Embedded System (Linux on ARM 9)

Course Overview: Linux on Embedded systems course takes an insight into the building, installing and using the Linux Kernel and Root File system on an embedded platform (ARM 9). It further explores the procedure to setup cross compiling environment, write kernel modules and application programs for a target board.

Duration : 3 days

Dates : As per mutual agreement

Time : 6 Hrs/Day

Eligibility : BE/BTech/ME/MTech (CSE / E&TC / Electronics) students

Corporate: Training with enhanced labs and customization for working professionals

Skills developed:

1. Learn about Key principles of embedded Linux OS
2. Porting Linux on advanced cross platform i.e. ARM 9
3. Get good expertise on building file system from scratch for target board
4. Expertise on advanced application and device driver development for target board

Results: At the end of training session student will be able to build and work Linux base embedded devices.

Pre-requisite: Basic knowledge Of Linux

NOTE: The workbook, study material and lab solutions will be provided.

DAY-1

1. Normal Vs Embedded Linux OS Architecture

- Linux Features
- Linux Kernel Source Directory structure
- Linux Kernel Components
- User Mode Vs Kernel Mode
- System Initialization –Linux Booting Process

LAB1: Exploring on kernel directory structure and files responsible for booting

2. Introduction to Cross Compiler and Makefile

- Cross Compiling Environment
- Importance of Makefiles

LAB2: Installing GNU arm gcc Cross compilation tool chain

3. Intro to the target board

(Samsung mini2440 – ARM 9)

- Minicom Utility- Connecting host to target device

LAB3: How to communicate to target device through Linux host machine

4. Kernel Configuration and compilation for target board

- Understand kernel configuration and build process
- Cross compilation of the kernel for target board
- Port Linux kernel on target board

LAB4: Cross compile and port Linux on target ARM board

DAY-2

5. The Root File system

- Creating a new root file system from scratch using Busybox
- Flashing the new root file system on target board

LAB5: Build, Cross compile and port root file system on target ARM board

6. Bring up embedded device and study its Linux OS components

- Process Management
- Thread Management
- Interrupt Management and Signal handling

LAB6: Create and Cross compile processes
LAB7: How to control Multithreading on board?
LAB8: How to handle signals on board?

7. Services on Board

- Compiling and setting up services on board

LAB9: An example service

DAY-3

6. Kernel Module programming for target board

- Introduction to Modules
- Writing Your first kernel module
- Statically linked vs Dynamically linked kernel modules
- Exporting symbols from modules
- Concurrency in the kernel
- Module Parameters and • Version dependency

LAB6: How to write kernel module and install it into the kernel?
LAB7: How to initialize and use kernel variables?
LAB8: How to achieve synchronization among kernel modules?
LAB9: How to control kernel parameters and peripherals, ports through kernel module programming?

7. Device deriver development for target board

- Types of Devices and Device Driver (DD)
- How to add new feature and functions in the kernel

LAB10: Implement char device driver example for target board

10. Debugging tools and techniques

- Introduction to kernel level debugging technique

LAB11: Demonstrate kdb/kgdb for kernel space programming

References and Guideline for Linux base embedded system