ACPI and FreeBSD

Part 2 (user guide)

Nate Lawson
nate@root.org

Bay Area FreeBSD Users' Group
September 6, 2006

Overview

- ACPI feature description
  - Configuration and device discovery
  - CPU power
  - Thermal
  - Video
  - Docking stations
  - Hotkeys and proprietary features
- FreeBSD ACPI (see man page)
  - Supported features
  - How to use/configure them
- How you can help

Device discovery

- ACPI handles discovery of devices that have no probe capability
  - Legacy keyboard/mouse
  - Serial, parallel, IRDA
  - Also provides standard drivers for some of these
    - SMI-activated special features (i.e. hotkeys)
    - Embedded controller
    - Batteries (SMBUS or EC-based access)
- ACPI provides additional features for standard devices
  - PCI locality and hotplug
  - ATA register data

FreeBSD Discovery

- Disable probing specific namespaces
  - `debug.acpi.avoid="ISAB"` (tunable)
  - Not necessary as probing rarely causes problems by itself these days
- Disable a specific device (not ACPI-specific)
  - `hint.acpi_thermal.0.disabled="1"` (tunable)
  - "APIC" most likely to cause problems outside of ACPI itself
    - Has some unknown problems with suspend/resume
    - As a side effect, disables SMP which sometimes interacts badly with ACPI
- Desired features
  - General mechanism for wiring device nodes to unit numbers (jhb, imp)
  - Support for hotplug PCI, large system locality, PCI Express

CPU Power

- CPU frequency/voltage
  - Save power while CPU is doing something
  - Absolute value: SpeedStep, PowerNow!, LongHaul
  - Relative value (% of absolute): acpi_throttle, TM2, nVidia
- Idle power states (C1 - Cn)
  - Save power while CPU is idle
  - Higher Cx numbers mean
    - More power savings but…
    - Longer time to transition (less responsive)
  - C1: same as HLT instruction
  - C2: clocks stopped, bus snooped
  - C3(+): clocks stopped, bus not snooped

FreeBSD CPU Power

- CPU frequency/voltage
  - Many drivers in 2 separate modules: acpi(4) and cpufreq(4)
  - Drivers cooperate with central framework (but aren’t dependent on each other) so just load both
  - Real-time management via powerd(8) or `sysctl dev.cpu.0.freq=733`
  - Disable acpi_throttle or p4tcc if experiencing hangs
- Idle power states (C1 - Cn)
  - Use `sysctl hw.acpi.cpu.cx_lowest=“C3”` or better, {performance,economy}_cx_state=“LOW” in rc.conf (see defaults/rc.conf)
  - C3 causes problems if APIC is enabled (timers stop, system hangs)
- Desired features
  - Better SMP Cx support
  - Dynamic detection of changes in available Cx states

Thermal

- Each fan or area of thermal control has a zone
- Zones offer settings and temperature monitoring
- System automatically selects, based on ACPI data
  - Cooling levels (i.e. fan speed)
  - Passive cooling (i.e. CPU slowdown or powering off devices)
  - Shutdown or suspend on critical temperatures

FreeBSD Thermal

- Manually turn on a fan
  - `hw.acpi.thermal.tz0.active=“1”` (sysctl)
  - Better to let system manage it
- If cpufreq driver(s) attached, passive cooling will automatically be engaged by acpi_thermal

Video

- Detection and control of which displays are active
- Example: enable TV out for projecting a talk
- Ambient light detection (dim automatically when in the dark)
- Auto color correction as display levels dimmed
- Most features standardized but moving to proprietary drivers (see later slides)

FreeBSD Video

- Load acpi_video(4)
- Enable a TV output
  - `hw.acpi.video.tv.active=“1”` (sysctl)
- Change the brightness
  - `hw.acpi.video.lcd.brightness=“7”` (sysctl)
- Desired features
  - Integrated support for DPMS for suspend/resume (jhb patch)
  - Ambient light, color correction, and other ACPI 3.0 features

Docking stations

- ACPI generates notifications when a docking station is attached
- It also provides a new namespace for drivers to evaluate
- Docking stations are offering a lot of features
  - Embedded PCI, floppy, printer ports
  - Motorized docking/lock mechanism

FreeBSD Docking

- Load acpi_dock(4)
- Undock a system (pressing “undock” button on dock ok too)
  - `dev.acpi_dock.0.status=“0”` (sysctl)
- Desired features
  - Dynamic probe of other device trees, not just dock
  - More testing

Hotkeys/proprietary devices

- Each manufacturer often has a proprietary driver for hotkeys, LEDs, built-in speaker volume, etc.
- Usually export system info/control via proprietary ACPI device nodes
  - Get current value of settings
  - Set new value
  - Get notify interrupt when an event happens (i.e. key pressed)
- Information on operation obtained through reverse engineering
  - "Poke it and see what happens!"

FreeBSD Hotkeys

- Load one of acpi_asus(4), acpi_ibm(4), acpi_panasonic(4), acpi_sony(4), etc.
- Read man page for specific info
- Hook a hotkey to an action via devd(8) (see /etc/devd.conf for examples)
  - Turn up volume when “volume up” hotkey pressed
- Control an LED via led(4)
  - Turn on your email light when notified by biff(1)
- Desired features
  - More support for various laptops
  - Generic hotkey script that aggregates generic codes (“volume up”) and performs actions

Conclusion

- Many ACPI functions are well-supported in FreeBSD
- Suspend/resume needs most urgent work
  - Video drivers especially problematic
  - Requires device-by-device audit
- Most ACPI functions are automatic, but with some tweaking you can save even more power

- Questions? nate@root.org