"Klidně to přeruš!"

aneb pojednání o zpracovávání HW přerušení na OS Linux

Petr Holášek / pholasek@redhat.com

Koho by měly zajímat přerušení?

- Administrátory
- Systémové inženýry
- Uživatele, které zajímá jak jejich OS funguje uvnitř

Obsah

- Co je to přerušení?
- Jak ho zpracuje kernel?
- Co je afinita přerušení?
- K čemu slouží irqbalance?

KERNEL

Přerušení

- HW upozorňuje CPU na nutnost obsluhy události
- Reprezentace v systému pomocí IRQ čísla
- Pin-based IRQ x MSI(-X)

Pin-based IRQ

- Vyvoláno elektrickým signálem na pinu sběrnice
- Pin-based IRQ může být sdílené (např. na PCI)
- Signál může "předběhnout" data

MSI (Message Signaled Interrupts)

- CPU obdrží přerušení po zápisu na určitou adresu
- Poprvé ve specifikaci PCI 2.2
- Vylepšené MSI-X poprvé u PCI 3.0
  - Podpora více přerušení pro 1 zařízení, individuálně konfigurovatelné
  - Použití u síťových karet (fronta paketů) nebo disků (porty)

Řadiče přerušení

- APIC (*Advanced Programmable Interrupt Controller*)
  - LAPIC (*local APIC*) - u CPU
  - IOAPIC (*I/O APIC*) - u zařízení
- Vzájemná komunikace po systémové sběrnici
- V minulosti speciální APIC sběrnice

IRQ domény

- Počet řadičů přerušení > 1
- => již neplatí mapovaní `intr = intr_lines[IRQ]`
- kernel knihovna `irq_domain`
- udržuje mapování Linux IRQ na HW přerušení

Interrupt handler

- Rutina volaná při přijetí přerušení
- Je nutné vyřídit požadavek rychle
- Vše zdlouhavé je delegováno na později
  - Top halves
  - Bottom halves (*tasklets*, *workqueues*)

USERSPACE

Rozhraní jádra

- `/proc/interrupts`
- `/proc/irq/X/`
- `/sys/devices/.../irq`

```
[pholasek@localhost ~]$ cat /proc/interrupts
           CPU0       CPU1
  0:         38          0  IR-IO-APIC-edge      timer
  1:       3676          0  IR-IO-APIC-edge      i8042
  8:          1          0  IR-IO-APIC-edge      rtc0
  9:      55260          0  IR-IO-APIC-fasteoi   acpi
 12:     456029          0  IR-IO-APIC-edge      i8042
 16:     514065          0  IR-IO-APIC  16-fasteoi   ehci_hcd:usb3
 18:          0          0  IR-IO-APIC  18-fasteoi   i801_smbus
 19:         30          0  IR-IO-APIC  19-fasteoi   mmc0
 23:        119          0  IR-IO-APIC  23-fasteoi   ehci_hcd:usb4
 24:          0          0  DMAR_MSI-edge      dmar0
 25:          0          0  DMAR_MSI-edge      dmar1
 26:     460503          0  IR-PCI-MSI-edge      0000:00:1f.2
 27:          1          0  IR-PCI-MSI-edge      xhci_hcd
 31:          0          0  IR-PCI-MSI-edge      xhci_hcd
 32:         27          0  IR-PCI-MSI-edge      mei_me
 33:     607203     118662  IR-PCI-MSI-edge      i915
 34:         92          0  IR-PCI-MSI-edge      snd_hda_intel
 35:    4763376          0  IR-PCI-MSI-edge      iwlwifi
 36:       8898          0  IR-PCI-MSI-edge      em1
NMI:        203        440   Non-maskable interrupts
PMI:        203        440   Performance monitoring interrupts
RES:     330885     314209   Rescheduling interrupts
CAL:       7327       1216   Function call interrupts
TLB:     264756     431348   TLB shootdowns`
```

Afinita přerušení

> *“ A natural liking for and understanding of someone or something:”*

- Určuje množinu procesorů přijímajících přerušení
- V Linuxu na více místech (cpu affinity, NUMA affinity)
- `/proc/irq/X/smp_affinity` - např. `0xf3`

Problémy distribuce přerušení

- Zařízení s vysokou frekvencí přerušení (síť.karty, disky)
- Zahlcení CPU
- Výpadky cache procesoru
- V kernelu sofistikované řízení chybí

Irqbalance

- démon běžící na většině distribucí
- na základě analýz a heuristik distribuuje přerušení na procesory
- obvykle není třeba zásah uživatele do běhu
- vstupuje do hry hlavně u intenzivnějších I/O operací
- https://github.com/Irqbalance/irqbalance

Algoritmus irqbalance

- Parsování souboru `/proc/interrupts`
  - zjištění všech přerušení a vytížení jednotlivých CPU
- Parsování hierarchie zařízení z `/sys/devices/...`
  - IRQ jsou umístěny do hierarchie
- Vyhodnocení přetížených procesorů
- Vyhodnocení nejvíce frekventovaných přerušení
- Na základě všech získaných údajů nastavení afinity jednotlivých IRQ
- `sleep(10);`

Možnosti nastavení/experimentů

- spustit intenzivnější síťovou/diskovou operaci
- `irqbalance --debug` + `watch cat /proc/interrupts`
- `--hintpolicy` - respektování afinit požadovaných ovladači
- `--banirq <IRQ>` - ignorovat určité IRQ
- ..., izolace specifických IRQ, uživatelská pravidla, atd.

Vývoj irqbalance

- Málo vývojářů, testování i patche vítány
- https://github.com/Irqbalance/irqbalance
- http://www.freelists.org/list/irqbalance

Zdroje

- *Computer Architecture: A Quantitative Approach by John L. Hennessy*
- kernel - Documentation/IRQ*
- `man irqbalance`

Děkuji za pozornost!