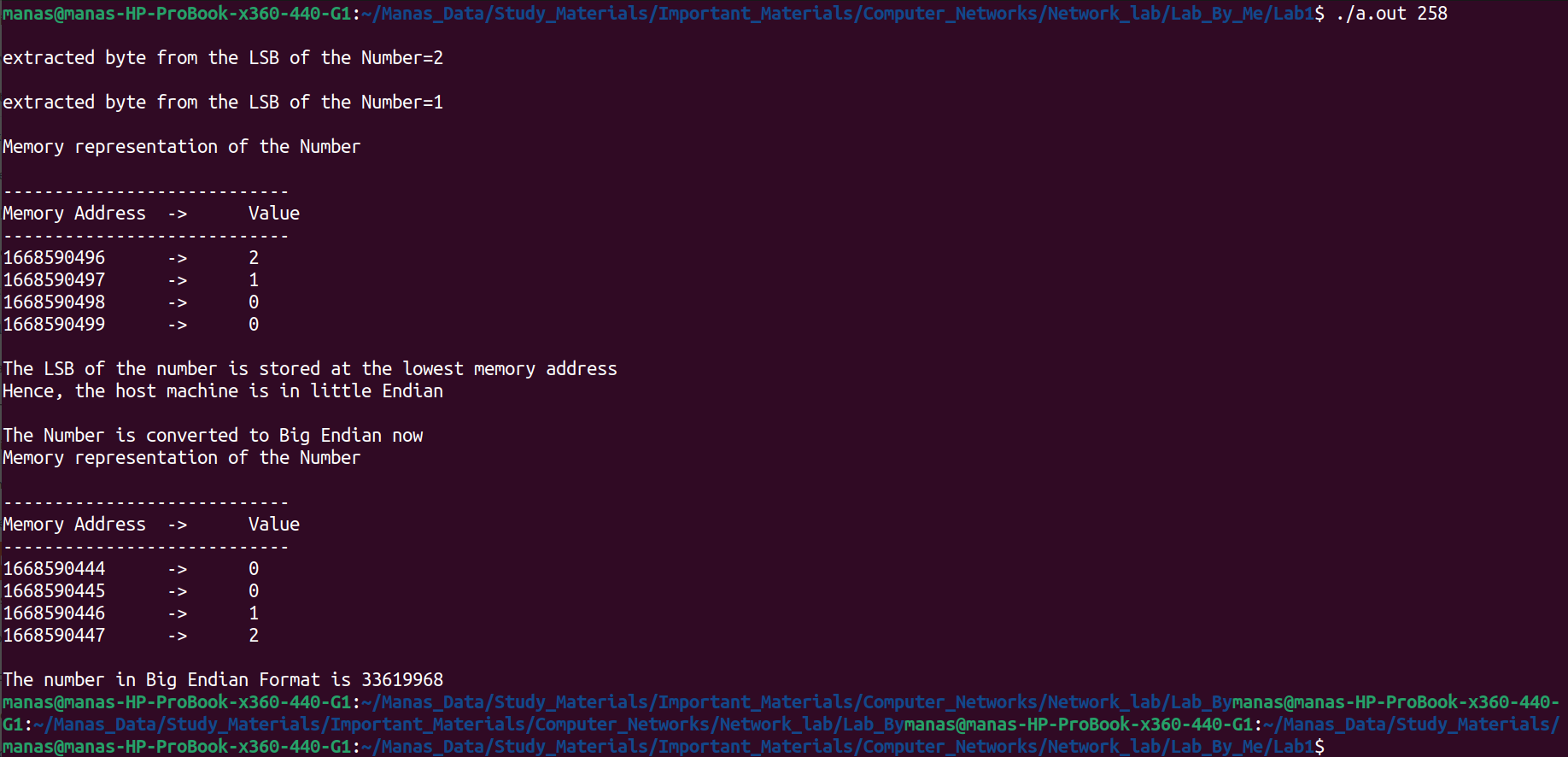
char ch2[2];

char ch3;

};

*Input/Output:*

1. 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:*

**Day-2**

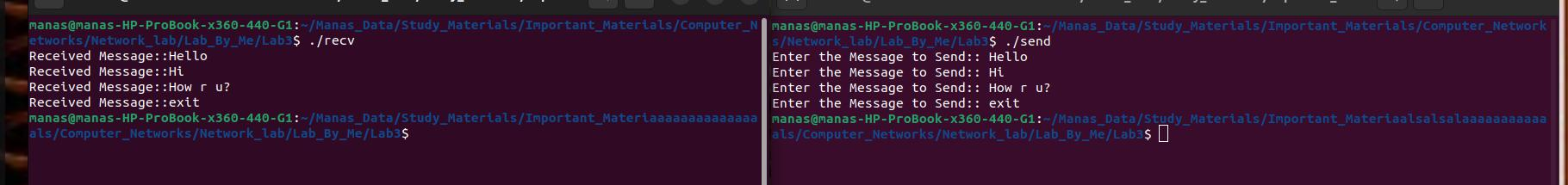
* *Aim of the experiment:*

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:*



**Day-3**

* *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).

**Day-4**

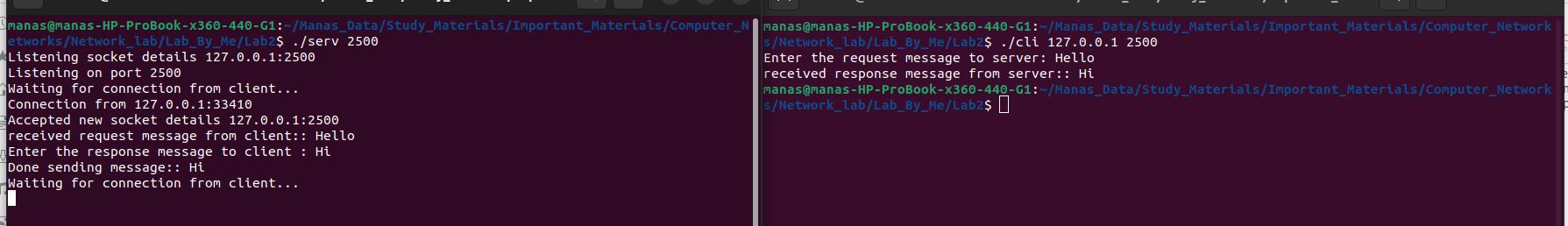
* *Aim of the experiment:*

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.
2. Print all the relevant messages during the connection establishment at both the ends.
3. 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:*



**Day-5**

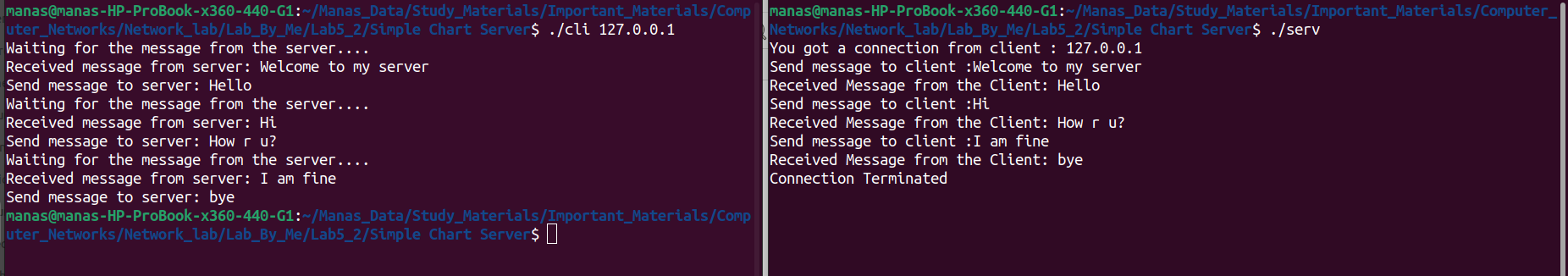
* *Aim of the experiment:*

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 terminates the respective connection with that client.

*Input/Output:*



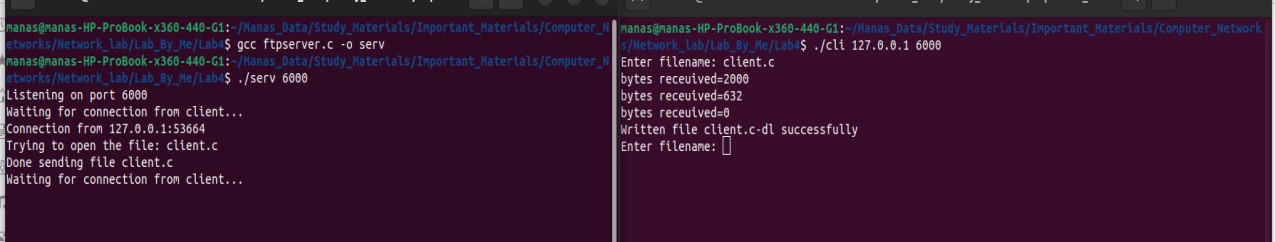
**Day-6**

* *Aim of the experiment:*

1. 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.
2. Server first sends the list of files present in the current directory at it's own end.
3. After receiving the same, client send the name of a file it wants to download from the server.
4. Finally, after receiving the same server uploads the file to the client.
5. After sending the file, server closes the client connection at its own end.

*Input/Output:*

**Day-7**

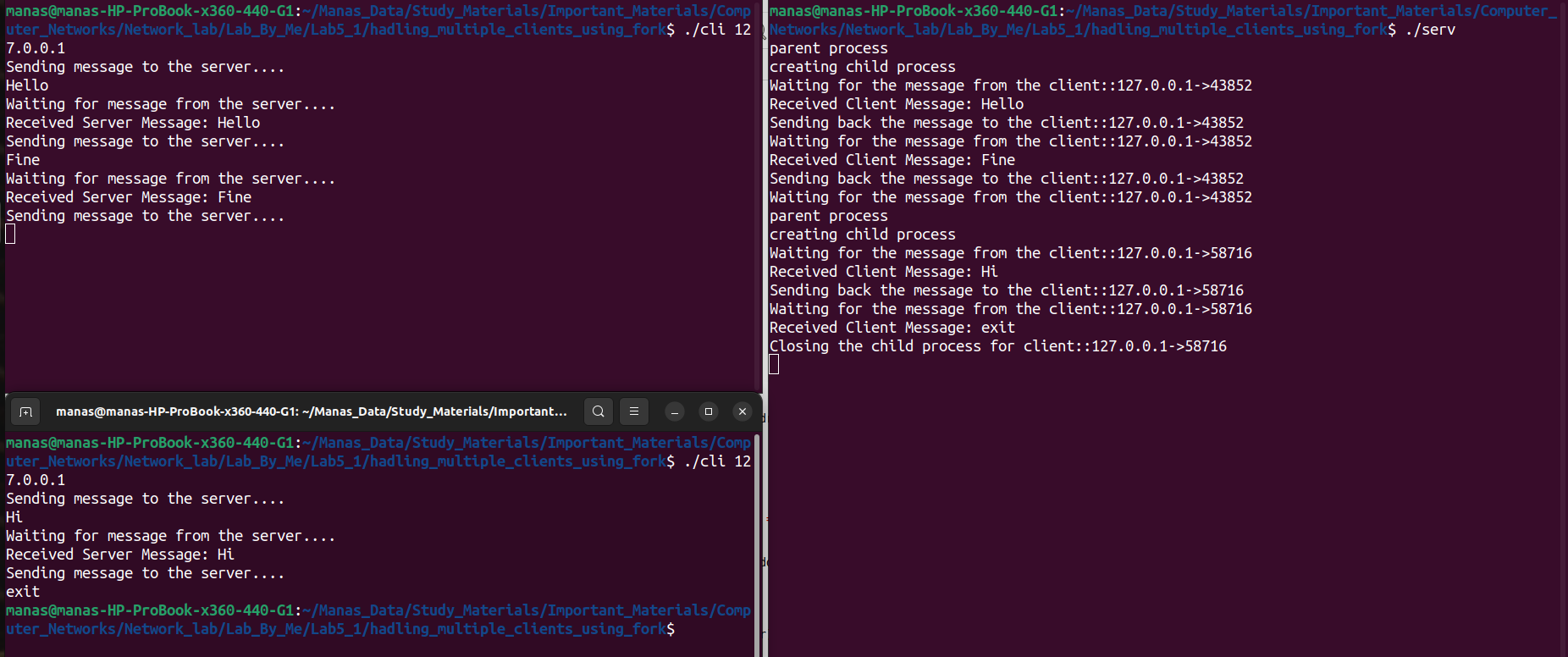
* *Aim of the experiment:*

1. What is I/O multiplexing and why it is required?
2. Discuss different types of I/O multiplexing.
3. 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:*



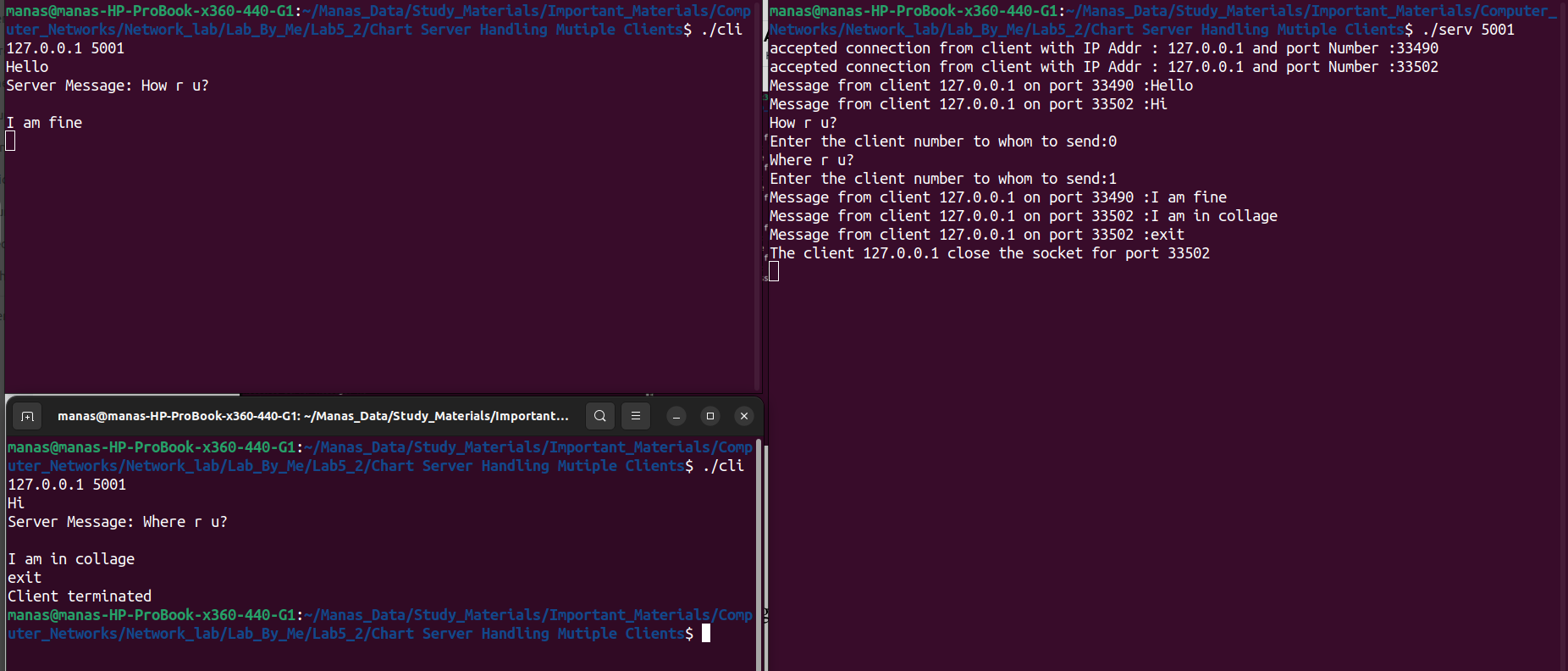
**Day-8**

* *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:*

**Day-9**

* *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.

**Day-10**

* *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 for 60 marks Consists 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.
* **Use a variety of evaluation methods to get a comprehensive assessment of student learning. (20 Marks)**
* End semester evaluation: Comprehensive *assessment of student learning and performance. (40 Marks)*

**Practice Problem Sets**: - Problems on socket configuration and communication on different application protocols will be given.

**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](https://www.amazon.in/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=W.+Stevens&search-alias=stripbooks), [Bill Fenner](https://www.amazon.in/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=Bill+Fenner&search-alias=stripbooks), and [Andrew Rudoff](https://www.amazon.in/s/ref=dp_byline_sr_book_3?ie=UTF8&field-author=Andrew+Rudoff&search-alias=stripbooks),
3. [Wireshark Tutorial for Beginners - YouTube](file:///C:\Users\KIIT01\Downloads\Wireshark%20Tutorial%20for%20Beginners%20-%20YouTube)([(https://www.youtube.com/watch?v=TkCSr30UojM)](https://www.youtube.com/watch?v=TkCSr30UojM)
4. Cisco Packet Tracer(https://www.netacad.com/courses/packet-tracer)