

NETWORK PROGRAMMING

1. Introduction

Objective: Sockets provide the application layer level interface for two systems to communicate over Network. The primary objective of the course is to learn basics and advanced techniques of socket based client server programming. Berkeley Socket API based network programming using C language in Unix environment is taught in the course.

Credits: 3-0-0

2. Prerequisite

Unix System Programming (Covered as part of OS Course), C Programming

3. Course Outline

UNIT – I (Ten Hours):

To review Unix System Programming and OS concepts relevant for Network Programming

Topics:

- Process Control (fork, vfork, wait, exec system calls, user ids and related system calls),
- Reliable Signal Implementation (signal, sigaction, sigprocmask, sigsuspend, sigpending system calls, handling SIGCHLD)
- Pthreads

UNIT – II (Two Hours):

Overview of TCP/IP Protocol Suite and Internet Applications

Topics:

- Overview of TCP/IP Protocols
 - TCP (State Transitions)
 - UDP
 - IP
- Popular Internet Applications and their protocol usage

UNIT – III (Three Hours):

For introducing fundamentals of sockets and common functions used in any Client/Server application.

Topics:

- Socket
 - Introduction, role as an application programming interface
 - Address Structure and address management functions for IPv4, IPv6
- Common functions
 - Byte Ordering, byte manipulation functions
 - readn, writen, readln functions

UNIT – IV (Five Hours):

Basics of TCP Client/Server application development

Topics:

- Client/Server Paradigm
- Iterative TCP Server
 - Socket, Bind, Listen, Accept, read, write, close system calls
- TCP Client
 - Socket, Bind, Connect, read, write, close system calls
- Concurrent TCP Server
 - Using fork
 - Using pthreads
- Example of TCP Client/Server Application

UNIT - V (Three Hours) :

Basics of UDP Client/Server application development

Topics:

- UDP Server and UDP Client

- recvfrom, sendto system calls
- connect system call in UDP Client and asynchronous errors

UNIT - VI (8 Hours) :

Miscellaneous topics for building more sophisticated client/server applications

Topics:

- Data representation issues
- I/O Multiplexing
 - select or poll system call
 - Design of TCP, UDP Servers using select or poll system call
- Socket Options
 - getsockopt, setsockopt system calls
 - Few examples of SOCKET, TCP, UDP, IP options and their role in client/server applications
 - Using fcntl
- DNS related functions

UNIT - VII (4 Hours):

Understanding the mechanisms for logging messages, making a server as a daemon server and developing server programs to be used by inetd super server.

Topics:

- syslogd server and syslog system call
- Daemon server using daemon_init function
- inetd super server

UNIT - VIII (4 Hours):

Debugging client server applications and understanding an implementation of application protocol

Topics:

- Tools for debugging TCP, UDP applications
 - tcpdump, tcpflow, netstat, ethereal
- Detailed analysis of implementation of an application layer protocol's client and server programs (like HTTP Server, HTTP Client)

UNIT - IX :

Advanced Network Programming topics (optional)

Topics;

- Broadcasting
- Multicasting
- Routing Sockets
- Raw Sockets
- Data link access and design of packet filters.

4. Reading Material

Text Books

1. Richard Stevens: "Advanced Programming in Unix Environment", Pearson Education Asia. (For review topics)
2. Richard Stevens: "Unix Network Programming Volume I (Networking APIs: Sockets and XTI)", Pearson Education Asia.

Reference Books

1. Douglas E. Comer and David L. Stevens: "Internetworking with TCP/IP Volume III: Client-Server Programming and Applications, Linux/POSIX Sockets Version", Prentice Hall.