



COURSE PLAN

Department	: COMPUTER APPLICATIONS
Course Name & code	: Network Lab & MCA 5163
Semester & branch	: 3 rd Semester, M.C. A
Name of the faculty	: VINAYAK M.
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.

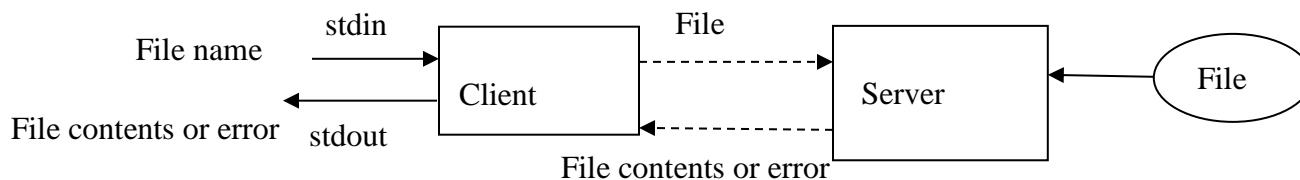


Fig 1 Client- Server model

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	Program Outcomes (POs) addressed
C01:	Understand Linux Commands and System Calls	3	
C02:	Implement Inter-process Communication between Processes	12	
C03:	Implement Socket Programming using C and Java	15	
C04:	Design application layer protocols using sockets	3	

Course Plan

L. No.	Programs
L1	<ol style="list-style-type: none"> 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	<p>Pipes</p> <ol style="list-style-type: none"> 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. 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.

L. No.	Programs
	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
L3	<p>FIFO</p> <p>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.</p> <p>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.</p> <p>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.</p> <p style="padding-left: 40px;">0-12 months 5%</p> <p style="padding-left: 40px;">13-24 months 6%</p> <p style="padding-left: 40px;">25-36 months 7%</p> <p style="padding-left: 40px;">>36 months 8%</p> <p>4.Write client-server program using named pipes. Client accepts - electricity customer number and number of units consumed. Server calculates the bill by applying different rate for different slabs of units consumed.</p> <p style="padding-left: 40px;">0- 50 units @ 1/- per unit.</p> <p style="padding-left: 40px;">51-100 units @ 1.5/- per unit.</p> <p style="padding-left: 40px;">101-200 units @ 2/- per unit.</p> <p style="padding-left: 40px;">>200 units @ 3/- per unit</p> <p>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.</p>

L. No.	Programs
	On receipt 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	<p>Message Queue</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
L5	<p>Shared Memory</p> <p>1. Write a client server program using shared memory to transmit data between server and client. Client accepts a Matrix and transmits to Server. Server displays received Matrix and finds transpose of the matrix. Transposed matrix is sent back to client and client displays the received transposed matrix.</p> <p>2. Write a client server program using shared memory to transmit data between server and client. Client accepts an integer and transmits to Server. Server displays received integer and check the number for Armstrong number. If the number is Armstrong number, server sends YES to client otherwise NO. Client displays the received message.</p> <p>3. Write a client server program using shared memory to transmit data between server and client. Client accepts State/Capital City and transmits to Server. Server displays received data (as state/capital city) and finds the capital or state depending what data received. Assume that Server has 2-D database which stores at least for 5 states and corresponding capital city. If corresponding city/state is found return</p>

L. No.	Programs
	the found information or NO MATCH FOUND to the client. Client displays the received information.
L6	Socket Programming-TCP <ol style="list-style-type: none"> 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. 5. Write a client-server socket program using TCP. Client accepts a classless IP address and passes to Server. Server displays received information and finds the mask and network address. Server returns mask and network address to client. Client displays the received information.
L7	Socket Programming-UDP Implement the programs given in L6 using UDP.

L. No.	Programs
L8	Socket Programming- Multi Client
L9	Java Socket Programming-TCP
L10	Java Socket Programming- UDP
L11	Java Network Application
L12	End Semester Exam

References

(Page 7 of 8)

- | |
|--|
| 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. Elliotte Rusty Harold, “Java Network Programming-Developing Networked Applications”, O’Reilly, 4th Edition, 2014 |