Unix Network Programming
Session: March' 2022 - July' 2022

1. **Course Number and Name:**

   CSE 4042, `Unix Network Programming`

2. **Credits and Course Format:**

   Grading Pattern = 1

   Credits = 4

   2 Classes/week, 1hr/Class,

   2 Labs/Week, 2hr/Lab, 4 Credits

3. **Target Students:**

   Programme: B.Tech. (6th Semester)

   Branch: CSE

4. **Instructor's Names:**

   (1) Mr. Sanjaya Kumar Jena, Assistant Professor (CSE), ITER
   `sanjayjena@soa.ac.in`

   (2) Dr. Triloknath Pandey, Associate Professor (CSE), ITER,
   `trilokpandey@soa.ac.in`

   (3) Dr. Rourab Paul, Associate Professor (CSE), ITER
   `rourabpaul@soa.ac.in`

   (4) Dr. Bharat Jyoti Ranjan Sahu, Associate Professor (CSE), ITER
   `bharatjyotisahu@soa.ac.in`

   (5) Dr. Shatarupa Dash,Assistant Professor (CSE), ITER
   `shatarupadash@soa.ac.in`

   (6) Dr. Chukhu Chunka, Assistant Professor (CSE), ITER,
   `chukhuchunka@soa.ac.in`

   (7) Dr. Dibyasundar Das, Assistant Professor (CSE), ITER
   `dibyasundardas@soa.ac.in`

   (8) Dr. Pratik Dutta, Assistant Professor (CSE), ITER
   `pratikdutta@soa.ac.in`

   (9) Dr. Sougata Sheet, Assistant Professor (CSE), ITER
   `sougatasheet@soa.ac.in`

(10) Dr. Sagarika Pattanaik, Assistant Professor (CSE), ITER
`sagarikapattnaik@soa.ac.in`

(11) Dr. Srijit Choudhury, Assistant Professor (CSE), ITER
`srijitchowdhury@soa.ac.in`

5. **Text Book and References:**
Text Book:

(1) W. Rechard Stevens, Bill Fenner, & Andrew M. Rudoff, ***UNIX Network Programming: The sockets Networking API***, Vol-1, 3rd Edition, Pearson.

References:

(1) Andrew S. Tanenbaum, ***Computer Networks***, 5th Edition, Pearson India.

(2) Kay A. Robbins & Steven Robbins, ***UNIX Systems Programming: Communication, Concurrency, & Threads***, Pearson Education

6. **Specific Course Information:**

(a) **Course Description:** This course combines an exposition of computer networking principles with network programming exercises and experiments. The course is an introduction to programming applications that use computer networks. The focus is on problem solving with emphasis on network programming. The operation and characteristics of major computer networks are studied because of their strong influence on programming interfaces (APIs) and application design.

(b) **Prerequisites and/or Co-requisites:**

☞ CSE 3034 Computer Networking

☞ CSE 3041 Unix Systems Programming

7. **Course Outcomes (COs):**
By the end of course through lectures, readings, homeworks, laboratory, assignments and exams, students will be able to:

CO 1. learn about protocols and associated algorithms for communication.

CO 2. describe the elementary socket functions required to write a complete TCP/UDP/SCTP client and server.

CO 3. write socket programs that communicate with each other using an application program interface.

CO 4. understand the name and address conversions in socket programming.

CO 5. contrast the IPV4 & IPV6 interoperability and daemon processes.

CO 6. understand unicasting, broadcasting and multicasting modes of communication.

8. **Brief List of Topics to Be Covered: (L: Lecture, P: Laboratory)**

Introduction, the transport layer, sockets introduction, Elementary TCP sockets, TCP client/server example, I/O, Socket Options, Elementary UDP and SCTP SCTP client server model, Name and address conventions, IPV4, IPV6, Daemon Processes, Advanced IO options, Unix Domain Protocols, Nonblocking I/O, IOCTL Operations, Routing Sockets, Broadcasting, Multicasting, Advanced UDP sockets, signal driven IO, threads, IP Options, Raw Sockets, Datalink access, Design Alternatives

Contact Hour	***Topics Covered***	***Remarks*** ***(if any)***
Week #1:		
L-0	Introduce the grading pattern, credit, classes and Lab session of the course. Motivation behind the course. Introduction: Uses of computer network, Network hardware, and network software	**(Class: w.e.f from XX-03-2022)** Academic regulation 2019, Tanenbaum
L-1	Reference models, examples of networks, metric units, performance: transmission time, propagation time, bandwidth, efficiency, throughput and related numericals	Tanenbaum
P-0	Practice of simple 'C' code required for unp lab: C structure, file handling and command line arguments etc., practice of some network commands: *ifconfig*, *netstat*, *ping*, *route*, etc.	Stevens Ch-1 & 2···
P-1	Introduction to IP addressing, classful and classless IP, subnet, subnet mask, Problem discussions: chapter-5 Tanenbaum exercise questions 26-33 and more similar problems, port numbers, and socket address.	Chapter-5 tanenbaum, Stevens Ch-1 & 2
Week #2:		
L-2	More workout on IP address. Introduction to transport layer, transport layer protocols, port address and socket address.	Tanenbaum, Stevens Ch-1 & Ch-2

L-3	Introduce the background of client and server communication using TCP	Tanenbaum, Forouzan
P-2	C programs using file descriptors, working with `/proc/PID/fd` for file descriptors visualization, read, write system call, create a TCP socket using `socket` function and display it's return value to identify that socket.	Steven Ch-1···
P-3	Demonstration of simple TCP daytime client-server program without details. Text Book section 1.2 and 1.5	Steven Ch-1
Week #3:		
L-4	The transport layer: overview of TCP/IP protocols, UDP, TCP, SCTP protocols, TCP connection establishment and termination, TCP state transition diagram	Stevens Ch2 2.1-2.6
L-5	SCTP association establishment and terminatio, port numbers	Stevens Ch2 2.7-2.9
P-4	**Elementary Sockets:** Sockets functions required to write complete TCP client and server, simple client server program.	Steven Ch-4, Page No: 95-97···
P-5	Program briefing socket address structure, value-result arguments, byte ordering function, byte manipulation function.	Steven Ch-3, Page No-67-80···
Week #4:		
L-6	Socket pair, TCP port numbers and concurrent servers, fork and exec functions	Stevens Ch-2, Ch-4.7,4.8
L-7	Workout on fork, and concurrent servers	Stevens Ch 4.7-4.9
P-6	Practice problems on TCP client-server models,	
P-7	*inet_aton*, *inet_addr*, *inet_ntoa*, *inet_pton*, *inet_ntop*, *sock_ntop* functions and thier uses	Steven Ch-3.6, 3.7, 3.8
Week #5:		

L-8	TCP Client/Server Examples: Introduction, TCP echo server-client	Stevens Ch-5
L-9	TCP Client/Server Examples contd...	Stevens Ch-5
P-8	Introduce *getsockname* and *getpeername* functions, uses of *fork()*, and *exec()* functions, concurrent servers	Steven Ch-3-9,
P-9	**TCP client/server examples:** Introduction, TCP echo server: *main* function, *str_echo* function, TCP Echo client: *main* function, *str_cli* functions, normal startup, and normal termination, Programs for practice-Assignments and practice	Exercise from Steven Ch-3, and 4
Week #6:		
L-10	**I/O Multiplexing:** Introduction, I/O models, blocking I/O model, nonblocking I/O model, I/O multiplexing model, signal-driven I/O model, asynchronou I/O model and comparisons.	Strevens Ch-6
L-11	**I/O Multiplexing:** Contd...	Strevens Ch-6
P-10	Practice on I/O multiplexing	Strevens Ch-6
P-11	Practice on I/O multiplexing	Strevens Ch-6
Week #7:		
L-12	POSIX Signal handling with examples, POSIX signal semantics	Steven Ch-5.8
L-13	SIGCHILD signal handling, uses of *wait* and *waitpid* functions, implementation and difference	Steven Ch-5.9, 5.11
P-12	Connection abort before *accept* returns, termination of server process, *SIGPIPE* signals, crashing of server host, and implementation , Practice on POSIX Signal handling with examples	Steven Ch-5.8
P-13	Crashing and rebooting of server host, shutdown of server host, and summary of TCP example, Practice on SIGCHILD signal handling, uses of *wait* and *waitpid* functions	Steven Ch-5.9, 5.11

Week #8:		
L-14	A mini project assignment for the students on concurrent server and multiplexing, **socket options:**	Steven Ch-7
L-15	**socket options:** contd···, *fcntl*, and *ioctl* functions	Steven Ch-7
P-14	**Elementary UDP sockets:** Introduction, *recvfrom*, *sendto* function, UDP echo server and echo client design programs	Steven Ch-8-1 - 8.6
P-15	Lost datagrams, verifying received response, server not running, connect function with UDP, Lack of flow control with UDP, Determining outgoing interface with UDP, TCP and UDP echo server using *select*	Steven Ch-8-7 - 8.15
Week #9:		
L-16	**Elementary SCTP sockets**, and SCTP client/server examples	Steven Ch-9, ch-10
L-17	**Elementary SCTP sockets**, and SCTP client/server examples contd...	Steven Ch-9, ch-10
P-16	SCTP client-server examples, Name and address conversion	Steven Ch-10, 11
P-17	SCTP client-server examples, Name and address conversion	Steven Ch-10, 11
Week #10:		
L-18	Name and address conversion	Steven Ch-11
L-19	IPV4 and IPV6 interoperability: IPV4 client-IPV6 server, IPV6 client-IPV6 server	
P-18	**Exercise:** 12.1 and 12.2	Stevens page no-362
P-19	Introduction to Deamon processes and design of protocol-independent daytime server that can be invoked by `inetd`	Stevens Ch-13
Week #11:		

L-20	Contd.. `inetd` superserver, **Exercise:** 13	Stevens page no-380
L-21	**Advanced IO functions:** socket timeouts, connect with a timeout, `recv`, `send`, `readv`, and `writev` functions in socket programming. `recvmsg` and `sendmsg` functions. Techniques to see on how much data is queued to be read on a socket, without reading the data.	Stevens page no-381
P-20	kqueue interface: Set up `kevent` structure for `kqueue`, create kqueue and add filters, loop forever, blocking in kevent and loop through returned events with example	Stevens page no-406
P-21	Practice: `str_cli` function using kqueue, Exercise page no-409	
Week #12:		
L-22	**Introduction to Unix domain protocols:** Unix domain socket address structure, Unix domain stream client-server program development, Unix domain datagram client-server program development	Stevens chapter-15
L-23	Exercise chapter 15	Stevens page no-432
P-22	**Introduction to nonblocking IO:** nonblocking read-write, connect, accept and program development	
P-23	**Practice:** Exercise on nonblocking IO	Stevens Chapter-16, page no-464
Week #13:		
L-24	`ioctl` **operations:** ioctl function, socket operations, file operations and interface configuration	Stevens pages 465-468
L-25	ARP cache manipulation with ioctl function, Routing table operations	Stevens page no-481
P-24	IOCTL commands and practice exercise chapter 17	Stevens page 484

P-25	**Routing socket introduction:** operations supported on a routing socket, datalink socket address structure, Example- fetch and print a routing table entry, sysctl operations, program to check whether UDP checksum are enabled or not.	Stevens Ch-18
Week #14:		
L-26	**Key management socket:** introduction, programs using key management	Stevens chapter-19
L-27	**Broadcasting:** Different forms of addressing, broadcast address, unicast vs broadcast modify **dg_cli** function using broadcasting, IP fragmentation and broadcast	Stevens chapter-20
P-26	Chapter exercise- Broadcasting	Stevens chapter 20
P-27	**Multicasting:** multicast addresses, multicast vs broadcasting on LAN, source-specific multicast, Simple Network Time Protocol and chapter exercise	Stevens chapter-21
Week #15:		
L-28	Threads, TCP echo server using Threads, IP options	Chapter 26
L-29	Introduction to RAW sockets, Raw socket creation, Rock socket input-output, ping and traceroute program	chapter 27-28
P-28	Introduction to datalink address, and programming examples and exercise	Steven Ch-16
P-29	Different client-server design alternatives	Steven ch-30

9. **Evaluation scheme (under Grading Pattern-1) out of 100%:**

Attendance:	05%
Assignments:	20%
Mid-Term:	15%
End-Term:	60% (15%(LAB Test) + 45%(End-Semester Examination))