Consult our trainings :

> Digital Technologies training > Application Development training > Mobile, real time and embedded training > Embedded Linux, BSP and uBoot training

Embedded Linux, BSP and uBoot Training

Hands-on course
Duration : 5 days
Ref : BLE
Price 2020 : Contact us
+ Program

This training will bring you knowledge of the stages to add a new BSP to uBoot and Linux. You will learn how to build an embedded systems toolchain and the embedded root filesystem. Major issues related to the Linux kernel will be pointed and finally you will embed the graphical libraries and Linux systems utilities.

Hands-on work

Every step of the training session is immediately applied as a case study on an embedded ARM board with a touch screen to test graphical developments.
PROGRAM

The cross development tools

Overview of an embedded system and of the Linux kernel architecture.
Cross development tool chain, gcc cross compiler, C libraries, glibc and uClibc, GNU debugger, GNU ELF tools.
Embedded development tools, QEMU, Buildroot, Busybox and Scratchbox

The universal Boot loader: uBoot

uBoot project overview. A walk through the source code. Supported architectures. Basic functionalities.
The uImage format for booting uBoot Images.
Configuration, compilation and installation in a QEMU sandbox for testing.
Development of a standalone program using uBoot as BIOS.
uBoot BSP. Adding a new SOC and a new board in the uBoot BSP tree.

Hands-on work
Add a new command to uBoot and test uBoot inside QEMU, generate a new BSP for uBoot and develop a simple stand alone program using uBoot as BIOS.

Linux kernel

Licenses implications and kernel modules development. Development cycles.
Kernel development tools, quilt, GDB, GIT, LTT. Configuration tool Kbuild.
The Linux boot process.
Devices drivers. The Linux driver framework and standard drivers.
The Linux BSP. Adding a new board to Linux.
Specific embedded systems drivers MTD drivers, CAN, SPI and I2C drivers.

Hands-on work
Modify the kernel tree to add a new driver to the kernel tree and generate a patch formatted for the

LKML. Develop a character driver outside of the kernel tree.

Root File system

Creation of a rootfs. A tiny root file system, back to UNIX fundamentals and the init program.
Manage users on an embedded system with busybox.
Dynamic libraries or static programs. Choosing a root file system architecture.
Building a rootfs as CPIO or as EXT2 file system.
Creating JFFS2, UBIFS or YAFFS file systems.

Hands-on work
Create rootfs from scratch using busybox and test it on a real ARM target.Use buildroot to add new applications.Add your own application developed using SCRATCHBOX.Test the buildroot generated rootfs.

Limits of Linux Embedded

Industrial realtime application.
Power management.
Embedded interfaces.
Complete embedded framework.
Debugging. Using QEMU to debug and embedded system.

Hands-on work
You will modify the Linux boot-logo using standard Linux graphical tools. Test a hard realtime solution. Debug an application on the ARM target.

PDF
Print
Send

+ Participants / Prerequisite

» Participants

Embedded systems architects and developers in charge of Linux or Linux application deployment on an embedded system.

» Prerequisite

Basic Linux administration and C development know-how.

+ Intra/Tailored

For how many persons*

Preferred Cities*

Preferred Dates*

Comments

Contact Informations

Gender* Choose

Name*

Surname*

Company*

Establishment

Department*

Address*

City*

Zip code*

Country*

Phone*

Professional email*

□ By checking this box, I certify that I have read and accepted the conditions for the use of my data regarding the General Data Protection Regulation (GDPR).
You can at any time modify the use of your data and exercise your rights by sending an email to rgpd@orsys.fr

□ By checking this box, I agree to receive commercial and promotional communications from ORSYS Training*. You can unsubscribe at any time by using the link included in our communications.

Send request Cancel

Book your place

Submit your request

Time schedule

Generally, courses take place from 9:00 to 12:30 and from 14:00 to 17:30.
However, on the first day attendees are welcomed from 8:45, and there is a presentation of the session between 9:15 and 9:30.
The course itself begins at 9:30. For the 4- or 5-day hands-on courses, the sessions finish at 15:30 on the last day