

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Convolution Neural Networks, Recurrent nets, autoencoders

Unix Network Programming (BHCS18C) Discipline Specific Elective - (DSE)

Credit: 06

Course Objective

This course introduces the concepts of Internet protocols, ports used during communication, Client/Server concepts and various transport protocols used in computer network applications and services. The objective is to equip the students with technical knowledge of it comprises of the study of the sockets used with TCP and UDP include IPV4 & IPV6.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Describe and analyse the various Internet Transport layer protocols used in TCP/IP AND UDP.
2. Comprehend the concepts and structures of both TCP based connection-oriented and UDP based connection-less client server applications.
3. Write various real-life client-server applications using socket programming.
4. Modify, maintain and extend the present internet client-server applications and write any new type of internet applications to suit the current needs of Internet users.

Detailed Syllabus

Unit 1

Introduction: Basics of Client Server applications, Example of day time client server, concurrent servers, protocols, sockets, port numbers.

Unit 2

Connection-oriented and Connection-less client server Applications: Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, close function, Socket Address Structures, Byte Ordering and Manipulation Functions, TCP Client and Server for Echo, Signal Handling in case of crashing and rebooting of server, Shutdown process function

Unit 3

Socket Options: Getsockopt and stockpot functions, Socket states, Generic socket option

Unit 4

Connection-oriented and connection-less Sockets: TCP-oriented basic concurrent client server applications, UDP oriented Echo client and server application, Handling of errors like lost datagram, Lack of flow control with UDP, determining outgoing interface with UDP.

Unit 5

Elementary name and Address conversions: Domain Name System, socket functions like gethostbyname, gethostbyname2, gethostbyaddr function, uname function, gethostname function, getservbyname and getservbyport functions.

Unit 6

Advanced Sockets: Daemon Processes, Multithreaded server, Raw sockets.

Practical

1. Implement TCP Echo client and TCP Echo server (Iterative).
2. Implement TCP Echo client and TCP Echo server (Concurrent).
3. Implement TCP daytime client and TCP daytime server (Iterative).
4. Implement TCP daytime client and TCP daytime server (concurrent).
5. Implement UDP Echo Client and UDP Echo Server.
6. Implement UDP daytime Client and UDP daytime server.
7. Implement TCP client and server (concurrent) where client gets input from the user and sends it to server. Server displays it on the screen. Server then gets another input from the user and sends it to client. Client displays it on the screen. The process continues till server or client sends “bye” to the other party.
8. Implement TCP client and server (concurrent) where client requests server to transfer a file. Assume file is smaller than 1K size. If the file is present on the server, it is sent to the client otherwise an error message is sent to client. Client copies the file on the hard disk and disconnects.

9. Implement UDP client and UDP server where server displays the IP address and port number of the client sending the datagram. Client sends a datagram (size 64 bytes) three times to the same server. Server sends the message back to client. Client reports the time elapsed in sending and receiving of the message. Use connected UDP sockets.
10. Write to program to
 1. display name of the host
 2. all IP addresses of the host.
 3. Check whether FTP and HTTP services are running on the system.
 4. Display the name of the service running on port number specified by user.

References

1. Stevens, R. W., Fenner, B., & Rudoff, A. M. (2010). *Unix Network Programming: The Sockets Networking API*. 3rd edition. PHI.

Additional Resources:

1. Forouzan, B. A. (2017). *Data Communication and Networking*. 4th edition. McGraw-Hill Education.
2. Stevens, R. W. (2009). *Unix Network Programming*. 1st edition. PHI.
3. Tanenbaum, A. S. (2012). *Computer Networks*. 5th edition. Pearson Education

Course Teaching Learning Process

- Use of ICT tools in conjunction with traditional class-room teaching methods
- Interactive sessions
- Class discussions

Tentative weekly teaching plan is as follows:

Week	Contents
1	Introduction, client server applications, protocols, port numbers
2-3	Sockets Functions, fork and exec function, Socket address structure
4	TCP Echo Server

5	Signal Handling
6-7	I/O Multiplexing
8-9	Socket Options, Getsockopt and stockpot functions, socket states, generic socket options
10	Elementary UDP sockets, TCP and UDP oriented client server applications
11	Elementary name and Address conversions, DNS, socket functions
12	Daemon Processes
13-14	Multithreaded server
15	Raw sockets

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Connection oriented sockets, connection less sockets, advanced sockets

Project Work / Dissertation (BHCS18D) Discipline Specific Elective - (DSE) Credit: 06

Course Objective

The students will undergo one semester of project work based on the concepts studied in a subject of their choice. The objective is to train the students for the industry by exposing them to prototype development of real life software.

Course Learning Outcomes