

Embedded Linux and ARM9 Applications

Synopsis

“Getting the Right Knowledge to the Right People at the Right Time”

Miracle Corporate Solutions (P) Ltd is leading in the software training industry with its unique and result oriented training methodologies on cutting edge domains. We have more than 2000 success stories in a very short span. We believe in quality and commitments. We provide skilled resources to the software industry to fulfill their acute demand of software engineers.”

ARM9 is the target processor for most of the mobile handsets today. Here we target specially the Linux based applications for an ARM9 target board. As we are moving towards 3G communication world, we are trying to put more and more services on our mobile terminal, offered through the cellular systems. Because of the limited memory and processing capability of the mobile handsets, Client/Server technologies are coming as right design possibilities for these services.

This course is designed after considering various requirements of our corporate clients. Course is also modified time to time as per the requirement variations of our clientele. Miracle is focused to provide skilled resources in embedded and mobile communication domain to the telecom software industry.

"Genius might be that ability to say a profound thing in a simple way."

—Charles Bukowski

Course Objectives:

After completing this course, students will be able to:

- Gaining Expertise in C & C++ Programming with data structures.
- Hands on debugging expertise on gdb debugger in Linux environment,
- Understanding of Real Time Operating Systems.
- Embedded Linux and embedded C Programming.
- ARM9 based target hardware - a true embedded environment.
- To develop an application to run on an embedded Linux system on ARM9.
- To understand, what is required to set-up a Linux cross development environment
- To understand, how to configure a standard Linux kernel for use in a cross development system.
- To understand, the steps to write, compile, download and debug an embedded Linux application with real hardware.
- To understand, how threaded applications fit into Linux.

Confidential

- Linux internals and device driver programming.
- Device Driver development on Linux platforms.

Suitable for

Technical professionals; IT Professionals who need to learn about current and future mobile wireless technologies; students studying or researching wireless communications and cell-phone technologies

Prerequisites

A degree (B.E., B.Tech, MCA, M.Tech) in Electronics/Electrical, Computer Science or Information Technology.

Delivery

This is instructor led Embedded Systems and Mobile Communication training. Each section of the material covered by the tutor is followed by hands-on practical exercises for which worked examples of the solutions are typically provided.

Contents

Advance C Programming

- Basics of C programming Compilation, Linking, Debugging, Execution, Header Files, Source Files Declaration & Definition
- Data Types, Scope of variables Static ,Global ,Local Variables, Loops, Conditional Statement
- Macros, Functions
- Arrays & Pointers & Structures
- Memory APIs, File I/Os.
- String Manipulations.
- VC++ 6.0 Environment (DLL/ Lib /EXE, Workspace Creation)
- Debugging
- Assignments

The Preprocessor

- The use of the pre-processor
- The basic preprocessor directives

Embedded Specifics

- Hardware access
- Use of pointers to access memory
- Use of IO ports
- Synchronization techniques
- Polling
- Interrupts
- Handling interrupts in C

Real-Time Issues

Confidential

- Scheduling
- Concurrency and resource protection
- Semaphores
- Threads and stacks

Introduction to IPC

This module includes learning of advance concepts of Linux Inter Process Communication.

- Process and Threads
- Understanding Fork() ;
- Why Inter Process Communication?
- Using FIFO,
- Using Message Queues,
- Using Socket,
- Using Pipes,
- Using Signals,
- Using FIFO,
- Using Shared Memory.
- Using Files

Developing Client Server Application

- Design and Development of a Client/Server Applications.
- Multi-thread FTP Client/Server Application with proper password authentication.

Data Structure curriculum

Introduction to DS.

- BIG –Oh notation, Algorithm complexity analysis.

Array and pointer:

- Array notation(pointer, direct etc)
- Pointer fundamentals, Pointer operations. Parameter passing as pointers etc.

Linked list Basic

- Data structures, Memory representations, Creation, Destructions,
- Insertion, Deletions(Interview specific)

Linked list Advanced

- Reversing link list(different methods)
- Doubly LL, Circular LL
- Interview questions on LL

Stack and queue

- Stack Fundamentals, implementation
- Queue Fundamentals, Implementation

Advance DS overview

- Trees, Binary Trees, Has Table etc.

Sorting and searching methods

- Various sorting techniques(Merge sort, Quick sort etc)
- Various search techniques(binary search, Radix search etc)

Embedded Linux on ARM9 Target Board

The course presents embedded and real-time concepts applied to Linux, using a ARM9 development board as the target. The host development system will be a standard PC running RedHat. We use the target as an example of a simple embedded system that can control hardware via a simple digital I/O interface.

Embedded Linux is rapidly becoming an Operating System of choice for many embedded developers. According to the LinuxDevices.com website, the significant reasons are that the source code is available, there are no runtime royalties and it is a robust reliable operating system which has excellent networking support. This course teaches embedded skills using an embedded environment. Unlike most Linux courses, that use PC's as the target, this course uses ARM9 based target hardware-a true embedded environment.

Introduction

- Real Life and Embedded Linux Systems
- Overview of Linux Strengths and Weaknesses
- Design and Implementation Methodology

Development Environment Setup

- Type of Host/Target Development Setups.
- Key development board components.
- Generic Architecture of an Embedded Linux System.
- System Startup.
- Types of Boot configurations.
- System Memory Layout.
- Type of Hardware Support.
- Cross Development Tool chain.
- Other important development tools

Introduction to Linux kernel

- Selecting a kernel
- Configuring the kernel
- Installing the kernel

Root File System Content

- Basic Root File system structure
- Libraries
- Kernel Modules
- Kernel Images
- Device Files
- Main System Applications.

- System Initialization process.

Root File System Setup

- Selecting a File system
- NFS Mounted Root File System
- Flash Mounted Root File System
- Concepts of RAM DISKS
- Concepts of CRAMFS, JFFS2

Boot loader Setup

- Boot loader basics
- Using LILO and GRUB with Disk devices
- U-Boot

Setting up Networking Services

- Network Login through Telnet
- Dynamic Configuration using DHCP
- Serving web content through DHCP

Debugging Tools

- Debugging applications using gdb
- Tracing
- Performance Analysis
- Memory Debugging
- In Circuit Debugger using JTAG interface

Embedded System Application Development

- Device Access from user space
- Device Driver Development

Linux Kernel and Device Driver Development:

Linux OS Architecture

- Overall Linux Architecture
- User/Kernel mode division
- Kernel Components
- Linux Kernel Scheduling
- Linux System Process Priorities and System Calls

Linux Driver Architecture

- Need and Use of Drivers
- Different Types of Linux Drivers

Char Driver

- Basic Concept
- Major/Minor Numbers
- Device registration
- Driver Compilations, Make files
- Loading of Kernel Drivers
- Hello World Kernel Driver
- Parameterized Hello World Kernel Driver

File Operations on Char Driver

- File Operation Concept
- Open/Close

- Read/Write
- User/Kernel Mode Memory Transfers

Driver Debugging Techniques

- Prints
- Procs File Systems
- Other Debugging Techniques

Kernel Driver Synchronization

- Basic of Synchronization
- Semaphores
- Mutexes
- Spin Locks

Kernel Memory Allocations

- Basic of Synchronization

IOCTL Usage

- Need and Use of IOCTLs
- Implementation of IOCTLs

Blocking IO Mechanism

- Blocking in Driver
- Wait_Event based Implementation

Asynchronous IO Mechanism

- Asynchronous usages
- Poll and Select Based Mechanism

Leftovers

- Linux Interrupt Handling
- Kernel Timer Concepts
- DMA and Memory Accesses

Projects

1. “Developing a Temperature Sensor driver for an ARM9 board on Linux Platform” *Miracle Corporate Solutions Ltd. NOIDA*

A complete character driver is developed for a temperature sensing device connected with our ARM9, KB9202 board. Our driver provides an easy access of this temperature sensor to various applications running on this platform. This is a user mode driver; the application can create multiple instances of this driver. This project requires the complete understanding of KB9202B board, processor data sheet and temperature sensor data sheet. Our target was to provide the real time access of sensor data to the application through a user mode driver

2. “Developing a Client/Server Application for an ARM9 board on Linux Platform to log the sensor data on a host PC” ” *Miracle Corporate Solutions Ltd. NOIDA*

The objective of this project is to develop a Remote Link between a server and various clients. This server should process the requests from various clients simultaneously. The Client application is running on target device which is a ARM9 based development board. We have various data sensors (Bluetooth, Temperature, Pressure etc.) connected to the board; Our task is to log these data on the server. The server publishes this data on the network which can be accessed by any web client.

3. “Developing a Ethernet Driver for Realtek 8192 board on Linux platform” *Miracle Corporate Solutions Ltd. NOIDA*

This driver is developed to provide Realtek 8192 card access to the applications; This is a network driver which is developed following Realtek8192 specifications and Linux guidelines. Our target was to provide initialization; data read and write access to the application through simple I/O routines. We have implemented different configuring parameters through IOCTL calls. User can also receive the events from card through simple call back routines registered to the driver.

This training experience count as a work experience that will remove the tag of fresher

NOTE: - All training will be given by Corporate Trainers only. All trainers from IIT (M.Tech.) background having four to seven years experience in respected field, working with CMM Level 5 companies

NOTE: The trainee has to sign the Non disclosure Agreement (NDA) before joining any industrial project.

Corporate Client: - LG Soft, Nokia, DCM Technologies, Mediatek, Motorola, Samsung Electronic, STMicroelectronic, Infosys, Infineon, Intel, FreeScale, Flextronics, Alcatel, Philips and many more

The major aims of the Program are to:

1. Provide a strong foundation in the emerging disciplines of RTOS, Embedded Systems and its applications for professionals in the software development industry.
2. Will improve the required skill set of the system software professional.
3. Program will provide perspective in RTOS, Embedding Systems and its applications in up coming field like Telecom, Wireless communications, Network Management, Automation and Process Controls.
4. Incorporates the required skills and experience on Embedded Systems and RTOS in the professionals for the exponentially growing industrials demands on this line.

Features:

- The study material and references will be provided to the trainees by Miracle Embedded Systems.
- This reference material is developed by our corporate trainers and software engineers from top notch industries.
- There will regular attendance of the student. One has to show at least 75% attendance then only he will be eligible for certification.
- The grades will be assigned on the basis of regular test results.
- There will be recognitions and rewards for well performing candidates.