



### COURSE PLAN

<b>Department</b>	<b>: COMPUTER APPLICATIONS</b>
<b>Course Name &amp; code</b>	<b>: Network Programming Lab &amp; MCA 4205</b>
<b>Semester &amp; branch</b>	<b>: 3<sup>rd</sup> Semester, M.C.A</b>
<b>Name of the faculty</b>	<b>: Nirmal Kumar Nigam &amp; Vinayak Mantoor</b>
<b>No. of contact hours/week</b>	<b>: 3 + 1 tutorial Session</b>

### **ASSESSMENT PLAN:**

- 1. In Semester Assessments - 60 %**
  - Lab test
    - two tests of 10 marks each (Max. Mark: 20)
  - Regular Evaluation
    - Two evaluation of 10 marks each (Max. Marks: 20) that includes
      - ✓ Observation (5 marks)
      - ✓ Execution (5 marks)
- Viva(10 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.

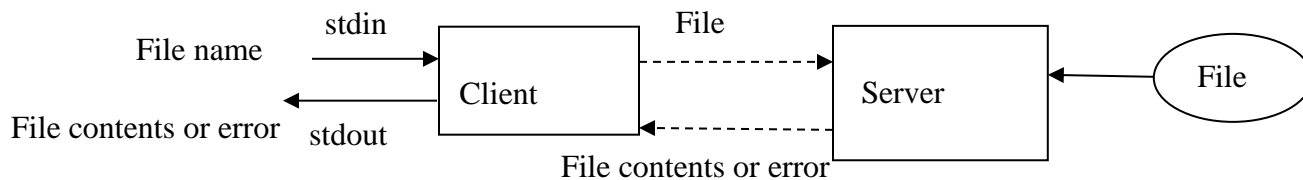


Fig 1 Client- Server model

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
<b>C01:</b>	Understand Linux Commands and System Calls	3	
<b>C02:</b>	Implement Inter-process Communication between Processes	6	
<b>C03:</b>	Implement Socket Programming using C	6	

## Course Plan

L. No.	Topics	Course Outcome Addressed
<b>L1</b>	<b>Basics of Linux System calls and file handling.</b> 1) Write a program to copy one file to another using file system calls. 2) Write a program using files to read the contents of an input file given through command line arguments. 3) Write a program using files to print the number of bytes specified of an input file. The inputs are given through command line arguments. 4) Write a program to accept two numbers before forking and find sum of numbers in the child and display. 5) Write a program to accept an array before forking and search an element in the array in child to display.	<b>CO1</b>
<b>L2</b>	<b>Pipes</b> 1) Write a Client Server Program using pipes. (as in figure 1) 2) Write a program to accept an array of integers in the parent process and pass them to child through pipe and child accepts an element to search from the array and sends the search result to the parent process. (A 2- way communication using multiple pipes concept).  <b>FIFO</b> 3) Using named pipe write a program to accept an array of integers in the parent process and pass them to child through pipe and child sends the sum of the elements to the parent process.	<b>CO2</b>
<b>L3</b>	<b>Message Queues</b> 1) Write a client server program using message queue where a client input a word and the server send the message whether the word is palindrome or not to the client.	<b>CO2</b>

L. No.	Topics	Course Outcome Addressed
	2) Implement the client-server calculator program using message queue. Client sends two numbers and operator to the server returns answer to the client to display.	
L4	<b>Client/Server programming using TCP Sockets</b> 1) Write a program to develop a Client and server application where the server reverses the sentence sent by the client. 2) Write a client server program in which server calculates the Net-salary of an Employee based on the details sent by the client.	CO3
L5	<b>UDP based Client /Server programs.</b> 1) Write a program to develop Client and server application in which the server checks whether the word sent by the client is palindrome or not. 2) Write a client server program in which server calculates the simple interest based on the details sent by the client.	CO3
L6	<b>End Semester Exam</b>	

### TUTORIAL

T.No	Topics
T1	Introduction on Linux System Calls-Example programs
T2	IPC – Pipes, FIFO
T3	Message Queue
T4	Socket Programming- UDP
T5	TCP

### References

W. Richard Stevens, “UNIX Network Programming Interprocess Communications”, Volume 2, Second Edition, Pearson Education, 2001.
A Rama Satish, “UNIX Programming”, Scitech Publications, 2009