Quiet Computing with BSD

Fan control with sysctl hw.sensors

Constantine A. Murenin
コンスタンティン

University of Waterloo
Waterloo, Ontario, Canada

AsiaBSDCon 2010
Tokyo University of Science, Tokyo, Japan
11/14 March 2010

Agenda

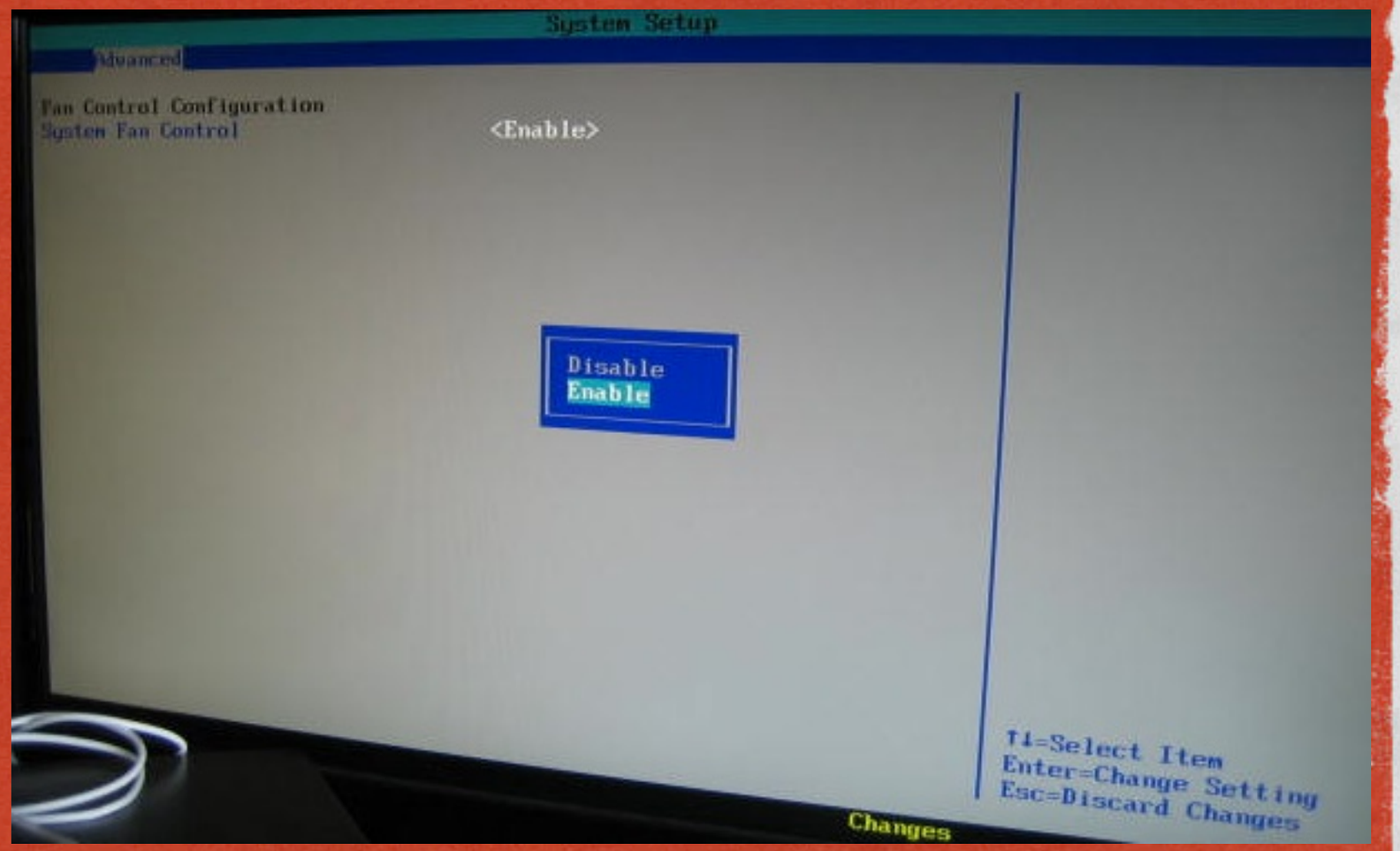
- Slow fans down, not speed ‘em up!
- Slower speed → less noise → less stress for the user
  - less stress → more motivation
- Concentrated on commonplace desktop hardware
- Winbond Super I/O chips are ubiquitous
- Don’t reinvent the wheel — let the chip do the job

Fan Control Through BIOS

Intel D201GLY2

BIOS → Advanced → Fan Control Conf. → System Fan Control

- Disable
- Enable



Fan Control Through ACPI

- *Thermal Zones* with *Active Cooling* objects
- Concept of a *Fan Device* with a binary state
- Temperature thresholds for Fan Device engagement
- Fan control is still rarely available through ACPI on common desktop hardware
- Not too useful for custom-assembled boxes anyway

Fan Control in Windows

- SpeedFan on Windows
  - Closed-source
  - Can control the fan duty cycle
  - Lacks interfacing for in-chip thermal-based control

Fan Control in Linux

- lm_sensors
  - too difficult to configure :p
  - not available on any BSD systems

Winbond Super I/O Hardware Monitors

- Several fan-controlling options:
  - Manual PWM/DC mode
  - Thermal Cruise
  - Fan Speed Cruise
  - Smart Fan III

How Do Fans Work?

- Generally, fans are rated for +12V (100% duty cycle)
- Most still run reliably at +7V (58% duty cycle)
- Few fans run at lower than +5V (42% duty cycle)

What's PWM?

- Pulse-width modulation
- Controls the amount of power sent to a load
  - voltage goes from high to low to high very rapidly, with a certain duty cycle
- More efficient when compared with rheostats
  (according to wikipedia)

Manual Mode

- The duty cycle is controlled through software
- Software-based monitoring loop would be required to accommodate reduced noise and excessive heat during high-workload
- Settings for each fan output:
  - duty cycle
  - PWM / DC and PWM frequency (if applicable)

Thermal Cruise

- Configure the chip to do the monitoring loop
- Settings:
  - target temperature and tolerance (e.g. +55℃ ±2℃)
  - fan start-up (e.g. 60%) and fan stop values (e.g. 40%)
  - should fan be turned off completely?
  - stop time / step-down time / step-up time

Fan Speed Cruise

- Settings:
  - target speed and tolerance
  - step-down time / step-up time
- But what's the benefit vs. the Manual mode?

Smart Fan III

- An advanced self-adjusting algorithm with variable target temperature and tolerance levels
- See Winbond datasheets for details

OpenBSD sysctl hw.sensors

- The fan-controlling patch released on 2009-05-08 on tech@openbsd.org
  - **http://sensors.cnst.su/fanctl/**tech@openbsd.org.2009-05-08.fanctl.patch.eml
- The following families of chips are supported:
  - W83627HF (PWM)
  - W83627THF / W83637HF (manual / thermal)
  - W83627EHF / W83627DHG (manual / thermal)

DragonFly BSD

- The fan-controlling patch was ported from OpenBSD to DragonFly BSD yesterday
- Available in the *fanctl* branch on git://leaf.dragonflybsd.org/~cnst/dragonfly.git
- Functionality is currently the same as with OpenBSD
- Will likely be integrated into the *master* branch after more testing

The Hack

- Integer values passed back to individual sensors through sysctl (not *struct sensor* itsels!)
- In the regular refresh procedure, sensor drivers check if a value has been placed, and update the chip if it has
- Chip readings are always taken from hardware
- Chip behaviour is NOT modified by this patch unless the user specifically requests so via sysctl

W83627EHF / W83627DHG

- Appear to be the latest and most popular chips
- percent{0,1,2,3} — summary and duty cycle
- temp{3,4,5,6} — target temperature
- temp{7,8,9,10} — temperature tolerance
- percent{4,5,6,7} and {8,9,10,11} — Start-up and Stop
- indicator{0,1,2,3} — PWM/DC mode switch

D201GLY2 / W83627DHG

```
hw.sensors.cpu0.temp0=58.00 degC
hw.sensors.lm1.temp0=45.00 degC (Sys)
hw.sensors.lm1.temp1=51.00 degC (CPU)
hw.sensors.lm1.temp2=14.50 degC (Aux)
hw.sensors.lm1.temp3=38.00 degC (Sys Target)
hw.sensors.lm1.temp4=unknown (CPU Target)
hw.sensors.lm1.temp5=unknown (Aux Target)
hw.sensors.lm1.temp6=unknown (CPU Target)
hw.sensors.lm1.temp7=2.00 degC (Sys Tolerance)
hw.sensors.lm1.temp8=unknown (CPU Tolerance)
hw.sensors.lm1.temp9=unknown (Aux Tolerance)
hw.sensors.lm1.temp10=unknown (CPU Tolerance)
hw.sensors.lm1.fan0=1854 RPM (Sys)
hw.sensors.lm1.volt0=1.34 VDC (VCore)
hw.sensors.lm1.volt1=12.20 VDC (+12V)
hw.sensors.lm1.volt2=3.33 VDC (+3.3V)
hw.sensors.lm1.volt3=3.33 VDC (+3.3V)
hw.sensors.lm1.volt4=-3.95 VDC (-12V)
hw.sensors.lm1.volt5=0.11 VDC
hw.sensors.lm1.volt6=1.62 VDC
hw.sensors.lm1.volt7=3.28 VDC (3.3VSB)
hw.sensors.lm1.volt8=0.03 VDC (VBAT)
hw.sensors.lm1.indicator0=Off (Sys Fan PWM/DC: PWM)
hw.sensors.lm1.indicator1=Off (CPU Fan PWM/DC: PWM)
hw.sensors.lm1.indicator2=Off (Aux Fan PWM/DC: PWM)
hw.sensors.lm1.indicator3=On (CPU Fan PWM/DC: DC)
hw.sensors.lm1.percent0=100.00% (Sys Fan PWM Thermal), OK
hw.sensors.lm1.percent1=100.00% (CPU Fan PWM Manual), OK
hw.sensors.lm1.percent2=100.00% (Aux Fan PWM Manual), OK
hw.sensors.lm1.percent3=100.00% (CPU Fan DC SmartIII), OK
hw.sensors.lm1.percent4=0.39% (Sys Fan Start-up Value), CRITICAL
hw.sensors.lm1.percent5=unknown (CPU Fan Start-up Value)
hw.sensors.lm1.percent6=unknown (Aux Fan Start-up Value)
hw.sensors.lm1.percent7=unknown (CPU Fan Start-up Value)
hw.sensors.lm1.percent8=29.41% (Sys Fan Stop Value), CRITICAL
hw.sensors.lm1.percent9=unknown (CPU Fan Stop Value)
hw.sensors.lm1.percent10=unknown (Aux Fan Stop Value)
hw.sensors.lm1.percent11=unknown (CPU Fan Stop Value)
```

Controlling Problems

- Chips are terribly miswired by MB manufacturers
- Often all fans are controlled by a single source
- Sometimes fans cannot be controlled at all
- Theoretically, this stuff can cause bad interactions with BIOS/ACPI/etc; in practice, it tends to work on desktop hardware without noticeable issues
- Some BIOSes have an annoying ‘fan error’ message

Live Demonstration!

Questions?
Comments?

Constantine A. Murenin
コンスタンティン

cnst
<C++@Cns.SU>