Publication date : 03/05/2024

Course : Embedded systems, ARM assembly programming

Practical course - 4d - 28h00 - Ref. ARM
Price : 2990 CHF E.T.

★★★★★ 5 / 5

This intensive course will teach you how to use the ARM assembler, program SoC "System on Chip" ARM9 and Cortex-A9 coprocessors and optimize libraries. It will give you all you need to develop software using this type of technology.

Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Mastering the development environment, debugger, compiler and linker
- ✓ Implement ARM assembler programs
- ✓ Integrating assembler routines into a C program
- ✓ Programming ARM9 and CortexA9 SoC coprocessors
- ✓ Opimizing bookstores

Intended audience

Computer scientists wishing to acquire comprehensive training in ARM assembler programming and the implementation of ARMv5 to ARMv7 SoCs.

Prerequisites

Basic knowledge of programming and microprocessors. Knowledge of the Linux system will facilitate practical work.

Practical details

Hands-on work

Practical work will be carried out under Linux, with an ARM emulator or on a board with an ARM processor.

Teaching methods

Theoretical presentation of the various aspects of programming, interspersed with practical exercises that enable immediate implementation of the concepts.

Course schedule

PARTICIPANTS

Computer scientists wishing to acquire comprehensive training in ARM assembler programming and the implementation of ARMv5 to ARMv7 SoCs.

PREREQUISITES

Basic knowledge of programming and microprocessors. Knowledge of the Linux system will facilitate practical work.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant’s academic progress throughout the training using multiple choice, scenarios, hands-on work and more.

Participants also complete a placement test before and after the course to measure the skills they’ve developed.

1 Introduction to ARM

- ARM architecture overview: from v5 to v7.
- ARM processor, operation, registers.
- ARM instructions. load/store architecture.
- ARMv5 architecture: TCM, Jazelle, domains, MMU.
- ARMv7 architecture: caches, monitor mode, trustzone.

Hands-on work

Get to grips with the development environment. Compiler, linker and simulator. Writing an ARM program.

2 Introducing ARM Assembler

- Basic ARM assembler operations.
- Conditional execution and branches.
- Arithmetic and logical operations.
- Memory access, battery management.
- Atomic operations.

Hands-on work

Writing various small programs in ARM assembler.

3 C interface and ARM assembler

- Use ARM assembler to optimize applications.
- Integrate assembler routines into a C program.
- Assembler function libraries.

Hands-on work

Writing a thread synchronization library in assembler.

4 Exception and device management

- ARM processor exception and interrupt handling.
- The interrupt vector.
- The different ways of managing an interruption.
- Presentation of some standard peripherals, PIC, timer, UART.

Hands-on work

Implementation of a timer interrupt on a versatile board.

5 ARM9 system programming

- Processor management by an OS.
- Operating modes. Switch to supervisor mode (SWI and SMC).
- CPSR and coprocessor management.
- The CP15 system coprocessor.
- MMU, domain and cache management.
- Barriers, Tiny Coupled Memory, Fast Context Switch.

Hands-on work

Switch to supervisor mode and back to user mode. Building a mini-OS, activating domain-based memory protection.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.

• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.

• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

6 CortexA9 system programming

- Cortex A9 to secure sensitive applications.
- Application of trustzone protection.
- Implementation of a secure monitor.
- Management of coprocessors, MMU and L1 and L2 caches.
- FCSE and ASID.

Hands-on work

Switch to monitor mode and manage a mini OS. Memory protection enabled by trustzone.