Website: http://www.farsight.com.cn
Add: Room 807, A tower, Caizhi Mansion, Haidian District, Beijing, P.R.China
Zip: 100083
TEL: +86-10-82600901
FAX: +86-10-82600385
E-mail: consultant@farsight.com.cn

Embedded Linux Enterprise Training Outline

Beijing FARSIGHT Technology & Information Co., LTD

Embedded Linux Development Training Program

TRAINING PURPOSE & OBJECTIVE

Farsight offers you the most comprehensive set of Linux training courses that enable you to build embedded system solutions and stay competitive now and into the future.
Attending our training program, you will be able to:
- quickly master the methods and work procedure of embedded system
- quickly learn Linux driver development
- clearly understand the Linux system structure and the mechanism of Linux system services
- fully comprehend advanced knowledge in Linux such as how to use kernel debugging tools
- build up better abilities of driver development analyses and problem solving
- greatly improve your work efficiency in the future.

Details of Training Program & Text

The training program starts with but not limited to the basic knowledge of embedded Linux, step by step leads you to the knowledge base of most key issues related to embedded system development.

TRAINING SCHEDULE

Day 1: Basis concept and debugging for Linux application programming.
Day 2: Theory of Linux kernel and debugging method.
Day 3: Basis of Linux device driver development.
Day 4: Advanced knowledge of Linux device driver development.
Day 5: Knowledge of drivers and file systems development.

Training Content

Day 1: Basic concept and application programming/debugging

1. **Linux Kernel Overview -- The history of Linux, Introduce to GPL and Internet resources.**
2. **Cross compile environment** – Develop host and target board. Sharing file in the development environment – using NFS.
3. **Tool chain** –Introduce to Gnu tools. GCC GDB OBJDUMP STRIP etc.
4. **Elf format** – Executable and linkable format details.
5. **How to get tool chain** – How to Compile and test tool chain
6. **Other GNU tools** -- autoconf, gnu make and some others
7. **User application development.**
8. **Debugging techniques and tools** --GDB、DDD、Devrocket
9. **Linux app and lib programming** – Static shared library, inter process communication, network programming.

Experiments

† Debugging a user application

† How to build a shared library

Day 2: Linux kernel and porting, debugging

1. **Linux kernel architecture**—Linux architecture and kernel source code directories
2. **Process and scheduling** – Processes, threads, kernel process management and O(1) scheduler.
3. **Kernel memory management** – kernel space versus user space, memory management and related functions.
4. **Static real-time priority** – programming with real-time process/thread.
5. **Linux boot up sequence** – analysis on Linux booting up
6. **Porting Linux** – poring Linux to a specific hardware and analyse platform dependent code.
7. **Linux system services --** System V Init Service，Kernel log Service klogd，System log service sysklogd，Super net service xinetd.
8. **bootloader** – introduce to system boot and bootloader
9. **kernel debug techniques** – introduce to kdbg/printk/oops.

Experiment:

† Programming with real-time multi-process/multi-threads.

† Configure system service.

Day 3: Linux Device Driver Basics

1. **Linux device driver model** – Some basic concepts of linux device driver.
2. **Build and run a module** – How to install a device driver module – a case study.
3. **Module utilities** – Introduce to some module related tools.
4. **Interrupt handling** – How to install interrupt handler (ISR), interrupt sharing method.
5. **Synchronization in kernel** – The concept of atomic operation, lock, etc.
6. **Character device driver.**
7. **How to access device driver** -- communication between kernel device driver and user space process.
8. **file_operation structure**
9. **enhanced character device driver** – advanced features such as ioctl, blocked I/O
10. **Proc filesystem.**

Experiment

† Write a character device driver

† Learn hot to use proc filesystem

Day 4: Deeper into Linux Device Driver

1. **Device filesystem：devfs and sysfs**
2. **Linux memory management.**
3. **DMA operation.**
4. **Cache management**
5. **Device driver mmap**
6. **Device structures and functions.**
7. **Linux-2.6.x device driver model** —explain the new device model in linux-2.6

Experiment

† Write a block device driver

† Walk into sysfs filesystem

Day 5 Some special driver and filesystem

1. **Some special driver** -- include framebuffer, overlay, timer/high resolution timer, Irda, etc.
2. **Virtual filesystem** – Introduce to general and virtual filesystem in linux—VFS.

3. **Ramdisk**—functions of ramdisk and how ramdisk works, how to build a ramdisk.
4. **MTD architecture** – concept of MTD blaock device, MTD character device.
5. **NOR/NAND flash**—explain the two kind of flash's features and differences.
6. **Some special filesystem** – Introduce to jffs2, cramfs, tmpfs and explain their features.
7. **Building filesystem image** – how to deploy filesystem image to NOR/NAND FLASH。
8. **Using Devrocket to build a filesystem for mobile phone**
9. **Busybox** –history of Busybox, modify and compile Busybox, introduce to startup scripts.

Experiments

† Make a ramdisk

† Build jffs2 and cramfs filesystem image using Devrocke, and deploy it to target board