A gateway to Your
Bright Future in the IT Industry

CELEBRATING 15 YEARS

QualityThought®

Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM

Eligibility:

All B.tech ECE & EEE students

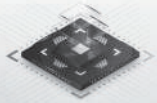
0 to 5 years students

Contact Now

89788 09743

CELEBRATING 15 YEARS
QualityThought®
Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM



CH1. Getting Started

- Why C Programming Language
- History & Features
- Compilation Model
- How to Compile & Run a C program
- Strategy of Desinging a Program

Ch2. Fundamentals of Programming

- Variables & Constants
- Keywords & Data Types
- Identifires & Rules
- I/O Functions

Ch3. Operators & Classifications

- Arithmetic Operators
- Logical Operators
- Increment Operators
- Decrement Operators
- Relational Operators
- Conditional Operators

CH4. Control FLOW STATEMENTS

- Sequential statements
- Decision making statements
- if,else,nested-if
- break,switch

CH5. Looping STATEMENTS

- For Looping
- While Looping
- Do—While Looping
- Continue Looping

CH6. C Pre-processor

- File inclusion
- Macro substitution
- Conditional Compilation
- #ifde, #ifndef

CH7. Arrays And String

- Definition and Declaration of Array
- Definition and Declaration of String
- Memory Layout & accessing Array Elements
- String Library Functions
- Two dimensional Arrays

CH8. POINTERS [PART 1]

- Definition &Declaration of Pointer
- Indirect Access using Pointers
- Pass by Reference
- Rela. b/w Arrays and Pointers
- Type Casting
- Pointer to an Array
- Array of Pointers

CH9. FUNCTIONS AND ITS TYPE

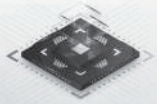
- Why Functions ?
- Function Declarations
- Function Prototypes
- Returning a Value or Not
- Arguments and Parameters
- Function Pointers
- Recursion and Recursive function

CH10. Scope & Lifetime Of Variables

- Block Scope
- Function Scope
- File Scope
- Program Scope
- The auto Specifier
- The static Specifier
- The register Specifier
- The extern Specifier
- The Const Modifier
- The Volatile Modifier

CELEBRATING 15 YEARS
QualityThought®
Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM



CH11. POINTERS

- Dynamic Storage Allocation -
- malloc(),calloc(),realloc(),free()
- Functions Returning a Pointer
- An Array of Character Pointers
- Two Dim.Arrays vs. Array of Pointers
- Command Line Arguments
- Pointers to Pointers
- Use of Function Pointers

CH13. Structures

- Fundamental Concepts
- Describing a Structure
- Operations on Structures
- Functions Returning Structures
- Passing Structures to Functions
- Pointers to Structures
- Array of Structures
- Functions Returning a Pointer to a Structure
- Structure Padding
- # pragma Definition

CH14: Structure Related (union)

- Typedef - New Name for an Existing Type
- Bit Fields
- Union
- Enumerations
- Volatile

CH15: DATA STRUCTURE USING C

- Why data structure ?
- Definition and Classification
- Singly link lists
- Circular link lists
- Double link list
- stacks and queues

CH7. Arrays And String

- Definition and Declaration of Array
- Definition and Declaration of String
- Memory Layout & accessing Array Elements
- String Library Functions
- Two dimensional Arrays

Micro Controller Programming with ARM Architecture

1. Introduction to Microcontroller
2. Introduction to Embedded Systems
3. Embedded Interfacing
4. Architecture of ARM
5. Tool setup & IDE
6. ARM Programming
7. GPIO Operations
8. PIN Function Selection Register's
9. I/O Direction settings
10. Timers and Interrupts
11. Memory Management
12. UART Protocol Communication
13. I2C Protocol
14. SPI Protocol
15. Ethernet Protocol
16. Wireless Technologies

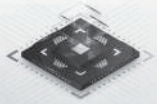
Bluetooth

Wi-fi

Gsm & GPRS

CELEBRATING 15 YEARS
QualityThought®
Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM



Linux System Programming

Introduction to Linux

- Linux History
- Linux / Unix Architecture & Components
- Linux Filesystem overview
- Memory Layout of a C Program

Development Tools & Utilities

- Compiler Stages gcc ()
- Building using the Makefile
- Binutils to decode the executable file formats – nm, objcopy, objdump etc

Exercises

- Writing a simple Makefile to compile the program
- Examine the symbols in executable
- Convert the file formats

System Calls in Linux

- Need for System Calls
- Using strace,
- Example programs on System calls

Exercises

- Locking a file & file regions
- Getting the time & formating the same

Memory Managment

- Need for memory management
- Memory Partitioning – Paging & Segmentation
- Virtual Addressing & its need
- Page Tables

Memory Manipulating Calls

Libraries

- Need for Libraries
- Types of Libraries – Static & Shared(dynamic)
- Examples on using the shared & static libraries

Exercises

- Creating the Static library & using the same in the applicaton
- Creating the shared library & dyamically linking with the same

Linux Processes

- Need for Process

Creating & Exec'ing the process

Waiting for the process termination

Zombie & Orphan process

Daemonizing the process

- Alarms & Timers

(Better to cover it as a part of signals)

Exercises

- Creating a Process
- Creating multiple processes
- Distinguishing between the child & parent processes
- Waiting & getting the child process status
- Creating a zombie & Orphan processes

Working with Files

- Overview of files in linux
- Linux File structure
- File related functions
- Seeking a file

Exercises

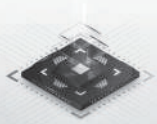
- Opening a file & performing a read/write operations
- Opening a file in different modes
- Program to copy a file

CELEBRATING 15 YEARS

QualityThought®

Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM



Managing the signals

- Need for Signals
- Overview of Signals in Linux
- Handling the signal
- Blocking/Masking the signals with signal sets

Exercises

- Registering a signal handler
- Restoring the signal disposition
- Using SIGCHLD signal to notify the parent process
- Examine the signal mask for the proces

Blocking the signal

Inter-Process Communication Mechanisms

- Need for IPCs
- Using Pipes & FIFOs
- Message Queues
- Shared Memory & Semaphores

Exercises

- Pipes to communicate amount the related processes
- Using popen function for the pipes
- Pipe Communication among unrelated processes
- Using Message Queues
- Communication using Shared Memory
- Synchronizing the access to Shared Memory

Threads

Need for Threads

Process Vs Thread

Creating thread with pthread APIs

Waiting for thread termination

- Creating a detached thread
- Cancelling the threads
- Clean-up handlers

Exercises

- Creating a thread
- Passing arguments to the thread
- Waiting for threads & examining the return value
- Modifying the thread attributes to create the detached thread
- Cancelling the threads
- Using the clean-up handler

Thread Synchronization

- Need for Synchronization
- Using Mutex
- Using Semaphores
- Reader / Writer Lock
- Conditional Variables

Exercises

- Example for mutex usage
- Example for semaphore usage
- Case study on Synchronizing threads
- Example on reader/writer Lock
- Using the Conditional variables

Introduction to Networking

- Need for Networking
- Layered Architecture
- OSI Protocol Layers and functionalities
- TCP/IP Protocol Layers
- Networking & Internetworking Devices
- Types of Networks – LAN, WAN, MAN

CELEBRATING 15 YEARS

QualityThought®

Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM

TCP/IP Stack Internals

- Internet Addressing concepts
- IP Address Vs H/W Address
- Unicast, Broadcast & Multicast addresses
- Subnettting/Supernetting
- Internet Protocol
- IP Concepts, Routing concepts
- UDP & TCP
- TCP dump
- RAW socket

Socket Programming

- Introduction to Sockets
  Socket APIsClient & Server
- Connectionless &
  Connection oriend Sockects
- Creating UDP/TCP server/Client
- Iteravite & Concurrent servers
- Iterative Connectionless &
  Conection oriented servers
- Concurrent server implementation
  using multiple processes

Exercises

- Creating local Client & Server
- Creating networked Client & Server
- Creating a connectionless sockets
- Implement concurrent & Iterative servers

WireShark Network Packet Analyzer Tool Exercises

- Using Wireshark to examine the packets

CELEBRATING 15 YEARS

QualityThought®

Transforming Dreams! Redefining Future!

Linux Kernel

Kernel Classifications

- Monolithic Kernels
- Micro Kernels
- The User space & Kernel space
- Tool Chains, Libraries, The Makefile

Module Programming

- The HelloWorld Module
- Module Stacking
- Module Parameters
- System Calls

The Virtual Filesystem

- Common Filesystem Interface
- Filesystem Abstraction Layer
- VFS Objects and Their Data Structures
- super block
- inode block
- data block
- boot block

Memory Management

- Kernel High level MMU
- Kernel Low level MMU
- Kernel Memory Allocators
- slab allocator
- page allocator
- fragment allocator
- pool allocator

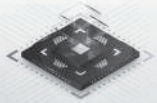
Mechanisms for Kernel Synchronization

- Reader/ Writer Semaphores
- Reader/ Writer Spinlocks
- Atomic Operations

Memory Allocation in the kernel
Kernel Timers and Time Management

- HZ & Jiffies, Delays
- Kernel Timers

EMBEDDED SYSTEM



Porc FS

- Information about processes
- communication between kernel space and user space
- /proc/devices

Character Drivers & Operations

- Registering a System Call
- System Call Handler
- Service Routines
- Character Drivers
- Synchronous Driver model

Character Drivers & Operations

- Registering a System Call
- System Call Handler
- Service Routines
- Character Drivers
- Synchronous Driver model

Device Numbers

- Major and Minor Numbers
- Registering and Unregistering
- Static and Dynamic allocations
- Important Structures

File Operations

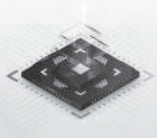
- File
- Inode
- Character Devices

cdev structure

- Adding, Allocating, Initializing & Deleting
- User Space Applications and Device Driver mapping
- Device file operations
- Access methods within the driver, open, read, write and close

CELEBRATING 15 YEARS
QualityThought®
Transforming Dreams! Redefining Future!

EMBEDDED SYSTEM



- Advanced Character Drivers
- Ioctl implementations
- Wait queues and pollings

Accessing Hardware

- Accessing I/O Ports
- Accessing I/O Memoiry

Embedded Linux Development Environment on ARM Board Beagleb one Black

- Editor in Linux (vi or vim)
- Tools for browsing source code (ctags and cscope)
- Strace diagnostic and debugging tool
- Static Tools (Lint, gcc, clang, cppcheck)
- Dynamic Tools (gcov, lcov,).
  Source code coverage tools lcov, gcov.
- GNU Build System, Configuration and generation of makefile tools
- Debuggingtools (GDB),
  Memory Profiling methodes Purify, electric-fence
  Memory Related Tools (valgrind),

Embedded Linux Development Environment on ARM Board Beagleb one Black

Toolchain Setup

- Introduction to Toolchain
- Toolchain Components
- Building Toolchain
- Toolchain compilation and usage

Bootloader Compilation

- Introduction to Bootloader
- 1 st and 2 nd Stage Bootloader
- U-Boot Bootloader Porting
- U-Boot Commands Lists
- U-Boot Image for Target Board

Clear Understading of Boot Up Sequence

- Getting Started w/ Beagle board
- Embedded Linux System boot up stages
- Beagle board boot up stages

Kernel Configuration

- Linux kernel Cross Compilation for Target board
- Browsing Linux Kernel Source
- Cross-Compilation of Kernel Source
- Generating Kernel Image
- uImage
- uImage on Target Board
- Application development and Cross Compilation

Kernel Procedures

- Booting up the kernel with NFS RootFS

Techniques for Optimizing the Boot up time

- Measuring & Analyzing the boot up time
- Optimization at Kernel space
- Optimization at User space

QualityThought

89788 09743

Quality Thought Infosystems India (P) Ltd.

#302, Nilgiri Block, Ameerpet, Hyderabad-500016

www.qualitythought.in | info@qualitythought.in