

NAME

apm - APM BIOS interface

SYNOPSIS

device apm

DEPRECATION NOTICE

This driver is scheduled for removal prior to the release of FreeBSD 13.0.

DESCRIPTION

apm is an interface to the Intel / Microsoft APM (Advanced Power Management) BIOS on laptop PCs.

apm provides the following power management functions.

1. When the system wakes up from suspended mode, **apm** adjusts the system clock to RTC.
2. When the system wakes up from suspended mode, **apm** passes a message to syslogd(8) comprising of system wakeup time and elapsed time during suspended mode.
3. **apm** slows CPU clock when there are no system activities (runnable processes, interrupts, etc.). This function is available only on systems whose APM supports CPU idling.
4. **apm** exports an application interface as a character device. Applications can control APM, or retrieve APM status information via this interface. **apm** exports the following interfaces. These symbols are defined in `<machine/apm_bios.h>`.

APMIO_SUSPEND

Suspend system.

APMIO_GET

Get power management information.

APMIO_ENABLE**APMIO_DISABLE**

Enable / Disable power management.

APMIO_HALTCPU**APMIO_NOTHALTCPU**

Control execution of HLT in the kernel context switch routine.

APMIO_GETPWSTATUS

Get per battery information.

Some APM implementations execute the HLT (Halt CPU until an interrupt occurs) instruction in the "*Idle CPU*" call, while others do not. Thus enabling this may result in redundant HLT executions because "*Idle CPU*" is called from the kernel context switch routine that inherently executes HLT. This may reduce peak system performance.

Also the system hangs up if HLT instruction is disabled in the kernel context switch routine, and if the APM implementation of the machine does not execute HLT in "*Idle CPU*". On some implementations that do not support CPU clock slowdown, APM might not execute HLT. **apm** disables **APMIO_NOTHALTCPU** operation on such machines.

The current version of **apm** does not call "*Idle CPU*" from the kernel context switch routine if clock slowdown is not supported, and it executes HLT instruction by default. Therefore, there is no need to use these two operations in most cases.

These interfaces are used by apm(8).

5. **apm** polls APM events and handles the following events.

Name	Action	Description
PMEV_STANDBYREQ	suspend system	standby request
PMEV_SUSPENDREQ	suspend system	suspend request
PMEV_USERSUSPENDREQ	suspend system	user suspend request
PMEV_CRITSUSPEND	suspend system	critical suspend request
PMEV_NORMRESUME	resume system	normal resume
PMEV_CRITRESUME	resume system	critical resume
PMEV_STANDBYRESUME		

	resume system	standby resume
PMEV_BATTERYLOW		
	notify message	battery low
PMEV_UPDATETIME		
	adjust clock	update time

SEE ALSO

apm(8), zzz(8)

AUTHORS

Tatsumi Hosokawa <hosokawa@jp.FreeBSD.org>

BUGS

WARNING! Many, if not most, of the implementations of APM-bios in laptops today are buggy. You may be putting your LCD-display and batteries at a risk by using this interface. (The reason this is not a problem for MS-Windows is that they use the real-mode interface.) If you see any weird behavior from your system with this code in use, unplug the power and batteries ASAP, if not immediately, and disable this code.

We are very interested in getting this code working, so please send your observations of any anomalous behavior to us.

When **apm** is active, calling the BIOS setup routine by using hot-keys, may cause serious trouble when resuming the system. BIOS setup programs should be called during bootstrap, or from DOS.

Some APM implementations cannot handle events such as pushing the power button or closing the cover. On such implementations, the system *must* be suspended *only* by using apm(8) or zzz(8).

Disk spin-down, LCD backlight control, and power on demand have not been supported on the current version.