NETWORK PROTOCOL DEVELOPMENT

Advanced C:

Architecture Of Simple Computer:

- CPU
- Memory
- I/O controllers
- Executable Image contents
- Text/Code
- Data (Initialized and uninitialized)
- Heap
- Stack

Development tools and Environment:

Compiler:

1. Compilation Stages
2. Object file format

Linker:

- Function of Linker
- Executable file format
- Executable file vs Executable Image in memory

Archive or Libray utility

Make utility

Debugger

Source Code Control System (SCCS)

Project Environment development:

- Module concept
- Interface functions
- Unit testing of module
- Test Driver
- Test Stub

Functions:

- Definition, Declaration/prototype, Invocation
- Function type and return value
- Output parameters
- Pass by value and pass by reference
- Local variables

- Static variables
- Thread of execution(stack frames)
- Stdio functions
- Passing parameters to a C function from the assembly language
- Accessing the parameters from the C function from an assembly language

Pointers:

- Pointers Vs Integers
- Pointer type
- Pointer de-reference
- Pointers and arrays
- Pointer arithmetic
- Array of pointers
- Pointers and Dynamic memory
- Function pointers

Arrays:

- Valid Indexes to array
- Addresses of elements of array
- Initialization
- Using pointer as an array
- Strings
- Passing an array to a function
- Two-dimensional array initialization
- Two dimensional arrays and pointers

User Defined Datatypes:

- Structures
- Unions
- Typedef
- Enums

Structures:

- Compound type
- Packing of elements within a structure
- Alignment and hole in the structure
- Structure pointers
- Accessing elements of a structure using structure pointers
- Dynamic allocation of memory for structures
- Self referential structures
- Passing structure parameters to functions
- Returning a structure or struct pointer by a function

Unions:

- Differences between union and structure
- Uses of unions

Bit Operations:

- Binary, Decimal and Hex conversions
- Logical versus Bit wise operations
- Masking a bit
- Testing a bit
- Setting a bit
- Testing a set of bits
- Setting a set of bits

Miscellaneous:

- Big Endian and Little endian
- Ascii codes and file formats
- Interpreting the contents of a file
- As a text
- As a numbers (int, short, float or combination)
- As an image
- As a sound
- As CPU instructions

File Operations:

- Storing structures in binary format
- Storing structures in ascii format

Essential Data Structures:

Arrays:

- Operations on arrays
- Strings

Linked Lists:

- Single linked lists
- Operations on linked lists
- Double linked lists

Stacks:

Queues:

- Linear queues
- Circular queues

Search Techniques:

- Linear search
- Binary search
- Hash based search

Packets or Messages

- Framing of messages
- Parsing of messages

File Formats:

Mini Project in C:

LINUX SYSTEM PROGRAMMING

Introduction to Linux:

- Process Management
- File Management
- Memory Management
- I/O management

Unix File I/O System Calls:

- File descriptors
- File types
- Stdin, Stdout and Stderr File descriptors
- Link or Relationship between File Descriptor and File or device
- File descriptors of same file but from multiple processes
- Unix File I/O calls (unbuffered i/o)
- open, create, close, lseek, read, write, dup, dup2
- fcntl, ioctl
- File types, IDs and Access permissions

Standard I/0 Library Functions:

- fopen, fread, fwrite, fclose & fseek
- Relationship between file descriptor and FILE pointer
- Character at a time I/O
- Line at a time I/O
- Formatted I/O

Reading and Writing Structures to Files:

- In ascii format
- In Binary format
- Modifying a structure in the file

The Environment of a Unix Process:

- How C program starts and terminates as process
- Memory layout of a C Program
- Main function, Command line arguments, Environment variables
- exit(), _exit() and atexit() functions

Process System Calls:

- Process Identifiers
- fork, vfork, exit, wait, waitpid, execv

Initial Process Relationships:

- Terminal Logins

Signals:

- Signal Concepts
- Signal(), kill(), raise(), alarm() and pause()

Inter Process Communication:

- Pipes
- FIFO (Named pipes)
- Message Queues
- Semaphores
- Shared Memory

Threads:

- Multi-threaded programming
- Synchronization and Mutual exclusion for threads
- POSIX Semaphores

NETWORK PROGRAMMING

Introduction to Networking:

- Need/Uses of Networking
- Network topologies
- LAN, MAN, WAN
- Typical media used in each
- Typical protocols used in each
- LAN Standards
- Ethernet, Token Ring, Token Bus, FDDI
- Ethernet Media (Thick, Thin, Twisted pair)
- WAN Standards
- Dial-up, Leased Line
- ISDN, DSL
- ATM
- Wireless

Network Protocol Layers:

- Use of Layers
- OSI Protocol layers
- TCP/IP protocol layers

Socket Programming:

- Concept of socket / socket pair
- Concept of Client and Server
- Concept connectionless and connection oriented protocols (UDP/TCP)
- Socket calls for UDP server and client
- Socket calls for TCP server and client
- Algorithms and Issues in Client software design
- Algorithms and Issues in Server software design
- Iterative, Connectionless Servers
- Iterative Connection-Oriented servers
- Concurrent, Connection-Oriented servers
- Serving multiple clients with a single process
- Serving multiple clients with one thread per client

UDP/TCP Applications:

- TFTP
- SMTP
- HTTP
- Mixed techniques

TCP/IP Stack Internals:

Internet Addresses:

- IP Address
- Hardware Addresses
- Unicast, Broadcast, Multicast in IP and HW addresses
- Ethernet Frame format
- ARP

Internet Protocol:

- Packet format
- Fragmentation and Re-assembly
- Routing

UDP:

ICMP:

TCP:

- Timeout and Retransmission
- Flow control
- State machine
- Congestion control
- Silly window syndrome
- Socket API Interface

Software Engineering:

- Software Development Life Cycle
- Requirement Specification
- Design (High level and Detailed)
- Coding, Coding standards
- Unit testing, Unit test plan, Test drivers, test stubs
- Integration and System testing and their test plans
- Acceptance test plan

Major Project:

- Finally, student will do a major project on network or wireless or telecom protocols using VxWorks or VxWorks like RTOS.

*****Note:** This outline is comprehensive and can be tailored based on course duration, depth of coverage, and the participants expertise levels. As technology continues to evolve, it is crucial to review and update the content regularly to incorporate emerging tools, practices, and industry best standards.