D7 - Power Management in Linux Drivers

Writing drivers with power management support

Objectives

- Configure the Linux kernel for optimal power management.
- Understand how Linux manages the power
  - To write drivers integrated in this scheme
  - To optimize low power modes
- Install and use power-management utilities

Labs are conducted on target boards, that can be:
Atmel ARM9-based boards, with Lauterbach JTAG probes.
We use the last linux kernel, as supported by the board.

Course environment

- A PC workstation and a target board per two trainees group with Lauterbach JTAG probe.
- Printed and PDF course material.
- Documentation and exercises solutions.

Prerequisite

- Good practice of C programming on Linux
- Basic knowledge of Linux kernel and driver programming (see our D3 - Linux Driverscourse)

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

First Day

Reminders on kernel programming

- Reminders on kernel module development
- Kernel objects

Exercise: Writing a kernel module creating and using kernel objects and sets

- The sysfs file system

Exercise: Interacting with a kernel module through a kernel object and the sysfs file sytem

Power-management : APM and ACPI

- The APM standard.
- The ACPI standard.

Exercise: Writing a user program to shut down the system.

Power-management, suspend and hibernate

- Freezing tasks

- Suspend in RAM
- Suspend on Disk
- Hibernate
- Debugging power management

Exercise: Writing a kernel module reacting to suspend in RAM and system wakeup

Second Day

Power management in drivers

- Device states vs System state
- Putting devices in low-power modes
- Power-management operations in drivers
- Controling system state changes from drivers

Exercise: Adapting a device driver to put the device in suspended state when unused

Cpu frequency governors

- The cpufreq user interface.
- The various governors available.

Idle Cpu governors

- The Idle task.
- The Idle device concept.
- CpuIdle governors.