

Novosti u kernel-2.6-cn paketu

Počevši od kernel-2.6-cn paketa u verziji 2.6.22.6-1, pojavila se dodatna sistemska ovisnost o [irqbalance](#) [1] Debian paketu. Riječ je o korisničkom servisu koji brine o pravilnoj raspodjeli sistemskih prekida po procesorima. Naime, standardno Linux jezgra ne raspodjeljuje prekide po više procesora (bili oni fizički odvojeni procesori ili dualcore procesori), nego je za to zadužen CPU0 -- prvi procesor kojeg sustav prepozna... Tipično postoji naivno balansiranje u Linux jezgri (CONFIG_IRQBALANCE=y), koje može i ne mora biti omogućeno, međutim irqbalance nudi bitno bolje balansiranje te prepoznaje [više tipova](#) [1] izvedbe procesora i shodno tome omogućava "pametnije" raspoređivanje. Vjerojatno se pitate kakve su to sve izvedbe procesora -- to su na primjer [NUMA](#) [2] arhitekturu, [SMT](#) [3] (npr. višejezgrene procesore), [HT](#) [4] (HyperThreading procesore), itd. Tipično sustav bez balansiranja izgleda ovako:

```
posluzitelj1 $ cat /proc/interrupts
              CPU0           CPU1
 0: 3742961528             0    IO-APIC-edge  timer
 1:          215             0    IO-APIC-edge  i8042
 8:           4             0    IO-APIC-edge  rtc
 9:           0             0    IO-APIC-level  acpi
14:          61             0    IO-APIC-edge  ide0
16:           0             0    IO-APIC-level  uhci_hcd
17:           0             0    IO-APIC-level  uhci_hcd
18:           0             0    IO-APIC-level  libata, ehci_hcd
19: 148171414             0    IO-APIC-level  eth0
21:          30             0    IO-APIC-level  ioc0
22: 175952664             0    IO-APIC-level  megaraide, ioc1
NMI: 3742961451 3742961338
LOC: 3743307940 3743307941
ERR:           0
MIS:           0
```

Možemo primijetiti da svi prekidi završavaju na procesoru CPU0, sto je npr. karakteristično za sustave bez balansiranja prekida -- bilo za one koji nemaju takvu mogućnost (HT-only procesori) ili oni koji nemaju tu podršku u kernelu odnosno irqbalance servis. Poslužitelj na kojem radi irqbalance imat će raspoređene prekide među procesorima, odnosno imat će bolje raspoređeno opterećenje i niže latencije obrade prekida (što je npr. korisno za mrežne kratice na opterećenijim poslužiteljima i vatrozidima). Jasno, nije samo dovoljno naivno rasporediti prekide po svim procesorima, irqbalance raspoređuje prekide u nekoliko kategorija... više o tome je moguće pročitati u dokumentaciji (<http://irqbalance.org/documentation.php>). Pokažimo kako to izgleda u praksi:

```
posluzitelj2 $ cat /proc/interrupts
              CPU0           CPU1
 0:          144             1    IO-APIC-edge  timer
 1:           1             1    IO-APIC-edge  i8042
 4:          80           1193    IO-APIC-edge  serial
 8:           8             11    IO-APIC-edge  rtc
 9:           0             0    IO-APIC-fastioi  acpi
12:           2             2    IO-APIC-edge  i8042
16:         310          116896    IO-APIC-fastioi  uhci_hcd:usb1, libata, eth0, nvidia
17:       40830          2324    IO-APIC-fastioi  uhci_hcd:usb5, ahci, libata, libata
18:          69          69804    IO-APIC-fastioi  uhci_hcd:usb2
19:           2             1    IO-APIC-
fastioi  uhci_hcd:usb3, uhci_hcd:usb6, ehci_hcd:usb7
```

20:	0	0	IO-APIC-fasteoi	uhci_hcd:usb4, ehci_hcd:usb8
21:	77063	110	IO-APIC-fasteoi	HDA Intel
NMI:	0	0		
LOC:	539775	471482		
ERR:	0			
MIS:	0			

uto, 2007-10-23 16:29 - Dinko Korunić **Vijesti:** [Linux](#) [5]

Kuharice: [Za sistemce](#) [6]

Kategorije: [Sistemci](#) [7]

Vote: 0

No votes yet

Source URL: <https://sysportal.carnet.hr/node/310>

Links

- [1] <https://github.com/Irqbalance/irqbalance>
- [2] http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access
- [3] http://en.wikipedia.org/wiki/Simultaneous_multithreading
- [4] <http://en.wikipedia.org/wiki/Hyper-threading>
- [5] <https://sysportal.carnet.hr/taxonomy/term/11>
- [6] <https://sysportal.carnet.hr/taxonomy/term/22>
- [7] <https://sysportal.carnet.hr/taxonomy/term/36>