

L-4 Embedded Linux

(c) *kaiwanTECH*. Click [here](#) to contact us.

Duration: 3 days	
Pre-requisites	
Mandatory	Preferable
Course L-0 “Linux CLI and Shell Scripting” <i>plus</i>	Strongly Recommended: Course level L-2 “Linux Kernel Internals”
Course C-1 “Programming in C” <i>plus</i>	
Course L-1 “Linux Systems Programming”	
<i>Below is the Outline TOC (Table Of Contents) document: it presents the (approximate) Day-wise Coverage.</i>	

Day 1

Module 1: Linux OS : Fundamental Prerequisites

- OS Architecture
 - Processor Protection Levels
 - Monolithic Kernel

Module 2: Embedded Linux Development

- Development ecosystem
 - Host and Target

GNU Toolchain

- Toolchain Components
- Building a Toolchain
 - The Roll-Your-Own Approach
 - Prebuilt Toolchains
 - Toolchain Build Utilities

Module 3 : The Linux Kernel Source Tree

- Who makes up the kernel dev community?
- Layout of the kernel source
 - A Brief Tour of the kernel source tree
- Kernel Releases
- Codebase Size

Module 4: Building a 2.6 / 3.x Linux™ Kernel

- The Kernel Repository
- The Kernel Development Process

- Download and Extraction

- Patches and Patch Management
- Configuring the Kernel
- Compiling the Kernel
- Installing the Kernel modules

Day 2

Embedded Linux Emulation with QEMU

Module 5 : Embedding 2.6 / 3.x Linux™ - A First Target: a small QEMU-based system

What is QEMU

Installation

Emulating an ARM Linux System

- ARM kernel and disk image

- Trying it out

User-Space Emulation with QEMU

Running a custom Linux kernel and Root Filesystem under QEMU

Mini-Project I : Embedding an ARM Linux kernel and (a simplistic) minimal root filesystem (no shared libraries) on an emulated e-Linux system.

Custom Scripts for building a QEMU/ARM eLinux system

Mini Project II : Embedding an ARM Linux kernel and minimal root filesystem (with shared libraries) on an emulated e-Linux system.

Day 3

Module 7 : Target Board Setup

Understanding the Board (the R Pi)

Communication between board and the host

Module 8 : Root Filesystem Content

BusyBox

Custom Applications

Libraries

System Initialization

- Standard System V init

- BusyBox init

Buildroot

- Introduction

- Configuration, uClibc-Buildroot Toolchain, Build

- Root Filesystem Choices

- Customizing

Module 9: Bootloaders

Introduction

u-boot : configuration, compilation & test cycle

Demo on an embedded-linux board / QEMU

Module 10 : Kernel Development – an Introduction

Practical Guide to Writing Kernel Code with LKMs

Setting up your Test System

The Hello World Module

Compiling, the Makefile, Insertion and Removal

Passing Parameters

Lab Assignments

Memory Management

The VM Model

Essential kernel MM APIs and their usage.

Module 11 : A Quick Overview of Kernel Debugging

Code-based techniques

The printk, ratelimiting, MSG macros

Convenience Header

Tracing with Ftrace

Analysing an Oops dump

Tools, Probing

A mention of KGDB and KDB

Kexec/Crash.

Wrap Up
