BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
AUGS/ AGSR Division

SECOND SEMESTER 2018-2019

Course Handout (Part II)

Jan 07, 2019

In addition to part-I (General handout for all courses appended to the timetable) this portion gives further specific details regarding the course:

Course Number & Title	: **IS F462 Network Programming**
Instructor In-Charge	: **HARI BABU K**
Course Website	:**https://sites.google.com/a/pilani.bits-pilani.ac.in/network-programming/**

1. Scope and Objective of the Course

This course is intended for software engineers involved in developing, maintaining and supporting distributed and network applications in UNIX environment. The course teaches about system programming necessary for server and client programming. It teaches programming aspects of low-level protocol TCP, UDP, raw sockets, data link level access, multicast, broadcast etc. It covers the recent developments in web programming and web server technologies. It will also teach about distributed programming aspects like RPC, web services and Protocol Buffers. Course structure involves interesting assignments and labs to strengthen the concepts.

2. Text Book

T1. W. R. Stevens,UNIX Network Programming, Vol I, Networking APIs: Sockets and XTI, Pearson Education, 3rd Edition.

T2. W.R.Stevens, UNIX Network Programming, Interprocess Communication, Vol II Pearson Education, 2nd Edition.

3. Reference Books

R1. The Linux Programming Interface: Linux and UNIX System Programming Handbook by Michael Kerrisk, No Starch Press © 2010

(http://library.books24x7.com/toc.aspx?bookid=41558)

R2. W.R. Stevens, Advanced Programming in the UNIX Environment, Pearson Education, 2008.

4. Course Plan:

a. Modules

Module	Theme	Learning Objectives
I	System Programming	• To understand and practice I/O, process and signal management in Linux systems • To understand and practice Inter-process communication (IPC) mechanisms

II	Client & Server Design	• To understand various I/O models and their applications. • To understand various client and server designs and their performance. • To understand how to create a daemon.
III	Web Servers & Web Applications	• To understand architectures of contemporary Web Servers and their scalability • To understand client-side scripting. • To understand web application frameworks on server-side
IV	Socket Programming - TCP,UDP	• To understand and practice Socket API for building TCP/UDP based client-server. • To understand API required to access DNS. • To understand the configuration level options available for Socket API • To understand protocol design and implementation.
V	Socket Programming - Low level	• To understand and practice the application of raw sockets and link level access API. • To understand how to do multicast and broadcast. • To understand Unix domain sockets
VI	Distributed Programming	• To understand the concept of distributed programming and how it is different from socket programming • To understand RPC, XML-RPC, JSON-RPC, SOAP • To understand how to create Web services • To understand data exchange formats such as XDR and protocol buffers
VII	Security	• To understand security issues in programming multi-user UNIX server systems. • To understand security attacks in network-facing servers.

Lectures	Module	Topic	Reference
1	I	Unix History; Fundamental Concepts; System Programming Concepts;	R1: Chapter 1,2,3
2-4		Unix File I/O; Standard I/O Library; fcntl; ioctl; Unix Processes; Program Execution; Error Handling; Unix Signals	R1: Chapter 4, 5, 6, 13, 20, 24-26
5-7	I	Unix IPC: Pipes, FIFOs, System V Message queues , System V Semaphores, System V Shared Memory, Memory mapping;	R1: Chapter 43-49, 51-55

			T2: Chapter 3,4,6
8	IV	Overview of Transport Layer Protocols: TCP, UDP; Client-server architectures;	T1: Chapter 2 + class notes
9-11		Sockets, Sockaddr structure; TCP and UDP Socket API; TCP client-server examples; UDP examples; Socket Options;	T1 : Chapter 3-5,7, 8 R1: Chapter 59
12-13		Domain name conversion API; IPv6 differences; IPv4-IPv6-compatibility; Adding reliability to UDP applications;	T1 : Chapter 11, 12, 22
14		Protocol Implementation Issues: encoding, framing;	T1: Chapter 5 R1: Chapter 59
15-17	II	Non-Blocking I/O; I/O multiplexing; Signal driven I/O; Asynchronous I/O (POSIX API); Client and server design with select() call; shutdown(); Advanced I/O API;	T1 : Chapter 6, 14, 25 + class notes R1: Chapter 63
18-19	V	Unix domain sockets: Addressing, Socket pair, Descriptor passing , Credential passing;	T1 : Chapter 15 R1: Chapter 34, 37
20	II	Daemon processes; inetd super server, sylogd;	T1:13
21-22	II	Overview of Pthreads; Pthreads Synchronization;	R1: Chapter 29
		Non-blocking I/O; Non-blocking connect; Client alternative designs; Performance analysis;	T1: Chapter 16
23-25		Preforking models; Prethreading models; Performance analysis; Case study: Apache; The C10K problem; Event-driven architectures; Concurrency models for UDP servers;	T1 : Chapter 22, R1: Chapter 60,61 T1: Chapter 22
26-27	III	Web Servers: Case studies of Apache, Nginx, Node.js API: CGI, FastCGI, SAPI, ISAPI Scalability with server scale-out. Memcache.	Class notes
28-29	III	Application development for web-based client and servers: Client-side programming: URL, HTTP, HTML Forms, DOM, Javascript Server-side programming: N-tier vs Model view controller (MVC). Case study of a select MVC framework.	Class notes
30-31	V	Broadcasting: concepts & implementation; Multicasting: addresses; concepts, implementation; Broadcasting & multicasting in IPv6;	T1 : Chapter 20,21
32-34	V	Raw Sockets: Socket creation; input, output; ping: design & implementation; trace route: design & implementation; UDP asynchronous errors;	T1 : Chapter 28,29
35	VI	Socket programming vs RPC; SUN RPC: high level API; port mapper; rpcgen; XDR; low-level API: authentication; multithreading;	T2 : Chapter 16 Class notes
36-37	VI	Web services, REST API. Case study. Google protocol buffers.	Class Notes
38	VI	Web sockets, WebRTC with examples.	Class Notes
39-40	VII	Security issues in programming. Buffer overflow attacks. DoS	R1: Chapter 38

		attacks.	

5. Evaluation Scheme:

Component	Duration	Weightage (%)	Date, Time & Venue	Remarks
Midterm Test	90 Mins	25%	15/3 2:00 -3:30 PM	Closed Book
Lab Participation (Individual)	-	10%		Take Home
Assignments (Maximum of two members per group)	-	30%		Take Home Assignments based on modules I, II, IV, V will be in C language. Others can be in any language.
Comprehensive Examination	3 hours	35%	11/5 FN	Partly open

6. Notices:

All notices shall be displayed only on course webpage.

7. Malpractices:

While coding assignments/lab exercises you are not allowed to share source code but discussions are allowed with others. Any copying detected among groups/individuals will be reported to appropriate authority.

8. Make-up Policy:

No makeup will be given for Labs and Assignment components. For tests, however, make-up will be granted strictly on prior permission and on justifiable grounds only.

9. Chamber Consultation Hours:

Tue 4-5PM in 6121-M

Instructor-in-charge

IS F462