Introduction to the ACPI specification and its implementation in the Linux Operation System

Max Berger
max.berger@phobos.fs.tum.de

October 15, 1999

Overview

- Overview
- ACPI is
- The ACPI specification:
  - Power Management
  - Thermal Management
  - Plug'n'Play
  - ASL
  - AML

Overview

- ACPI4Linux:
  - Bad Bioses
  - `/proc` Interface
  - Memory Management
  - APM Compatibility
  - Kernel Interface
  - AML Virtual Machine
  - ACPI in Linux 2.3.19
- Future Visions

ACPI is

- **O**perating **S**ystem defined **P**ower **M**anagement (OSPM)
- Thermal management
- Device configuration
- Platform independent
- Publically available
- A standard by Toshiba, Intel and Microsoft

ACPI is *NOT*

- The successor of APM
- Only for Windows
- Only for Intel i386 (compatible) architectures
- Able to cook coffee

Power Management

- OSPM
- Global states, sleeping states
- Hibernation and soft-off
- CPU sleeping states
- Device sleeping states

OSPM

Operating **S**ystem defined **P**ower **M**anagement

- The OS decides when to sleep
- The OS decides which sleeping state to enter
- The OS can deny sleeping
- The OS has the the ability to power the computer off *or even on*

$\Rightarrow$ All power management decisions are up to the OS

Global states, sleeping states

G0 (S0) Awake

G1 (S1,S2,S3) Sleeping

- **S1** Fast wakeup, high power consumption
- **S2** Slower wakeup, less power consumption
- **S3** Slow wakup, even less power consumption

G2 (S4,S5) Hibernation (S4) and soft-off (S5)

G3 Mechanical off

Hibernation and soft-off

- The system is close to off
- Currently known from laptop and ATX systems
- Minimal power consumption
- Wakeup by special events (modem ring, key on keyboard, RTC)
- Hibernation:
  - System context is written to disk
  - OS or BIOS hibernation

CPU sleeping states

- Defined as C0 (fully awake) to C3 (sleeping)
- Individial for each CPU
- CPU throttling (only for all CPUs)
- OS can decide which state to use based on system load

Device sleeping states

are individual for each device:

D0 Device is awake

D1/D2 Device is sleeping

- **D1** Some features are still available, less power consumption
- **D2** Less features are still available, minimal power consumption

D3 Device is off

Thermal Management

- The system is divided into "thermal zones"
- The OS decides what to do if one thermal zone reaches critical temperature
- Two ways of cooling:
  - Active cooling
  - Passive cooling
- Please Note: *The specification does* **NOT** *require an emergency shutdown!*

Active cooling

- Traditional way of cooling devices
- Devices are cooled by turning on another device, usually a fan
- Used when:
  - Performance is important (compiling, rendering, games)
  - Noise dosn't matter
  - Power consumption doesn't matter

Passive cooling

- Heat is reduced by generating less of it
- Unused devices are powered off
- Used when:
  - Power is scarce (laptops, servers)
  - Noise is important (library, speeches, professional sound applications)
  - Performance doesn't matter (eMail, simple office applications)

Plug'n'Play

ACPI defines new device enumeration:

- Includes and superseeds enumeration/configuration for "legacy" devices such as ISAPNP, PCI ($\leq$ 2.0), PCMCIA
- ACPI is enumeration standard in PCI 2.1
- enumeration of previously unknown devices (Fans)
- Platform independent

Device Configuration

Few, simple, easy functions:

CRS **C**urrent **r**essource **s**ettings

DIS **Dis**able device

PRS **P**ossible **r**essource **s**ettings

SRS **S**et **r**essource **s**ettings

ASL

ACPI control method **S**ource **L**anguage

- Assembler-like language
- (Preferred) source language for AML
- Somehow comparable to Java sourcecode

AML

ACPI control method **M**achine **L**anguage

- Platform independent but still very close to the hardware
- Can be used to write abstract drivers
- AML code is provided by an ACPI compliant BIOS
- Somehow comparable to Java bytecode

ACPI4Linux

- Started by Simon Richter and Max Berger in March 99
- In mainstream kernel since 2.3.19
- Current patch by Andrew Henroid
- Web-Site: `http://phobos.fs.tum.de/acpi/`
- Mailing List: `acpi@phobos.fs.tum.de`
- Most things are (unfortunately) only future visions

Bad Bioses

- Bad BIOS = supplies bad AML code
- Windows:
  - ACPI is turned of for bad BIOSes
- Linux:
  - ACPI is split up into smaller subsystems
  - Sysadmin decides which parts to use

Status: Planned

/proc interface

Provides in an human readable format:

- ACPI tables
- ACPI namespace
- Thermal information, such as CPU temperature
- Current sleeping states of devices

Status: Probably obsolete, will be an user space program

Memory management

- ACPI tables are in regular memory
- Tables need to be saved
- Some space can be freed after mapping into own memory
- FACS: Needs to stay reserved
- ACPI defines an `int 15h` memory extension

Status: Done

APM Compatibility

All APM tools should still work:

- Emulate `/proc/apm`
- Emulate `/dev/apm`
- Power off on shutdown
- Readjustment of clock after wakeup

Status: Planned

Kernel interface

Still not exactly clear, some ideas:

- `/dev/acpi`, probably only for ACPI-daemon
- Daemon registers with kernel
- Userprograms register with daemon
- Some things need to work without user space

Status: Heavy changes

AML Virtual Machine

- The "heart" of the ACPI implementation
- Problem: Userspace or Kernelspace?
- Current idea: Two VMs
  - Kernel VM optimized for size, not speed
  - Userspace VM built for speed and extensibility

Status: Heavy changes

ACPI in Linux 2.3.19

- ACPI is a misc device (minor 167)
- Kernel is just piping through to an user space program
- Problems:
  - Security/authorisation?
  - Device initialisation?
  - Depends on daemon for proper functionality (Thermal management?)

Status: Might stay until 2.5

Future Visions

- Programs register with ACPI, tell what ressources they need
- All devices are put to sleep unless they are needed
- The computer automatically recognized characteristical CPU loads, and selects appropriate CPU sleeping
- ACPI 2.0

Status: Planned