**ACPI Implementation in Linux 2.6**

**The Small Sleeper**

The new kernel has seen the continued advance of ACPI into the Linux mainstream. The standardized hardware configuration facilities should be familiar to most people from the kernel 2.4, but power management has now been enhanced with the addition of several sleep modes. **BY TIMO HÖNIG**

What a success story. Almost any laptop, pre-configured PC, components or operating systems you look at claim to be ACPI-aware. Since the introduction of the first version of the Advanced Configuration and Power Interface way back in 1996, the specification has aimed to finally send off the APM (Advanced Power Management) standard and the inflexible plug & play BIOS into retirement.

ACPI is often seen merely as a replacement for APM. But that is not strictly true: the C for Configuration is an integral part of ACPI, which provides a uniform and (as far as this is feasible) operating system independent hardware setup interface. Linux 2.6 uses ACPI to route PCI interrupts. If you want to use the Advanced Programmable Interrupt Controller (APIC) on a single-CPU system, instead of the venerable PIC – to avoid resource conflicts or interrupt sharing – there is no alternative to ACPI. Also, the ACPI subsystem tells the kernel which PCI devices are hotplug capable, and how to configure these devices. The routines are to be found in `drivers/pci/hotplug/acpihp_glue.c`.

ACPI uses the ACPI Source Language (ASL) and its compiled offshoot, the ACPI Machine Language to provide an abstraction layer for hardware-dependent functions. AML describes the hardware and the steps needed to access it. Every ACPI-compatible operating system has an AML interpreter for AML bytecode.

**Two-Layered Specification**

The ACPI specification [1] divides the ACPI architecture into two layers. The first of them, low level, comprises the following ACPI architecture:

- ACPI tables
- ACPI BIOS
- ACPI registers

The ACPI tables describe the ACPI hardware and its configuration in AML. The central data structure of each ACPI system uses definition blocks to stipulate how to access the hardware. When a system boots, the ACPI BIOS stores the ACPI tables in memory. The ACPI BIOS is also involved in sleep and resume operations (Suspend-to-RAM, Suspend-to-Disk).

The second, or high, level is part of the operating system and uses the ACPI core and ACPI drivers to provide an API that the low level ACPI components can access. The AML interpreter is part of this level.

**ACPI on Linux 2.6**

The kernel configuration groups the ACPI power management functions [2] below `Power management options (ACPI, APM)` (see Figure 1). If you want to use Suspend-to-RAM, you will first need to select `Prompt for development and/or incomplete code/drivers below Code maturity level options` to display the `Sleep States` option with the other ACPI options.

On booting, the kernel calls `acpi_boot_init()` to access the ACPI tables and parses them via the ACPI BIOS. The Differentiated System Description Table...
Relaxed AML

an error-free DSDT, and the
provide a BIOS upgrade with
if the manufacturer does not
DSDTs on their systems. Even
manufacturers use faulty
their configurations.

hardware components and
fications for ACPI-compatible
est. It contains the speci-
(DSDT) is of particular inter-

You can use the entries in /proc/acpi and
/sys to access the ACPI interface. Root
privileges are required to write to these
locations. If you have not mounted sysfs,
enter the following to do so:

Listing 1: Disassembling a DSDT with iasl

```
01 [root@sunshine:-]$ cp /proc/acpi/dsdt ~/dsdt
02 [root@sunshine:-]$ iasl -d ./dsdt.dsl
03 Intel ACPI Component
04 Architecture
05 ASL Optimizing Compiler / AML
Disassembler version 20030918
[Sep 18 2003]
06 Copyright (C) 2000 - 2003
Intel Corporation
07 Supports ACPI Specification
Revision 2.0b
08 Loading Acpi table from file
dsdt
09 Acpi table [DSDT] successfully
installed and loaded
10 Pass 1 parse of [DSDT]
11 Pass 2 parse of [DSDT]
12 Parsing Deferred Opcodes
(Methods/Buffers/Packages/Regi
ons)
13
14 Disassembly completed
15 Disassembly completed, written
to *dsdt.dsl*
16 [root@sunshine:-]$ head
   dsdt.dsl
17 /*
18 * Intel ACPI Component
19 * AML Disassembler version
20 20030918
21 * Disassembly of dsdt. Thu
22 Dec 11 17:28:16 2003
23 DefinitionBlock ("DSDT.aml",
"DSDT", "TOSHIB", "2000 ",
537003284)
24 |
25 Name (\_S0. Package (0x04)
26 |
27 [root@sunshine:-]$ 
```

memory, the chipset and the
peripheral devices (hard
disks, USB and so on). To
prevent loss of memory con-
tent, power is supplied to the
RAM. Suspend-to-RAM theo-
retically allows the system to
go to sleep, and wake up
again, within a few seconds.
My experience is that Sus-
pend-to-RAM rarely works. A
good thing that the kernel
still has it tagged as experi-
mental.

Suspend-to-Disk (S4) saves
more power than any
other sleep state, as it stores
the current memory content,
the registers, and the states of peripheral
devices on the hard disk (Linux uses a large swap
partition to do this) before completely
halting the system. On re-booting, Sus-
pend-to-Disk reads the content of main
memory, and the previous system state,
from the hard disk, and restores the
previous state. Linux 2.6 supports Sus-
pend-to-Disk independently of ACPI and
APM. For comparison’s sake: Linux 2.4
do not support Suspend-to-RAM at all,
although Suspend-to-Disk is possible,
using the kernel patches provided by the

Software Suspend and
Suspend-to-Disk

Confusingly, Linux 2.6 has two alterna-
tive Suspend-to-Disk (S4) methods:
Software Suspend (CONFIGSOFTWARE-
SUSPEND) and Suspend-to-Disk (CON-
FIG_PM_DISK). The former is tagged as

You can use the entries in /proc/acpi and
/sys to access the ACPI interface. Root
privileges are required to write to these
locations. If you have not mounted sysfs,
enter the following to do so:

cat /proc/acpi/dsdt >
Proc filesystem (see
Figure 1).

The sysfs /sys defaults 0 0 entry in
the /etc/fstab file ensures that sysfs will
mount automatically when you boot.

Sleep States

The individual sleep states (see Table 1)
can be manipulated via /sys/power/state.
cat /sys/power/state will tell you the
states your system supports. echo -n
“Sleep_State” /sys/power/state sets
the system to the specified state.

Suspend-to-RAM (S3) switches off the
whole system with the exception of the
main memory: that is, the CPU, cache
experimental. In contrast, Suspend-to-Disk support is a stable branch of Suspend-to-Disk.

Software Suspend assumes a resume boot prompt parameter – it is not possible to compile this into the kernel. The option tells the kernel the swap partition it should use to store the memory content and status information. It makes sense to create a permanent boot prompt parameter in the boot manager configuration file, for example `resume=/dev/hda2`.

The alternative Suspend-to-Disk support provided by Software Suspend allows you to specify the swap partition in the kernel parameter `pmdisk`. In contrast to `resume`, you can bind `pmdisk` to the kernel.

**Insomnia**

Suspend-to-RAM and Suspend-to-Disk both assume that the current drivers are suspend and resume-aware. The driver model was modified to accomplish this to suspend and resume-aware. The driver both assume that the current drivers are.

Before calling `– if this has not been done previously.

If you are experiencing difficulty both with Suspend-to-Disk (Software Suspend or Suspend-to-Disk), and on resuming, you can edit the kernel parameters manually. `resume=noresume` or `pmdisk=off` will prevent the kernel from attempting to restore the previous state on booting. The Software Suspend variant additionally requires you re-initialize the swap partition, `mkswap /dev/hda2` in our example, and then re-assign it to the system: `swapon /dev/hda2`.

**Dynamic Clock Speed for Mobile CPUs**

More recent mobile CPUs can reduce or increase their clock speeds depending on the current CPU load (Intel Speedstep and AMD Power Now!). To leverage this feature with Linux 2.6, enable `cpufreq` as shown in Figure 2 and make sure you select the correct `cpufreq` driver. The `cpufreq` interface changed during development of Linux 2.5, and is to be found in `sysfs` in the final release. If you intend to use a kernel compiled with these features, you should use `dmesg | grep cpufreq` to check if the kernel was able to initialize the feature (see Listing 3).

If `cpufreq` quits with an error message, you should first re-check that you have the correct `cpufreq` driver. As strange as this may sound, we had to enable Plug & Play BIOS kernel support (CONFIG_PNP and CONFIG_PNPBIOS) in our lab to get the `cpufreq` ACPI driver to run.

**Entries**

Assuming that `cpufreq` has initialized correctly, the following entries should be available below `/sys/devices/system/cpu/cpu0/cpuinfo_max_freq` and `cpu_min_freq` contain the maximum and minimum processor speeds for the current CPU. However, `scaling_max_freq` and `scaling_min_freq` define the clock speeds between which the CPU will actually scale. `echo speed > scaling_max_freq` and `echo speed > scaling_min_freq` allow root to change the values. `speed` must be a value between `cpuinfo_max_freq` and `cpuinfo_min_freq`.

The `scaling_available_governors` entry tells you what kind of clock speed manipulation strategies the system provides: `powersave`, `userspace`, or `perform- ance`. The currently enabled strategy is stored in `scaling_governor`. Root can enter `echo Governor > scaling_governor` to enable a different strategy, where Governor must be one of the three known strings defined in `scaling_available_governors`.

To remove the need for command line echoes, the Cpwdyn [8] project has developed a daemon that uses the `cpufreq` interface. Cpwdyn helps you modify the clock speed strategies of modern CPUs to reflect your own requirements.

**ACPI Throttling**

Many CPUs that do not support dynamic clock speed manipulation can use ACPI-based throttling instead. This typically has a positive effect on heat and noise levels, and might be a good idea for servers with nothing to do outside of office hours.

The CPU will need to support throttling. To find out whether your CPU has this support, check `/proc/acpi/processor/CPUs/throttling` for a list of available throttling states. The T0 state tells the CPU to run at full speed. In any other throttling state the CPU is slowed down by the percentage defined in `/proc/acpi/processor/CPUs/throttling`.

**ACPI for Laptops**

The ACPI features described so far are especially interesting for laptop users. `/proc/acpi/battery/BATx/info` provides

---

**Listing 2: Suspend script**

```bash
01 #!/bin/sh
02 rmmod usbblusb-storage hid;
03 # Remove USB Module
04 rmmod ohci-hcd; # Remove Host Controller Module
05 # Remove USB Core Module
06 echo -n "mem" > /sys/power/state; # Suspend to RAM
07 #modprobe usbcore; # Load USB Core Module
08 modprobe ohci-hcd; # Load Host Controller Module
09 modprobe usbblusb-storage
10 # Load USB Module
11 12 exit 0;
```
details on your machine’s battery status. The charge state, capacity, and voltage are available in /proc/acpi/battery/BATX/state. If the value drops below a specific threshold, which can be defined by typing echo threshold /proc/acpi/battery/BATX/alarm, ACPI can use /proc/acpi/events to alert the user.

The /proc/acpi/ac_adapter/ADPx/state entry tells you if the mains adapter is attached. Plugging in and unplugging the adapter is logged in /proc/acpi/events. The button driver (CONFIG_ACPI_BUTTON kernel option) and the (CONFIG_ACPI_THERMAL) thermal driver use the same approach. The button driver alerts when the power, sleep, or lid buttons are pressed. The thermal driver monitors the CPU temperature (the current value is available in /proc/acpi/thermal_zone/THRM/state) and triggers when the threshold defined in /proc/acpi/thermal_zone/THRM/ trip_points is reached.

Laptop owners can use the ACPI daemon (acpid [9]) to respond to the messages in /proc/acpi/events on their machines. When the lid is closed, acpid automatically tells the machine to suspend to disk.

Despite the variety of ACPI functions, most owners of modern laptops will soon become disillusioned when they notice that they cannot use keyboard shortcuts to change the display brightness on Linux, and that the function [Fn] keys do not provide the expected functionality. Good news for Asus and Medion notebook users: the ASUS/Medion Laptop Extras (CONFIG_ACPI_ ASUS) driver option creates entries below /proc/acpi/asus that allow you to modify the brightness of your display, and control your laptop’s LEDs. The driver converts [Fn] keyboard combinations to ACPI events. A userspace daemon [10] is available for all other functions.

Toshiba notebooks can also leverage a special ACPI driver known as the Toshiba Laptop Extras (CONFIG_ACPI_TOSHIBA). Unfortunately, the module simply uses /proc/acpi/toshiba/keys in the proc filesystem to pass [Fn] keystrokes, instead of using standard ACPI events. The other /proc/acpi/toshiba entries manipulate the brightness and the fan. There are userspace daemons available for the Toshiba extensions [11].

**Where’s the Docs?**

Users experiencing issues with Linux 2.6 and ACPI can try following their noses, and searching the Web for documentation. Unfortunately, there isn’t any. Current special publications collectively steer clear of the topic “ACPI as a Replacement for the Plug&Play BIOS”. The documentation on the kernel itself is the only exception, but it is fairly insubstantial apart from a few notes on ACPI-based power management. The best results that a spot of googling returned were the ACPI4Linux mailing list at [12] and a few newsgroups, where at least you can join forces with other people facing similar problems.

**Conclusion**

The current spate of ACPI hardware will put the operating system developers under pressure to do something about it. Some of you may remember the ACPI problems that accompanied MS Windows 2000. The goals of the ACPI specification with systematic hardware configuration and unified power management, are urgent and important.

On the upside, the recent implementation of PCI interrupt IRQ routing via ACPI on Linux is approaching stability. However, Linux 2.6 rightly tags the new power management suspend-to-RAM feature as experimental – it provides too little support to too few machines.

What’s worse, but this is not Linux’ fault, is that many systems on the market have faulty DSDTs – even though some of them are by manufacturers who are members of the ACPI consortium. Mere mortal users will probably be out of their depth if expected to provide patched DSDTs to set things straight. A lot of work still needs to be done before ACPI can really assume the role of a central interface that will configure any known hardware.

**Listing 3: The cpufreq test**

```
01 [root@sunshine:~]# dmesg | grep cpufreq
02 cpufreq: CPU0 - ACPI
performance management activated.
03 cpufreq: *P0: 750 MHz, 22000
mW, 250 uS
04 cpufreq: P1: 350 MHz, 9800 mW, 250 uS
```

**Figure 2: cpufreq ACPI options for current mobile CPUs.**

> **INFO**