

Fix for irq/9-acpi 93% CPU on Mac Pro 1,1

Running Linux (Debian 13 / Ubuntu)

Hardware: Apple Mac Pro 1,1 (2006) • Dual Xeon Woodcrest • EFI32 firmware • ATI Radeon HD 5770
Tested on: Debian 13.4.0 (Trixie), kernel 6.12.74, March 2026

Symptom

After installing Debian 13 (or any modern Linux), one CPU core is permanently pinned at ~93% by the kernel thread `irq/9-acpi`. The machine is stable but wastes an entire core handling a phantom ACPI interrupt storm, and may experience occasional hard crashes with no logged cause.

Diagnosis

Confirm the IRQ storm:

```
cat /proc/interrupts | grep acpi
sleep 60
cat /proc/interrupts | grep acpi
```

The counter will be climbing at roughly 27,000 increments per second.

Identify the offending GPE:

```
sudo grep -v "^          0" /sys/firmware/acpi/interrupts/gpe*
```

You will see something like:

```
/sys/firmware/acpi/interrupts/gpe11: 5187273  EN STS enabled  unmasked
```

GPE11 is the culprit — an Apple firmware event that fires continuously and means nothing under Linux.

Root Cause

Apple's ACPI implementation on the Mac Pro 1,1 generates a continuous stream of GPE11 events. macOS handles or silently ignores these, but Linux dutifully attempts to service each one, resulting in IRQ 9 firing ~27,000 times per second. This pins one CPU core and creates enough system pressure to occasionally cause hard crashes with no kernel log entry — the kernel dies in interrupt context before it can write anything to the journal.

IRQ 9 is the hardware interrupt line that ACPI uses to signal the kernel. GPE11 is the specific ACPI General Purpose Event firing constantly within ACPI, which causes all the IRQ 9 activity. IRQ 9 is the messenger; GPE11 was sending millions of unnecessary messages.

Fix — Part 1: GRUB Boot Parameters

Standard Linux ISOs will not boot on EFI32 hardware. Use the Debian mac-specific netinst image (look for the *-mac-* variant at cdimage.debian.org/debian-cd/current/amd64/iso-cd/). Once installed, edit `/etc/default/grub`:

```
sudo nano /etc/default/grub
```

Set:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet acpi=ht"
```

Then apply:

```
sudo update-grub
```

Important: `acpi=off` causes a kernel panic on this hardware. `acpi=ht` is the safe minimum — it keeps just enough ACPI for CPU enumeration while disabling the rest.

Fix — Part 2: Mask GPE11 at Boot

Create a systemd service to mask GPE11 on every boot:

```
sudo nano /etc/systemd/system/mask-gpe11.service
```

Add the following content:

```
[Unit]
Description=Mask ACPI GPE11
After=acpid.service

[Service]
Type=oneshot
ExecStart=/bin/sh -c 'echo mask | tee /sys/firmware/acpi/interrupts/gpe11'
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Enable and start the service:

```
sudo systemctl enable mask-gpe11.service
sudo systemctl start mask-gpe11.service
```

Verification

After rebooting, wait two minutes for the boot backlog to drain, then run:

```
cat /proc/interrupts | grep acpi && sleep 60 && cat /proc/interrupts | grep acpi
```

Both lines should show identical numbers — the counter has stopped. All four CPU cores are free.

Result

- ✓ IRQ storm eliminated — IRQ 9 counter frozen after boot settles
- ✓ Full CPU performance restored — all 4 cores available
- ✓ No more random hard crashes
- ✓ GPE11 masked permanently via systemd service on every boot

Note: GPE11 on the Mac Pro 1,1 corresponds to an Apple firmware event that has no meaningful handler under Linux. Masking it causes no loss of functionality — power management, thermals, and device hotplug are unaffected.