

Embitude

Let's have fun with Embedded

Online Training on Embedded Linux Porting Advance By Pradeep Tewani

Program Description

The Program provides the deeper insights into Porting the Linux for Embedded Platforms. It gets deeper into various Embedded Linux Components such as Bootloaders, Kernel & Root Filesystem. Further to this, the complete source code flow for Bootloaders & Kernel makes the Embedded Linux Porting easier. The key concepts such as Device Tree Blob (DTB) is covered in detail

Course Objective

The Embedded Linux Porting Hacks attempts to serve multiple objectives:

- Comfortability with Bootloaders Code Flow to ease the process of Porting
- Comfortability with Kernel Code Flow
- Comfortability with Board Support Packages in Bootloader & Kernel
- Complete Understanding of the Root Filesystem
- Comfortable with Device Tree Binary (DTB)

Target group:

Professionals/Students looking to expertise at Embedded Porting

Pre-requisite

Knowledge of C Programming with comfortability in Linux environment

Comfortability with Embedded Linux Fundamentals

- Boot Up Flow
- Kernel Configuration & Building
- Bootloaders Configuration & Building
- Kernel Command Linux Arguments and so on.

Embedded Linux Terminologies

- Ramdisk
- Initramfs
- Root File System and so on.

Interested in Learning the Fundamentals? Visit [Embedded Linux Porting Hacks](#)

Website: <https://embitude.in> Email: info@embitude.in M: 7760263901

Flat No. 203, V4 Bliss, 12th Cross, 4th Main, Vinayaka Layout, Bhattarahalli, K. R. Puram, Bangalore - 560049

Methology

Every theoretical topic is accompanied by corresponding hands-on/assignment to get the deep understanding of the topic.

Assessment

Assignment Based

Learning Outcome

- Understand the first stage bootloader code flow
- Understand the u-boot code flow
- Understand the Board Support Packages (BSP) in bootloaders
- Understand the Root Filesystem & its need
- Understand the Platform drivers
- Understand the Linux kernel code flow
- Understand Device Tree Blob (DTB)

Session 1: First Stage Bootloader

- Understanding the Complete Code Flow for X-loader
- Understanding the Board Support Packages (BSP) Placement

Exercises/Assignments

- Implementing the Low Level Code for Controlling the LED

Session 2: Second Stage Bootloader

- Understanding the complete code flow for the Second Stage Bootloader
- Understanding the Board Support Package (BSP) Placement

Exercises/Assignments

- Configuring the U-Boot to save the environment Variable in EEPROM
- Tweaking the U-boot code to support Multi Boot System

Session 3: U-Boot: Adding the Support for the New Board

- What does it take to add the support for new board in U-Boot
- Steps to add the support for new board

Exercises/Assignments

- Add the support for custom board in U-boot

Session 4: Root Filesystem

- What is Root Filesystem?
- What does basic minimal Root Filesystem Contain?
- What is Busybox?
- Steps to create the basic minimal Root Filesystem

Exercises/Assignments

- Creating a basic minimal Root Filesystem

- Adding the Init Script in the Root Filesystem

Session 5: Q & A and Assignments Review

Session 6: Platform Drivers

- Need for Platform Drivers
- Platform Device & Platform Device Drivers
- Registering a Platform Driver
- Binding the Driver & Device
- Passing the information from device to the driver

Exercises/Assignments

- Writing a Simple Platform Driver
- Enhancing the Platform Driver to control on-board LEDs

Session 7: Kernel Porting

- Understanding the Complete Kernel Code Flow
- Board Support Packages in Kernel
- Device Tree Binary in Linux Kernel

Exercises/Assignments

- Modifying DTB to Support the Custom Platform
- Enhancing the Platform Driver to Enable the support for DTB

Session 8: Wrap Up

- Q & A
- Assignments Review
- What Next?