MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

COURSE PLAN

Department   : COMPUTER APPLICATIONS

Course Name & code   : Network Lab & MCA 5163

Semester & branch   : 3rd Semester, M.C. A

Name of the faculty   : VINAYAK M. NIRMAL KUMAR NIGAM ARCHANA

No. of contact hours/week   : 3

ASSESSMENT PLAN:

1. **In Semester Assessments   - 60 %**
   - Lab test   -   Four tests each of 10 marks each (Max. Marks: 40)

   - Execution & Record Check   -   Two evaluation of 10 marks each (Max. Marks: 10) that includes
        - ✓ Observation (5 marks)
        - ✓ Execution (5 marks)

2. **End Semester Examination - 40 %**

   - Lab examination of 3 hours duration (Max. Marks: 40)

The programs in the lab should be developed as per the client server model as shown in the fig 1 for illustration of the various methods of IPC.
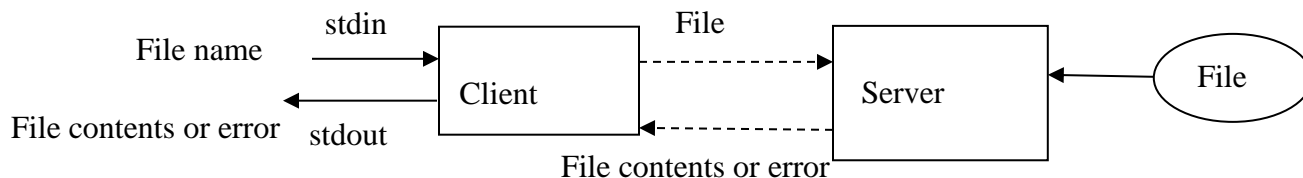

Fig 1 Client- Server model

*(Page **1** of **9**)*

The client reads a file name from the standard input and writes it to the IPC channel. The server reads this file name from the IPC channel and tries to open the file for reading. If server can open the file, it responds by reading the file and writing it to the IPC channel, otherwise server responds with an error message. The client then reads from the IPC channel, writing what it receives to the standard output. The two dashed lines in Fig 1 between the client and server, correspond to some form of IPC.

INSTRUCTIONS TO STUDENTS
1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring their observation book without fail.
4. They should implement the program individually.
5. While implementing the programs students should see that their programs would meet the following criteria:
   - Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
   - Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs and outputs
   - Statements within the program should be properly indented
   - Use meaningful names for variables and functions.
   - Make use of Constants and type definitions wherever needed.
6. Once the programs get executed, they should show the program and results to the instructors and copy the same in their observation book.
7. Program template that should be followed while writing the program in the observation book is
   - Program Title
   - System Calls used
   - Program Code
   - Result

8. Questions for lab tests and exam need not necessarily be limited to the questions in the manual, but could involve some variations and / or combinations of the questions.

Course Outcomes (COs)

At the end of this course, the student should be able to:

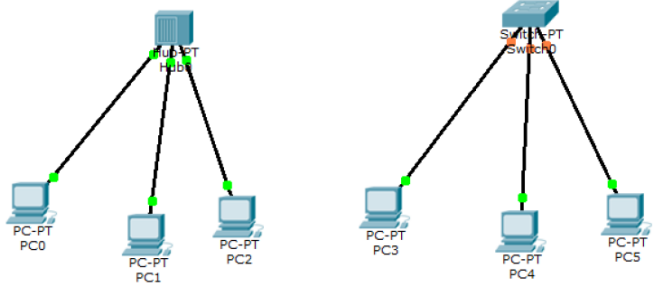|		No. of Contact Hours	Marks	PO
C01:	Implement Inter-process Communication between Processes	15	45	
C02:	Implement Socket Programming using C & Unix.	9	27	
C03:	Construct network with connecting devices- switch, hub & routers and configuring.	6	19	
C04:	Construct networks and simulate application protocols- DHCP, HTTP & FTP	3	9	
	Total	**33**	**100**	

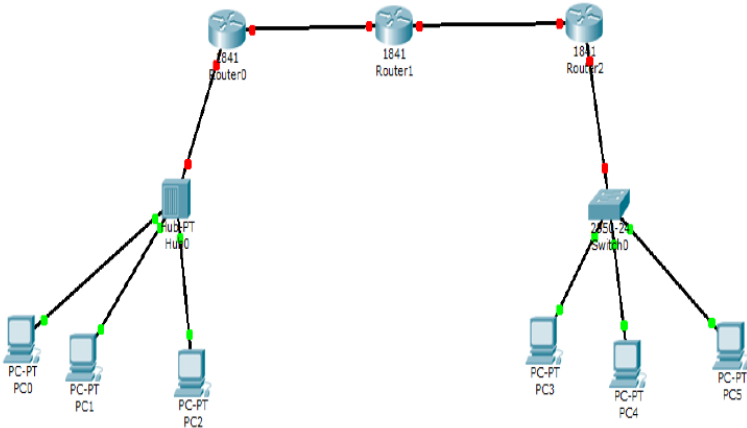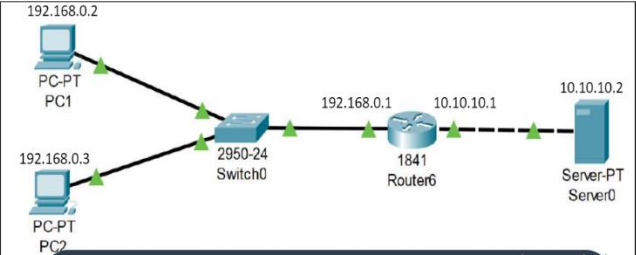## Course Plan

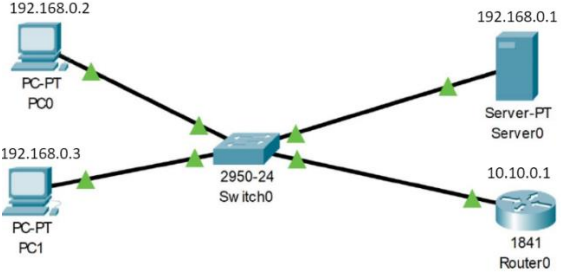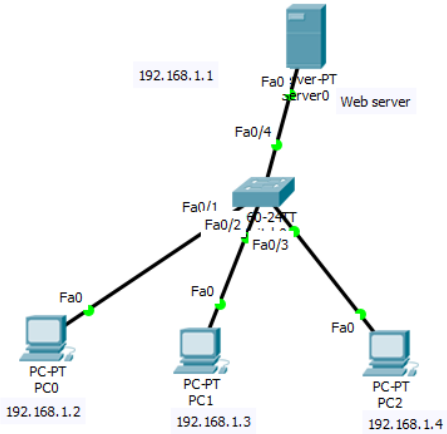| L. No.	Programs
L1	1. Write a program to copy one file to another using file system calls. 2. Write a program to demonstrate the concept of parent & child processes using fork () system call in which the parent reads a file name from the keyboard and child process uses the file name and reads first 10 characters from the file. 3. Write a program to accept an array before forking and sort the array in child and display. 4. Write a program to accept an array before forking and search the array in child for an element accepted and display the result. 5. Write a program to write some characters to a file and close it in the parent. Child should open the same file in child and display. 6. Accept two numbers and operator (+, -, *, /) as command line arguments. Perform numerical operation as per operator in the child and display the result. 7. Write a program to accept an integer before forking and child has to generate multiplication table corresponding to the integer.
L2	**Pipes** 1. Write a program using pipe to send a string from parent to child in order to print number of characters and digits in the string.

L. No.	Programs
	2. Write a program using pipe to accept integer at parent and pass it to child. Child should check that number is Armstrong number or not and accordingly send a message - YES or NO to parent. 3. Write a program using pipe to accept an array of integers at parent and pass it to child. Child will pick only even numbers and return array of even numbers to parent. 4. Write a program using pipe to accept details of a student such as - RegNo, Mark1, Mark2, Mark3 and send it to child. Child will find average and accordingly assign Grades (A+, A, B, C, D, E, F- assume some criteria) and return grade to Parent. 5. Write a program using pipe to accept a matrix on n X n and pass it to child. Child will multiply each element of user's choice to each element return new matrix back to parent. 6. Write a program using pipe to accept a string and pass it to child. Child will check the string whether it is palindrome or not. If palindrome then return message PALLINDROME otherwise NOT PALLINDROME to Parent.
L3	**FIFO** 1. Write client-server program using named pipes. Client accepts a string and passes to Server through a named pipe and server checks the string for palindrome and returns a message - YES or NO to Client. 2. Write client-server program using named pipes. Client accepts Username, password from user and passes them to server for authentication. Server has pre-stored username and passwords and it compares username and passwords and returns message -Welcome <username> or Login Access denied. 3.Write client-server program using named pipes. Client accepts Deposit amount and tenure (in years) and passes to server. Server has pre-stored information about different interest rates for different tenures. Accordingly, server applies appropriate interest rate and returns maturity amount and interest earned back to client. 0-12 months 5% 13-24 months 6% 25-36 months 7% >36 months 8% 4.Write client-server program using named pipes. Client accepts - electricity

MIT/GEN/F-1a/R0

L. No.	Programs
	customer number and number of units consumed.
	Server calculates the bill by applying different rate for different slabs of units consumed.
	0- 50 units @ 1/- per unit.
	51-100 units @1.5/- per unit.
	101-200 units @ 2/- per unit.
	>200 units @ 3/- per unit
	5. Write client-server program using named pipes. Client accepts - passenger name, source, destination and number of seats and sends to server. Assume that server has pre-stored information about - Source, Destination, ticket_rate and number of seats available. On receival of information from client, server checks for availability of seats. If available, accordingly decrement seats and sends bill amount to client, otherwise display message to client that -Seats not available
L4	**Message Queue**
	1.Write a client server program using message queue to sort array of elements. Client takes input from user a set of integers and send to server using message queue for sorting. The server reads message queue and return sorted array to client for displaying at client.
	2.Write a client server program using message queue to simulate calculator operations (+, -, *, / between two numbers). Client accepts two numbers and operation to be performed from the user and writes to message queue. Server reads from message queue and performs the required operation to produce result. The server returns the result to client using message queue in order to display result.
	3.Write a client server program using message queue to cypher the text. Client take input a line of text from the user and sends to server using message Queue. Server reads text from the message queue and cypher the text (cypher method- simple one any of your choice) and return cyphered text to Client for display.
	4. Write a client server program using message queue to book multiplex tickets. Assume that there are 5 ticket categories and each category there are 20 tickets. Assume that this is pre-stored information and available at server. From client program, User inputs- Name, Phone no, Ticket category and Number of tickets and pass it to server for ticket reservation. Depending on users input, decrement the number of seats in corresponding category, send booking information to client for displaying on the screen.

MIT/GEN/F-1a/R0

L. No.	Programs
L5	**Socket Programming-TCP** 1. Write a client-server chat socket program using TCP protocol to chat in infinite loop until client ends the chat session. 2. Write a client-server socket program using TCP. Client accepts an IP address and sends to server. Server receives IP address and displays it. Server process the IP address to identify the class of IP address (A, B, C, D, E). Server sends class and default mask to the client via socket. Client displays the result. 3. Write a client-server socket program using TCP. Assume that Server has pre-stored information about five Account number, PIN and Balance. Client accepts account number, pin and withdrawal amount and sends to Server If account holder is valid then performs withdrawal operations provided enough balance is available otherwise display invalid Accno/PIN or No enough Balance. Assume that account can maintain 1000 balance. 4. Write a client-server socket program using TCP. Client accepts and sends coefficients (a, b, c) of a quadratic equation to server. Server calculates roots of quadratic equation and return roots to client. Client displays the roots of quadratic equation. Consider all possibilities of a,b,c values.
L6	**Socket Programming-UDP** 1. Write a client-server chat socket program using UDP protocol to chat in infinite loop until client ends the chat session. 2. Write a client-server socket program using UDP. Client accepts an IP address and sends to server. Server receives IP address and displays it. Server process the IP address to identify the class of IP address (A, B, C, D, E). Server sends class and default mask to the client via socket. Client displays the result. 3. Write a client-server socket program using UDP. Assume that Server has pre-stored information about five Account number, PIN and Balance. Client accepts account number, pin and withdrawal amount and sends to Server If account holder is valid then performs withdrawal operations provided enough balance is available otherwise display invalid Accno/PIN or No enough Balance. Assume that account can maintain 1000 balance. 4. Write a client-server socket program using UDP. Client accepts and sends name of a file to server. Server reads 10 characters at a time from the file and sends to client. Client displays the data received and writes to another file.

MIT/GEN/F-1a/R0

L. No.	Programs
L7	**Socket Programming- Multi Client** 1. Write a multi client-server chat socket program using TCP protocol to chat in infinite loop until client ends the chat session. 2. Write a multi client-server socket program using TCP. Client accepts an IP address and sends to server. Server receives IP address and displays it. Server process the IP address to identify the class of IP address (A, B, C, D, E). Server sends class and default mask to the client via socket. Client displays the result. 3. Write a multi client-server socket program using TCP. Assume that Server has pre-stored information about five Account number, PIN and Balance. Client accepts account number, pin and withdrawal amount and sends to Server If account holder is valid then performs withdrawal operations provided enough balance is available otherwise display invalid Accno/PIN or No enough Balance. Assume that account can maintain 1000 balance. 4. Write a multi client-server socket program using TCP. Client accepts and sends coefficients (a, b, c) of a quadratic equation to server. Server calculates roots of quadratic equation and return roots to client. Client displays the roots of quadratic equation. Consider all possibilities of a, b, c values.
L8	Construct a 3 or more-node topology by connecting a hub and switch and realize the working of hub & switch. Consider IP address for each system as follows. | Node ID	IP address
PC0	192.168.2.2
PC1	192.168.2.3
PC2	192.168.2.4
PC3	192.168.2.5
PC4	192.168.2.6
PC5	192.168.2.7 | Create the above two networks and connect them with HUB or SWITCH Perform following experiments and observe, what type of packets get transferred? Whether packets are unicasted or broadcasted? • Set ICMP packet transfer/Ping between PC0 and PC2 and record the result.

MIT/GEN/F-1a/R0

L. No.	Programs
	• Set ICMP packet transfer /Ping between PC3 and PC5 and record the results. What is the difference between hub and switch?
L9	Implement different network design topologies - Bus, Star, Ring and transfer the data packet from one PC to another PC. Choose proper IP addresses of any appropriate class. Ping from one device to another and observe the traffic.
L10	Construct & connect two or more networks by configuring router, nodes with RIP protocol. Simulate the communication within and between networks. Create the above two networks and connect them using Router. Configure the network and routers with Valid IP and RIP protocol. Run PING and observe the results- packet exchange and Routing tables created.
L11	Construct simple networks to simulate the application protocols- HTTP, FTP and DHCP. Configure the server for FTP application Protocol and observe the result- uploading a text file from one client to Server and downloading file from server via another client. Configure the server for DHCP application Protocol and observe the result- assignment of IP's to DHCP client devices

(*Page 8 of 9*)

L. No.	Programs
	  Configure the server for HTTP application Protocol and observe result-accessing web pages from browsers in clients
L12	**End Semester Exam**

References

1. W. Richard Stevens, "UNIX Network Programming Inter-Process Communications", Volume 2, Second Edition, Pearson Education, 2001.

2. A Rama Satish, "UNIX Programming", Scitech Publications, 2009

3. Douglas E Comer, David L Stevens," Internetworking with TCP/IP-Volume III", Pearson Education, Second Edition,2004

4. Jesin A, Packet Tracer Network Simulator (1e), Packt Publishing, 2014.

MIT/GEN/F-1a/R0