# M.Tech. (Computer Networks and Information Security)

## Course Structure and Syllabus

### I Year – I Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core Course I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Structures and Algorithms</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Networking</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course III</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Programming</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Elective I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Database Internals</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>2. Software Defined Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. TCP/IP Protocol Suite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Cloud Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Core Elective II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Internet of Things</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>2. Embedded Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Distributed Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Distributed Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Open Elective I</strong></td>
<td>Open Elective – 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Laboratory I</strong></td>
<td>Network Programming Lab</td>
<td>25</td>
<td>75</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Seminar I</strong></td>
<td>Semina</td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td>24</td>
<td>8</td>
<td>28</td>
</tr>
</tbody>
</table>

### I Year – II Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core Course IV</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Security</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course V</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireless Networks</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course VI</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Security Standards and Applications</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Elective III</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Internet Technologies and Services</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>2. Digital Water Marking and Steganography</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Security Threats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Core Elective IV</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Storage Area Networks</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>2. Distributed Systems Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Cyber Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Information Systems control and Audit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Open Elective II</strong></td>
<td>Open Elective – 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Laboratory II</strong></td>
<td>Network Security Lab</td>
<td>25</td>
<td>75</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Seminar II</strong></td>
<td>Semina</td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td>24</td>
<td>8</td>
<td>28</td>
</tr>
</tbody>
</table>

### II Year - I Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comprehensive Viva-Voce</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Project work Review I</strong></td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td>24</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

### II Year - II Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project work Review II</strong></td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td><strong>Project Evaluation (Viva-Voce)</strong></td>
<td></td>
<td>150</td>
<td>--</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td>24</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>
Open Electives

1. Basic Computer Programming skills are required for all open electives. Additionally, knowledge on the specified area mentioned in prerequisites is required for opting the open elective.

2. Note: A student can register for any open elective subject provided that he has not already registered for the same subject.

<table>
<thead>
<tr>
<th>S.NO</th>
<th>Open Elective subject</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>&quot;R&quot; Programming</td>
<td>Maths, Statistics</td>
</tr>
<tr>
<td>2.</td>
<td>Android Application Development</td>
<td>Java</td>
</tr>
<tr>
<td>3.</td>
<td>Algorithmics</td>
<td>----------</td>
</tr>
<tr>
<td>4.</td>
<td>Big Data Analytics</td>
<td>Data Bases, Maths</td>
</tr>
<tr>
<td>5.</td>
<td>Bioinformatics</td>
<td>Data Structures</td>
</tr>
<tr>
<td>6.</td>
<td>Biometrics</td>
<td>----------</td>
</tr>
<tr>
<td>7.</td>
<td>Computer Forensics</td>
<td>Maths, Data Structures</td>
</tr>
<tr>
<td>8.</td>
<td>Cyber Security</td>
<td>Internet Technologies</td>
</tr>
<tr>
<td>9.</td>
<td>Database Internals</td>
<td>---</td>
</tr>
<tr>
<td>10.</td>
<td>Distributed Systems Security</td>
<td>Information Security</td>
</tr>
<tr>
<td>11.</td>
<td>E-Commerce</td>
<td>Internet Technologies</td>
</tr>
<tr>
<td>12.</td>
<td>Embedded Systems</td>
<td>Digital logic</td>
</tr>
<tr>
<td>13.</td>
<td>Information Security and Audit</td>
<td>---</td>
</tr>
<tr>
<td>15.</td>
<td>Internet of Things</td>
<td>Java</td>
</tr>
<tr>
<td>16.</td>
<td>Internet Technologies and Services</td>
<td>Java</td>
</tr>
<tr>
<td>17.</td>
<td>Java Programming</td>
<td>---</td>
</tr>
<tr>
<td>18.</td>
<td>Linux Programming</td>
<td>---</td>
</tr>
<tr>
<td>19.</td>
<td>Mobile Computing</td>
<td>Java</td>
</tr>
<tr>
<td>20.</td>
<td>Mobile Application Security</td>
<td>Mobile Application Development</td>
</tr>
<tr>
<td>21.</td>
<td>OpenStack Cloud Computing</td>
<td>Linux Programming</td>
</tr>
<tr>
<td>22.</td>
<td>Operations Research</td>
<td>Maths, Data Structures</td>
</tr>
<tr>
<td>23.</td>
<td>Principles of Information Security</td>
<td>---</td>
</tr>
<tr>
<td>24.</td>
<td>Scripting Languages</td>
<td>---</td>
</tr>
<tr>
<td>25.</td>
<td>Social Media Intelligence</td>
<td>---</td>
</tr>
<tr>
<td>26.</td>
<td>Software Engineering</td>
<td>---</td>
</tr>
<tr>
<td>27.</td>
<td>Storage Area Networks</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>28.</td>
<td>Web Usability</td>
<td>---</td>
</tr>
</tbody>
</table>
DATA STRUCTURES AND ALGORITHMS

Objectives:
- The fundamental design, analysis, and implementation of basic data structures.
- Basic concepts in the specification and analysis of programs.
- Principles for good program design, especially the uses of data abstraction.
- Significance of algorithms in the computer field
- Various aspects of algorithm development
- Qualities of a good solution

UNIT I
Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big Oh, Omega and Theta notations, Complexity Analysis Examples. Data structures-Linear and non linear data structures, ADT concept, Linear List ADT, Array representation, Linked representation, Vector representation, singly linked lists-insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, circular lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

UNIT II
Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack, implementation of recursion, Circular queue-insertion and deletion, Dequeue ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap, java.util package-ArrayList, Linked List, Vector classes, Stacks and Queues in java.util, Iterators in java.util.

UNIT III

UNIT IV

UNIT V
Search trees- Binary search tree-Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees-Definition and examples only, Red Black trees –Definition and examples only, B-Trees-definition, insertion and searching operations, Trees in java.util- TreeSet, Tree Map Classes, Tries(examples only),Comparison of Search trees. Text compression-Huffman coding and decoding, Pattern matching-KMP algorithm.

TEXT BOOKS:
1. Data structures, Algorithms and Applications in Java, S.Sahni, Universities Press.

REFERENCE BOOKS:
1. Java for Programmers, Deitel and Deitel, Pearson education.
6. Classic Data structures in Java, T.Budd, Addison-Wesley (Pearson Education).
7. Data structures with Java, Ford and Topp, Pearson Education.
M.Tech- I Year - I Sem. (CNIS)

COMPUTER NETWORKING

Objectives:
- To introduce the high speed networks that has spurred the development of new applications.
- To identify the design issues related to the Internet protocol (IP), entire TCP/IP protocol suite and network technologies dominating the high-speed scene.

UNIT – I

UNIT – II

UNIT-III

UNIT – IV

UNIT – V

TEXT BOOKS:

REFERENCE BOOKS:
NETWORK PROGRAMMING

Objectives:
- To understand Linux utilities
- To understand file handling, signals
- To understand IPC, network programming in Java
- To understand processes to communicate with each other across a Computer Network.

Unit-I
Introduction to Network Programming: OSI model, Unix standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application.
Sockets: Address structures, value – result arguments, Byte ordering and manipulation function and related functions Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, concurrent servers. Close function and related function.

Unit-II
TCP client server: Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.
Elementary UDP sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP, determining outgoing interface with UDP.
I/O Multiplexing: I/O Models, select function, Batch input, shutdown function, poll function, TCP Echo server,

Unit-III
socket options: getsockopt and setsockopt functions. Socket states, Generic socket option IPV6 socket option ICMIPv6 socket option IPV6 socket option and TCP socket options.
Advanced I/O Functions: Introduction, Socket Timeouts, recv and send Functions, readv and writev Functions, recvmsg and sendmsg Functions, Ancillary Data, How Much Data Is Queued? Sockets and Standard I/O, T/TCP: TCP for Transactions

Unit-IV
Elementary name and Address conversions: DNS, gethost by Name function, Resolver option, Function and IPV6 support, uname function, other networking information.
Daemon Processes and inetd Superserver – Introduction, syslogd Daemon, syslog Function, daemon_init Function, inetd Daemon, daemon_inetd Function
Broadcasting: Introduction, Broadcast Addresses, Unicast versus Broadcast, dg_cli Function Using Broadcasting, Race Conditions
Multicasting: Introduction, Multicast Addresses, Multicasting versus Broadcasting on A LAN, Multicasting on a WAN, Multicast Socket Options, mcast_join and Related Functions, dg_cli Function Using Multicasting, Receiving MBone Session Announcements, Sending and Receiving, SNTP: Simple Network Time Protocol, SNTP (Continued)

Unit-V
Raw Sockets: Introduction, Raw Socket Creation, Raw Socket Output, Raw Socket Input, Ping Program, Tracerouter Program, An ICMP Message Daemon,
Remote Login: Terminal line disciplines, Pseudo-Terminals, Terminal modes, Control Terminals, rlogin Overview, RPC Transparency Issues.

Text Books:
1. UNIX Network Programming, by W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Pearson Education
DATABASE INTERNALS

Objectives:
By the end of the course, you will know:
- History and Structure of databases
- How to design a database
- How to convert the design into the appropriate tables
- Handling Keys appropriately
- Enforcing Integrity Constraints to keep the database consistent
- Normalizing the tables to eliminate redundancies
- Querying relational data
- and processing the queries
- Storage Optimizing Strategies for easy retrieval of data through index
- Triggers, Procedures and Cursors, Transaction Management
- Distributed databases: management system concepts and Implementation

UNIT I
Database System Applications, Purpose of Database Systems, View of Data – Data Abstraction, Instances and Schemas, Data Models – the ER Model, Relational Model, Other Models – Database Languages – DDL, DML, Database Access from Applications Programs, Transaction Management, Data Storage and Querying, Database Architecture, Database Users and Administrators, ER diagrams., Relational Model: Introduction to the Relational Model – Integrity Constraints Over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design, Introduction to Views – Altering Tables and Views, Relational Algebra, Basic SQL Queries, Nested Queries, Complex Integrity Constraints in SQL, Triggers

UNIT II

UNIT III
Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control, Deadlocks – Performance of Locking – Transaction Support in SQL.
Concurrency Control: Serializability, and recoverability – Introduction to Lock Management – Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques – Concurrency Control without Locking.
Crash recovery: Introduction to Crash recovery, Introduction to ARIES, the Log, and Other Recovery related Structures, the Write-Ahead Log Protocol, Check pointing, recovering from a System Crash, Media recovery

UNIT IV
Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing – Clustered Indexes, Primary and Secondary Indexes, Index data Structures – Hash Based Indexing, Tree based Indexing
Storing data: Disks and Files: - The Memory Hierarchy – Redundant Arrays of Independent Disks.
Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM)
B+ Trees: A Dynamic Index Structure, Search, Insert, Delete.
Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendable Vs Linear Hashing.

UNIT V
Distributed databases: Introduction to distributed databases, Distributed DBMS architectures, Storing data in a distributed DBMS, Distributed catalog management, Distributed query processing Updating distributed data, Distributed transactions, Distributed concurrency control, Distributed recovery

TEXT BOOKS:

REFERENCE BOOKS:

1. Introduction to Database Systems, C.J.Date, Pearson Education.
2. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, PHI.
9. Distributed Databases Principles & Systems, Stefano Ceri, Giuseppe Pelagatti, TMH.
SOFTWARE DEFINED NETWORKS

Objectives:

- This course provides a comprehensive introduction to Software Defined Networking (SDN) and presents SDN in context with more familiar network services and challenges.
- It also offers a unique perspective of the business case and technology motivations for considering SDN solutions.
- It identifies the impact of SDN on traffic management and the potential for network service growth.
- It instills the knowledge needed to manage current and future demand and provisioning for SDN.
- It provides students with the basic concepts and explains the importance of virtualization, particularly the impact of virtualization on servers and networks.
- It also introduces students with the impact on service providers, legacy networks, and network vendors.

UNIT I

UNIT II
SDN Implementation: Introduction, SDN Implementation, SDN Design, Separation of the Control and Data Planes, Edge-Oriented Networking, SDN Operation, Service Providers and SDN.

UNIT III

UNIT IV

UNIT V

TEXT BOOKS:

REFERENCE BOOKS:
2. Software Defined Networks: A Comprehensive Approach by Paul Goransson, Chuck Black, Publisher Morgan Kaufmann.
TCP/IP PROTOCOL SUITE

Objectives:
- To Describe how the TCP/IP protocol suite works
- To Describe the functions of static and dynamic IP addresses
- To Explain the major functions of networks with the OSI seven-layer model
- To Describe the major functions of networks with the TCP/IP model

UNIT - I
Introduction to TCP/IP, The OSI Model and TCP/IP Protocol Suites, Underlying Technologies; IP Addressing, Sub netting and Super netting, CIDR, Delivery and Routing of IP Packets

UNIT - II
Internet Protocol (IP), ARP and RARP, Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP)

UNIT - III
User Datagram Protocol (UDP), Transmission Control Protocol (TCP); Routing Protocols (RIP, OSPF, HELLO and BGP)

UNIT - IV
Application Layer and Client-Server Model, BOOTP and DHCP; Domain Name System (DNS), Telnet and Rlogin

UNIT - V
File Transfer Protocol (FTP), Trivial File Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP), Hyper Text Transfer Protocol (HTTP)

TEXT BOOKS:

REFERENCE BOOK:
TCP/IP Unleashed, Pearson Education.
CLOUD COMPUTING

Objectives:
To learn the new computing model which enables shared resources on demand over the network.
To learn about the pay-per-use scenarios.
To learn about the new kind of service models and deployment models.
To learn about the virtualization technology.
To learn the python programming or various services and models.

UNIT-I
Principles of Parallel and Distributed Computing, Introduction to cloud computing, Cloud computing Architecture, cloud concepts and technologies, cloud services and platforms, Cloud models, cloud as a service, cloud solutions, cloud offerings, introduction to Hadoop and MapReduce.

UNIT –II
Cloud Platforms for Industry, Healthcare and education, Cloud Platforms in the Industry, cloud applications. Virtualization, cloud virtualization technology, deep dive: cloud virtualization, Migrating in to cloud computing, Virtual Machines Provisioning and Virtual Machine Migration Services, On the Management of Virtual Machines for cloud Infrastructure, Comet cloud, T-Systems,

UNIT-III
Enterprise cloud computing Paradigm, Federated cloud computing Architecture, SLA Management in Cloud Computing, Developing the cloud: cloud application Design.

UNIT-IV

UNIT-V
Cloud management, Organizational Readiness and change management in the cloud age ,Cloud Security ,Data security in the cloud, Legal Issues in the Cloud , Achieving Production Readiness for the cloud Services

TEXT BOOKS:
4. Cloud computing: Dr Kumar Saurab Wiley India 2011.

REFERENCES:
1. Code in the Cloud: Mark C.Chu-Carroll 2011, SPD.( Second part of IV UNIT)
2. Essentials of cloud computing : K Chandrasekharan CRC Press.
INTERNET OF THINGS

Objectives:
To introduce the terminology, technology and its applications
To introduce the concept of M2M (machine to machine) with necessary protocols
To introduce the Python Scripting Language which is used in many IoT devices
To introduce the Raspberry PI platform, that is widely used in IoT applications
To introduce the implementation of web based services on IoT devices

Unit I
Introduction to Internet of Things – Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs
Domain Specific IotTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle

Unit II
IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT
Basics of IoT System Management with NETCOZF, YANG- NETCONF, YANG, SNMP NETOPEER

Unit III
Introduction to Python - Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling
Python packages - JSON, XML, HTTPLib, URLib, SMTPLib

Unit IV
IoT Physical Devices and Endpoints - Introduction to Raspberry PI-Interfaces (serial, SPI, I2C)
Programming – Python program with Raspberry PI with focus of interfacing external gadgets, controlling output, reading input from pins.

Unit V
IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs
Webserver – Web server for IoT, Cloud for IoT, Python web application framework
Designing a RESTful web API

Text Book:
EMBEDDED SYSTEMS

Objectives:

- To explain various embedded system applications and design requirements.
- To construct embedded system hardware.
- To develop software programs to control embedded system.
- To generate product specification for embedded system.

UNIT I

UNIT II

UNIT III
Embedded Programming Concepts: Software programming in Assembly language and High Level Language, Data types, Structures, Modifiers, Loops and Pointers, Macros and Functions, object oriented Programming, Embedded Programming in C++ & JAVA

UNIT IV

UNIT V

TEXT BOOK:

REFERENCE BOOKS:
4. An Embedded Software Primer, David E. Simon, Pearson Education.
5. Micro Controllers, Ajay V Deshmukhi, TMH.
7. Introduction to Embedded Systems,Shibu K.V,TMH.
DISTRIBUTED SYSTEMS

Objectives:
- To explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are;
- To list the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions;
- To recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems;
- To design a distributed system that fulfills requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognize when this is not possible, and explain why;
- To build distributed system software using basic OS mechanisms as well as higher-level middleware and languages.

Unit-I

Unit-II
Distributed OS, Its kernel, Processes and Threads, Naming and Protection, Communication and Invocation, Virtual Memory, File Service components, Design issues, Interfaces, implementation techniques, SUN network file systems

Unit-III
SNS – a name service model, its design issues, Synchronizing physical clocks, Logical time and logical clocks, Distributed coordination. Replication and its architectural model, Consistency and request ordering, Conversation between a client and a server, Transactions, Nested Transactions.

Unit-IV
Concurrency control Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control.
Distributed Transactions and Nested Transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed Deadlocks, Transactions with replicated data, Transaction recovery, Fault tolerance, Hierarchical and group masking of faults.

Unit-V
Cryptography, Authentication and key distribution, Logics of Authentication, Digital signatures.
Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy, Release consistency and Munin, Overview of Distributed Operating systems Mach, Chorus.

TEXT BOOK:

REFERENCE BOOKS:
DISTRIBUTED COMPUTING

Objectives:
- Foundation of cooperative distributed systems engineering
- Supporting technologies with a special attention to agent-oriented paradigm
- Service-oriented computing and grid computing
- The implementation component includes a term-project

UNIT I
Introduction
The different forms of computing, The strengths and weaknesses of Distributed computing, Operating system concepts relevant to distributed computing, the architecture of distributed applications. Paradigms for Distributed Applications, choosing a Paradigm for an application (trade-offs).

UNIT II
Cluster Computing
Parallel computing overview, cluster computing – Introduction, Cluster Architecture, parallel programming models and Paradigms, Applications of Clusters.

UNIT III
Grid Computing
Introduction, Grid Computing Anatomy – Architecture, Architecture and relationship to other Distributed Technologies, Grid computing road map. Merging the Grid services Architecture with the Web Services Architecture.

UNIT IV
Open Grid Service Architecture – Introduction, Architecture and Goal, Sample Use cases: Commercial Data Center, National Fusion Collaboratory, Online Media and Entertainment. OGSA platform Components, Open Grid Services Infrastructure.

UNIT V
Globus GT 3 Toolkit – Architecture, Programming Model, A sample implementation, High Level services, OGSI.NET Middleware Solutions.

TEXT BOOKS:

REFERENCE BOOKS:
NETWORK PROGRAMMING LAB

Objectives:
- To gain hands-on experiences in installing and administering computer systems and networks, in particular, the UNIX version.
- To implement networking and Internet protocols via programming and TCP/IP protocol architecture; user datagram protocol.
- To implement shell script that accepts a list of files.

LIST OF SAMPLE PROBLEMS/EXPERIMENTS:

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script that accepts any number of arguments and prints them in the reverse order.
7. Write a shell script that determines the period for which a specified user is working on the system.
8. Write a shell script to list all of the directory files in a directory.
9. Write an interactive file-handling shell program - Let it offer the user the choice of copying, removing or linking files. Once the user has made a choice, have the program ask him for the necessary information such as the file name, new name and so on.
10. Write a shell script to find factorial of a given integer.
11. Write a shell script to find the G.C.D. of two integers.
12. Write a shell script to generate a multiplication table.
13. Write a shell script that copies multiple files to a directory.
14. Write a shell script that counts the number of lines and words present in a given file.
15. Write a shell script that displays the list of all files in the given directory.
16. Write a shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two division options: one returns the quotient and the other returns remainder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add (-a), subtract (-s), multiply (-m), quotient (-q) and reminder (-r).
17. Write a shell script to reverse the rows and columns of a matrix.
18. Write a sed command that deletes the first character in each line in a file.
19. Write sed command that deletes the character before the last character in each line of a file.
20. Write an awk script that reads a file of which each line has 5 fields – ID, NAME, MARKS1, MARKS2, MARKS3 and finds out the average for each student. Print out the average marks with appropriate messages.
21. Write an awk script to find the factorial of a user supplied number.
22. Is ls –l command produces long listing of files.
23. Write an awk script 1) to print the selected fields (Ex: size and name of the files) from the file listing. 2) to print the size of all files and number of files.
24. Write an awk script to count the number of lines in a file that do not contain vowels.
25. Write an awk script to find the number of characters, words and lines in a file.
27. Write a C program that counts the number of blanks in a text file a. Using standard I/O b. Using system calls.
28. Implement in C the following UNIX commands using system calls
   a. cat    b. ls    c. mv
29. Write a program that takes one or more file/directory names as command line input and
    reports the following information on the file.
    a. File type.
    b. Number of links.
    c. Time of last access.
    d. Read, Write and Execute permissions.
30. Write a c program to emulate the UNIX ls –l command.
31. Write a c program that creates a directory, puts a file into it, and then removes it.
32. Write a c program that searches for a file in a directory and reports whether the file is
    present in the directory or not.
33. Write a c program to list for every file in a directory, its inode number and file name.
34. Write a c program that creates a file containing hole which is occupying some space but having
    nothing.
35. Write a c program that demonstrates redirection of standard output to a file.
Ex: ls > f1.
36. Write a c program to create a child process and allow the parent to display “parent” and the child to
    display “child” on the screen.
37. Write a c program to create a Zombie process.
38. Write a c program that illustrates how an orphan is created.
39. Write a c program that creates a child process to execute a command. The command to be
    executed is passed on the command line.
40. Write a c program that accepts two small numbers as arguments and then sums the two numbers in
    a child process. The sum should be returned by child to the parent as its exit status and the parent
    should print the sum.
41. Write a c program that illustrates how to execute two commands concurrently with a command pipe.
Ex:- ls –l | sort
42. Write c programs that illustrate communication between two unrelated processes using
    named pipe.
43. Write a c program in which a parent writes a message to a pipe and the child reads
    the message.
44. Write a c program that illustrates suspending and resuming processes using signals.
45. Write a c program that displays the real time of a day every 60 seconds, 10 times.
46. Write a c program that runs a command that is input by the user and prints the exit status if the
    command completes in 5 seconds. If it doesn’t, then the parent uses kill to send a SIGTERM signal
    to kill the child process.
47. Write a C program that illustrates file-locking using semaphores.
48. Write a C program that implements a producer-consumer system with two processes. (Using
    semaphores).
49. Write client and server programs (using C) for
    b. Interaction between server and client processes using Internet Domain Sockets.
50. Write a C program (sender.c )
    a. To create a message queue with read and write permissions.
    b. To write 3 messages to it with different priority numbers.
51. Write a C program (receiver.c) that receives the messages (from the above message queue as
    specified in 63.a) and displays them.
52. Write C program that illustrates two processes communicating via shared memory.
53. Design TCP iterative Client and server application to reverse the given input sentence
54. Design TCP iterative Client and server application to reverse the given input sentence
55. Design TCP client and server application to transfer file
56. Design a TCP concurrent server to convert a given text into upper case using multiplexing system
    call “select”
57. Design a TCP concurrent server to echo given set of sentences using poll functions
58. Design UDP Client and server application to reverse the given input sentence
59. Design UDP Client server to transfer a file
60. Design using poll client server application to multiplex TCP and UDP requests for converting a given
    text into upper case.
61. Design a RPC application to add and subtract a given pair of integers
TEXT BOOKS:

4. UNIX and Shell Programming, M.G. Venkatesh Murthy, Pearson Education.