

Obsah

Články

Přerušení	1
Obsluha přerušení	5
Vektor přerušení	7
IRQ	7

Reference

Zdroje článků a přispěvatelé	9
Zdroje obrázků, licence a přispěvatelé	10

Licence článků

Licence	11
---------	----

Přerušení

Přerušení (anglicky **interrupt**) je v informatice metoda pro asynchronní obsluhu událostí, kdy procesor přeruší vykonávání sledu instrukcí, vykoná obsluhu přerušení a pak pokračuje v předchozí činnosti. Původně přerušení sloužilo k obsluze hardwarových zařízení, které tak signalizovaly potřebu obsloužit (tj. odebrat z vyrovnávací paměti vstupně-výstupního zařízení data nebo do ní další data nakopírovat, odtud označení *vnější přerušení*). Později byla přidána *vnitřní přerušení*, která vyvolává sám procesor, který tak oznamuje chyby vzniklé při provádění strojových instrukcí a synchronní *softwarová přerušení* vyvolávaná speciální strojovou instrukcí, která se obvykle používají pro vyvolání služeb operačního systému.

Obsluha přerušení

Přijde-li do procesoru signalizace přerušení, je v případě, že obsluha přerušení je povolena, nejprve dokončena právě rozpracovaná strojová instrukce. Pak je na zásobník uložena adresa následující strojové instrukce, která by měla být zpracována, kdyby k přerušení nedošlo. Pak je podle tabulky přerušení vyvolána obsluha přerušení, která obslouží událost, kterou přerušení vyvolalo. Obsluha přerušení je zodpovědná za to, aby na jeho konci byl uveden stav procesoru do stavu jako na jejím začátku, aby výpočet přerušené úlohy nebyl ovlivněn, což se z důvodu vyšší rychlosti obvykle dělá softwarově (některé procesory umožňují uložit svůj stav pomocí speciální strojové instrukce). Na konci obsluhy přerušení je umístěna instrukce návratu (RET, někdy speciální IRET), která vyzvedne ze zásobníku návratovou adresu a tak způsobí, že z této adresy bude vyzvednuta následující strojová instrukce. Přerušená úloha tak až na zpoždění nepozná, že proběhla obsluha přerušení.

Tabulka přerušení umožňuje, aby procesor mohl rozlišit více různých přerušení (rozlišených čísly), ke každému vyvolat odpovídající obsluhu přerušení (podprogram) a aby šlo jednotlivé obsluhy umístit na libovolná místa v paměti. Obsluha přerušení je obvykle uložena v ovladači, který spolu s novým hardwarovým zařízením do operačního systému instalujeme.

Související informace naleznete také v článku [Obsluha přerušení](#).

Typy přerušení

Vnější přerušení

Vnější přerušení (též hardwarové přerušení) je označováno podle toho, že přichází ze vstupně-výstupních zařízení (tj. z pohledu procesoru přicházejí z *vnějšku*). Vstupně-výstupní zařízení tak má možnost si asynchronně vyžádat pozornost procesoru a zajistit tak svoji obsluhu ve chvíli, kdy to právě potřebuje bez ohledu na právě zpracovávanou úlohu.

Vnější přerušení jsou do procesoru doručována prostřednictvím řadiče přerušení, což je specializovaný obvod, který umožňuje stanovit prioritu jednotlivým přerušením, rozdělovat je mezi různé procesory a další související akce.

Související informace naleznete také v článku [Řadič přerušení](#).

Vnitřní přerušeni

Vnitřní přerušeni vyvolává sám procesor, který tak signalizuje problémy při zpracování strojových instrukcí a umožňuje operačnímu systému na tyto události nejvhodnějším způsobem zareagovat. Jedná se například o pokus dělení nulou, porušení ochrany paměti, nepřítomnost matematického koprocessoru, výpadek stránky a podobně.

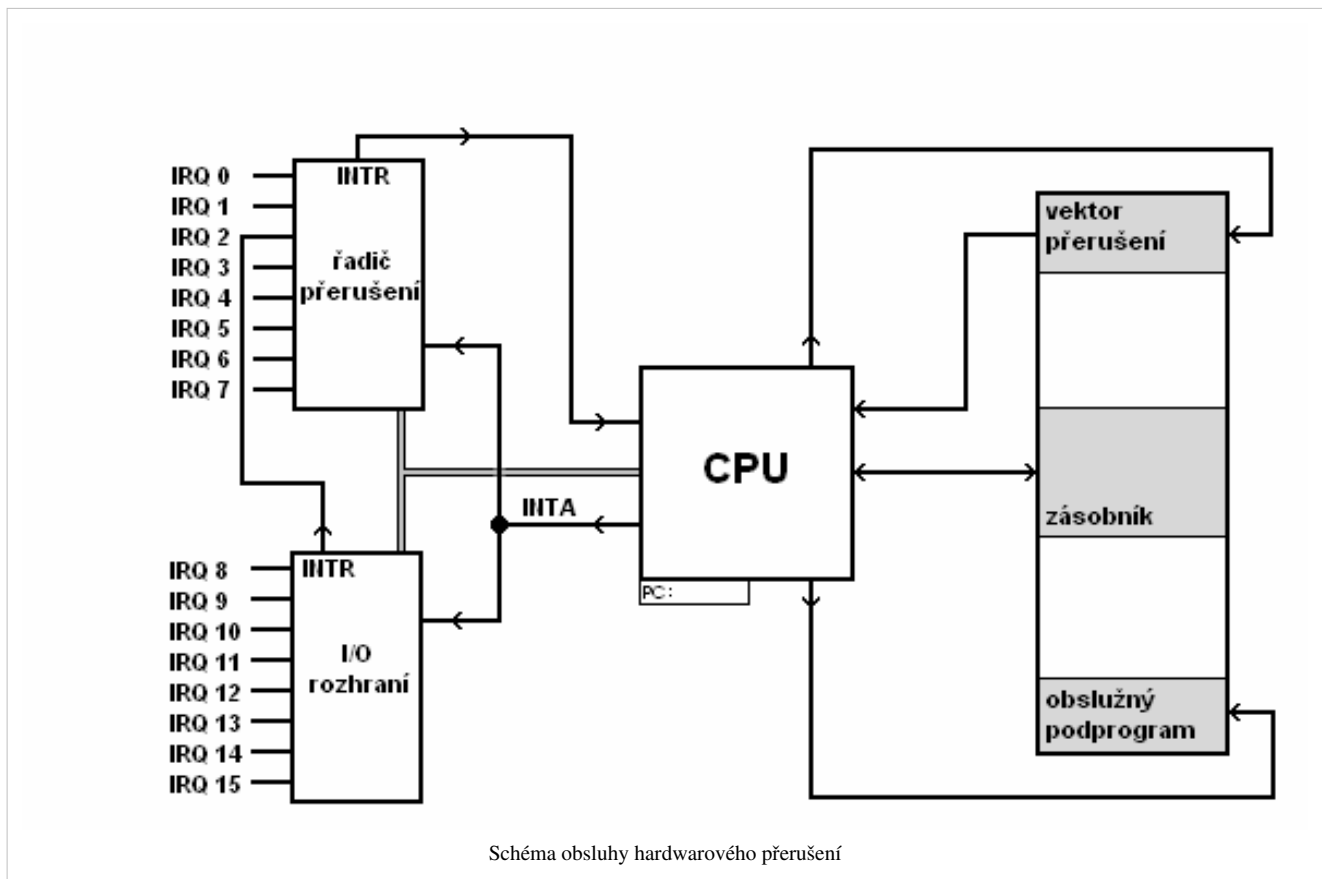
Softwarové přerušeni

Softwarové přerušeni je speciální strojová instrukce (obvykle je jich v procesoru k dispozici několik, procesory Intel mapují všechna přerušeni na softwarová přerušeni). Tento typ přerušeni je na rozdíl od druhých dvou typů synchronní, je tedy vyvoláno zcela záměrně umístěním příslušné strojové instrukce přímo do prováděného programu. Jedná se o podobný způsob, jako vyvolání klasického podprogramu (podprogramem je zde ISR uvnitř operačního systému), avšak procesor se může zachovat jinak. Instrukce softwarového přerušeni se proto využívá pro vyvolání služeb operačního systému z běžícího procesu (tzv. systémové volání). Uživatelská úloha tak sice nemůže skočit do prostoru jádra operačního systému, ale může k tomu využít softwarové přerušeni (kterých je omezené množství a vstupní body lze snadno kontrolovat). Při využití privilegovaného režimu může softwarové přerušeni aktivovat privilegovaný stav.

Popis implementace

Průběh hardwarového přerušeni

1. Vnější zařízení vyvolá požadavek o přerušeni
2. I/O rozhraní vyšle signál IRQ na řadič přerušeni (na port IRQ 2)
3. Řadič přerušeni vygeneruje signál INTR – „někdo“ žádá o přerušeni a vyšle ho k procesoru.
4. Procesor se na základě maskování rozhodne obsloužit přerušeni a signálem INTA se zeptá, jaké zařízení žádá o přerušeni.
5. Řadič přerušeni identifikuje zařízení, které žádá o přerušeni a odešle číslo typu přerušeni k procesoru
6. Procesor uloží stavové informace o právě zpracovávaném programu do zásobníku.
7. Podle čísla typu příchozího přerušeni nalezne ve vektoru přerušeni adresu příslušného obslužného podprogramu.
8. Vyhledá obslužný podprogram obsluhy přerušeni v paměti a vykoná ho.
9. Po provedení obslužného programu opět obnoví uložené stavové informace ze zásobníku a přerušeni program pokračuje dál.



Průběh softwarového přerušeni

1. Při příchodu přerušeni se uloží stavové informace o právě zpracovávaném programu do zásobníku.
2. Zakáže se další přerušeni.
3. Procesor zjistí vektor přerušeni (podle operandu)
4. Nalezne obslužný podprogram a vykoná ho.
5. Po návratu z podprogramu obnoví uložené stavové informace o přerušeni programu.

Softwarové přerušeni v architektuře x86

Architektura x86 má dvě instrukce pro softwarové přerušeni: `int` a `int03`, speciální krátkou formu určenou jako breakpoint pro debugery.

Má šest výjimek:

1. dělení nulou
2. krokovací přerušeni - na i386 rozšířeno na víceúčelové debugovací přerušeni
3. NMI
4. breakpoint - vyvolaný již zmíněnou instrukcí `int03`
5. přetečení - je vyvolané instrukcí `into` pokud je ve vlájkách zaznamenáno přetečení
6. překročení mezí - volané instrukcí `bound` pokud byly překročeny meze pole, přidané v 80186
7. chybná instrukce

U procesoru i386 přibýlo 9 dalších:

1. nedostupnost koprocessoru - nepřítomnost nebo nepřepnutá úloha - dřív bylo hlášeno externím hardwarovým přerušeni
2. dvojitý výpadek - volané pokud dojde k výjimce při vyvolávání výjimky

3. překročení limitu segmentu koprocosem
4. chybný TSS - chyba při pokusu o změnu úlohy
5. nepřístupný segment
6. překročení limitu zásobníku
7. obecná chyba ochrany - většina chyb související se segmentací kromě těch obslužených ostatními výjimkami
8. výpadek stránky - pokus o přístup ke stránce, která je namapována, avšak není v operační paměti přítomna
9. (nepoužito)
10. chyba koprocosem

Přerušeni u PC AT

U počítačů PC AT jsou standardně použita čísla vnějších přerušeni takto:

IRQ 0	Řadič systémového času
IRQ 1	Řadič klávesnice
IRQ 2	Připojení řadičů přerušeni z I/O rozhraní (kaskáda)- zde jsou sjednoceny IRQ 8 až IRQ 15
IRQ 3	Sériový port 2
IRQ 4	Sériový port 1
IRQ 5	Paralelní port 2
IRQ 6	Řadič disket
IRQ 8	Hodiny reálného času
IRQ 9	Volný, podle připojeného zařízení
IRQ 10	Volný, podle připojeného zařízení
IRQ 11	Volný, podle připojeného zařízení
IRQ 12	Volný, podle připojeného zařízení
IRQ 13	Numerický koprocosem
IRQ 14	Řadič pevného disku
IRQ 15	Volný, podle připojeného zařízení

Obsluha přerušení

Obsluha přerušení (zkratka **ISR**, anglicky *interrupt handler* nebo též *interrupt service routine*), je v informatice speciální procedura, která je v operačním systému vyvolána při obsluze přerušení. Přerušení je asynchronní událost, při jejímž příchodu je přerušena činnost procesoru, je vyvolána obsluha přerušení a poté činnost procesoru pokračuje na místě, kde bylo přerušení vyvoláno. Obsluha přerušení je obvykle součástí ovladačů zařízení, které se instalují do operačního systému kvůli obsluze připojených hardwarových zařízení (tzv. *vnější přerušení*), ale slouží též pro obsluhu přerušení vyvolaných uvnitř procesoru (tzv. *vnitřní přerušení*) a *softwarových přerušení* vyvolaných speciální strojovou instrukcí.

Obsluha přerušení

Přerušení je původně asynchronní mechanismus, pomocí kterého pomalá vstupně-výstupní zařízení získávala pozornost procesoru nezávisle na právě prováděné činnosti.[KOLÁŘ, 12] Když pásková jednotka nebo pevný disk dostane pomocí strojových I/O instrukcí za úkol přečíst určitá data, nemusí procesor (právě prováděný program) čekat, až pomalé zařízení příslušná data připraví pro vyzvednutí. Díky existenci přerušení může procesor pokračovat v jiné činnosti pomocí převedení procesu ze stavu běžící do stavu blokováný (viz článek Proces (program)). V okamžiku, kdy jsou data v zařízení připravena (typicky ve vyrovnávací paměti), vyvolá zařízení přerušení. Procesor dokončí právě prováděnou strojovou instrukci a pomocí tabulky přerušení je vyvolána obsluha přerušení (ISR) pro zařízení, které o přerušení požádalo.

Obsluha přerušení je obvykle součástí ovladače zařízení. Na svém začátku musí obsluha přerušení nejprve na vhodné místo v operační paměti uložit stav procesoru a jeho registry, aby přerušovaný proces po návratu nic nepoznal (aby nebyl narušen přerušovaný výpočet). Pro uložení těchto informací je možné použít speciální strojovou instrukci, avšak častěji je používána sada běžných instrukcí, protože hardwarová podpora je buď pomalá nebo neúplná. Po uložení stavu procesoru a registrů, které obsluha přerušení bude používat, dojde k vlastní obsluze zařízení, při které jsou z vyrovnávací paměti zařízení odebrána data (viz příklad výše) nebo naopak doplněny další instrukce a data (například u grafické karty).

Po provedení obsluhy zařízení je obnoven stav procesoru a procesor pokračuje ve vykonávání programu následující instrukcí v místě, kde došlo k přerušení. Přerušovaný program tak (až na zpoždění) nepozná, že byl přeruš.

Rozdělení obsluhy přerušení

Obsluha vstupně-výstupního zařízení je typicky časově kritická operace, která nemůže být přerušena (z důvodu časování komunikace pomocí I/O instrukcí), protože by příslušný hardware mohl ztratit data (nová data by ve vyrovnávací paměti přepsala starší, o které by se muselo znovu žádat) a podobně (některá zařízení vyžadují přesně časované souslednosti pokynů, mohlo by dojít k předání neúplných instrukcí atd). Proto je nutné během vykonávání obsluhy přerušení zakázat přerušování (vykonávání ISR není možné přerušit) a povolit ho až na konci provádění obsluhy přerušování. V takovém případě však dochází v operačním systému k velkým latencím (zpožděním), kdy počítač během dlouhé obsluhy přerušování není schopen vykonat jinou činnost (například obsluhu myši, překreslování obrazovky a podobně), což uživatel vnímá negativně (systém se „zasekává“).

Aby nedocházelo k dlouhým prodlevám při obsluze přerušování (a například nebyla ztracena data z kritických zařízení při obsluze méně významných zařízení), je obvykle v operačním systému rozdělena obsluha přerušování na dvě části (například v jádře linuxu).^[1] První část obsluhy přerušování (anglicky *First-Level Interrupt Handler (FLIH)*), označována též jako *hard interrupt handlers* nebo *fast interrupt handlers*) provede jen nejnnutnější úkoly, zaregistruje v systému činnosti, které je možné udělat později a skončí.

Druhá část obsluhy přerušování (anglicky *Second-Level Interrupt Handlers (SLIH)*, *slow/soft interrupt handlers*, v Linuxu *bottom half handlers* nebo *deferred procedure call*, tj. odložené volání procedury) je prováděna později, kdy

je možné ji provést v lépe naplánovaném čase. Odloženým způsobem se typicky zpracovávají data od/pro zařízení, které dodává větší objemy dat (síťová karta, pevný disk). V takovém případě jsou v první části obsluhy přerušeni nakopírována připravená data ze zařízení do operační paměti a druhá část obsluhy přerušeni se věnuje jejich vlastnímu zpracování, například doručení dat do paměťového prostoru cílového programu.

Rozdělením obsluhy přerušeni je možné výrazným způsobem snížit latenci systému a zvýšit jeho průchodnost. Využívají ji všechny moderní operační systémy (Microsoft Windows, Linux, atd.). Pokud by tato možnost nebyla využita, mohlo by u počítačů s vyšším I/O zatížením docházet ke ztrátám dat z důvodu jejich přepsání ve vyrovnávacích pamětech s omezenou velikostí, protože by systém trávil většinu času v nepřerušitelných dlouhých obsluhách přerušeni. Operační systém reálného času (RTOS) tuto techniku používá, aby splnil požadavky na nízkou latenci.

Polling

Obsluha přerušeni je typicky vyvolávána pomocí přerušeni. Je-li však v systému generován příliš vysoký počet přerušeni, snižuje se výrazným způsobem výkonnost systému, protože režie vyvolání obsluhy pomocí přerušeni je poměrně vysoká. V takovém případě je možné využít takzvaný *polling*, což znamená vyřazení přerušeni a naplánování vyvolávání obslužné rutiny (ISR) synchronním způsobem. Operační systém pak periodicky vyvolává ISR a ta zjistí stav zařízení a eventuálně provede jeho obsluhu. Některé ovladače zařízení obsahují podporu pollingu (například vysokorychlostní síťové karty) a přecházejí do tohoto režimu automaticky při dosažení určité míry I/O zátěže.

Vlákna přerušeni

Některé operační systémy (například Solaris, Mac OS X, WinCE a FreeBSD) používají odlišný způsob obsluhy přerušeni, který se označuje jako vlákna přerušeni (anglicky *interrupt threads*). Obsluha přerušeni je realizována jako samostatné vlákno (proces, kernel thread) s vysokou prioritou, které má povoleno přerušeni, ale co je důležitější, může být blokováno pomocí mutexů. Takové řešení značně zjednodušuje zamykání v jádře. Vlákno může také být přerušeno jiným vláknem s vyšší prioritou.

Literatura

- KOLÁŘ, Petr. *Operační systémy* [online]. Liberec: 2005-02-01, [cit. 2008-08-30]. S. 7. Dostupné online. ^[2]

Reference

[1] <http://www.tldp.org/LDP/tlk/kernel/kernel.html>

[2] <http://www.nti.tul.cz/~kolar/os/>

Související články

- Přerušeni – asynchronní událost v počítači, která vyvolává obsluhu přerušeni
- Kernel – jádro operačního systému
- Ovladač zařízení – anglicky *driver*

Externí odkazy

V tomto článku byl použit překlad textu z článku *interrupt handler* (http://en.wikipedia.org/wiki/interrupt_handler?oldid=250609489) na anglické Wikipedii.

Vektor přerušení

Vektor přerušení je adresa paměti handleru přerušení nebo index pole tabulky vektorů přerušení, která obsahuje paměťové adresy přerušení. Tabulka bývá většinou umístěna na začátku paměti, jak je běžné u procesorů architektury x86. Zvláštním přerušovacím vektorem je resetovací vektor (vyvolatelný signálem RESET), který pouze skočí na adresu F000:FFF0.

Jinými slovy se dá také **vektor přerušení** popsat jako, paměťový prostor obsahující první instrukci procesu, který má procesor při vyvolání přerušení provést.

IRQ

IRQ je zkratka z anglického *Interrupt ReQuest*. Označuje signál, kterým požádá zařízení procesor o pozornost, tedy požádá o přerušování probíhajícího procesu za účelem provedení důležitější akce.

IRQ se provádí následovně: Zařízení sdělí řadiči přerušování, že potřebuje provést přerušování. Řadič přerušování poté upozorní CPU, že jsou čekající přerušování. Když se CPU dostane do stavu kdy je ochotné přerušování přijmout, přerušování probíhající výpočet, zeptá se řadiče na nejdůležitější nezamaskované čekající přerušování a spustí jeho obsluhu. Ta typicky odloží právě probíhající proces, provede základní obsluhu zařízení, informuje řadič přerušování o dokončení obsluhy přerušování, obnoví stav procesu a pokračuje dále.

Přerušování často bývají identifikovány indexem s formátu zkratky IRQ následované číslem přerušování. Například pro programovatelné řadiče z rodiny Intel 8259 existuje 8 přerušování, které bývají běžně označovány jako IRQ0 až IRQ7. V počítačových systémech založených na x86 architektuře, které používají 2 z programovatelných řadičů jsou přerušování číslována od IRQ0 do IRQ15.

Některé procesory mají řadič přerušování integrovaný, ale častější je uvedený stav kdy je řadič logicky oddělen. Řadič obvykle vyhodnocuje priority přerušování, umožňuje některá přerušování ignorovat a pamatuje si, která přerušování čekají na vyřízení.

x86 IRQ

Pro seznam přerušování a diskuzi ohledně hardwarových IRQ pro systémy x86 se podívejte do článku Intel 8259.

Na systémech, které využívají Intel 8259 je typicky využito 16 IRQ. IRQ 0 až 7 jsou spravovány jedním Intel 8259 programovatelným řadičem přerušování a IRQ 8 až 16 druhým Intel 8259 PIC. První PIC, primární, tzv. master, je jediný, který přímo komunikuje s procesorem, kdežto druhý, tzv. slave, namísto procesoru signalizuje prvnímu pomocí přerušování IRQ 2, a master předává tento signál přímo procesoru. Existuje tedy 15 IRQ dostupných pro hardware.

Na novějších systémech s použitím tzv. Intel APIC Architecture je typicky k dispozici 24 IRQ a 8 dalších je využito pro předávání PCI přerušování tak, aby se vyhnulo konfliktu mezi dynamicky konfigurovanými PCI přerušováními a staticky konfigurovanými ISA přerušováními. Na dřívějších APIC systémech existovalo pouze 16 IRQ nebo pouze s Intel 8259 řadiči PCI přerušování byly předávány do 16 IRQ s použitím PIR obvodu integrovaného do southbridge.

Nejjednodušší způsob, jak zjistit tyto informace na Microsoft Windows je použít Správce zařízení nebo Systémové informace Microsoft. Na systémech Linux, IRQ mapování může být zobrazeno použitím příkazu `cat`

`/proc/interrupts` nebo `procinfo` programy.

Master PIC

- IRQ 0 — systémový časovač.
- IRQ 1 — přerušení klávesnice.
- IRQ 2 — předává signály z IRQs 8–15
 - každé zařízení konfigurované tak, aby používalo IRQ 2 ve skutečnosti používá přerušení IRQ 9
- IRQ 3 — sériový port controller pro COM2.
- IRQ 4 — řadič sériového portu COM1.
- IRQ 5 — LPT port 2 nebo zvuková karta
- IRQ 6 — řadič disketové mechaniky (viz disketa)
- IRQ 7 — LPT port 1 nebo zvuková karta.

Slave PIC

- IRQ 8 — hodiny reálného času
- IRQ 9 — *open interrupt / available* nebo SCSI host adapter
- IRQ 10 — *open interrupt / available* nebo SCSI nebo NIC
- IRQ 11 — *open interrupt / available* nebo SCSI nebo NIC
- IRQ 12 — myš na PS/2
- IRQ 13 — matematický koprocesor nebo integrovaná floating point unit nebo meziprocesorové přerušení.
- IRQ 14 — primární ATA kanál
- IRQ 15 — sekundární ATA kanál

ATA rozhraní obvykle obsluhuje hard disky a CD-ROM mechaniky

*V tomto článku byl použit překlad textu z článku *Interrupt_request* ^[1] na anglické Wikipedii.*

Reference

[1] http://en.wikipedia.org/wiki/Interrupt_request?oldid=389315500

Zdroje článků a přispěvatelé

Přerušení *Zdroj:* <https://cs.wikipedia.org/w/index.php?oldid=11491091> *Přispěvatelé:* Ben Skála, DaBler, Hkmaly, Koroner, Kvido, Milan Keršláger, Množina, Pec1, SroubekJirka, Zagothal, 5 anonymní úpravy

Obsluha přerušení *Zdroj:* <https://cs.wikipedia.org/w/index.php?oldid=11225895> *Přispěvatelé:* Hobr, Jvs, Martin.Hetver, Milan Keršláger, Zik, 3 anonymní úpravy

Vektor přerušení *Zdroj:* <https://cs.wikipedia.org/w/index.php?oldid=11341039> *Přispěvatelé:* Daewoo, Jvs, 1 anonymní úpravy

IRQ *Zdroj:* <https://cs.wikipedia.org/w/index.php?oldid=11081507> *Přispěvatelé:* AleyCZ, Hkmaly, Honzap, Jj14, LiMr, Martin Kozák, MigNov, MiroslavJosef, 4 anonymní úpravy

Zdroje obrázků, licence a přispěvatelé

Soubor:Obsluha_preruseni.gif Zdroj: https://cs.wikipedia.org/w/index.php?title=Soubor:Obsluha_preruseni.gif Licence: Public Domain Přispěvatel: SroubekJirka

Licence

Creative Commons Attribution-Share Alike 3.0
[//creativecommons.org/licenses/by-sa/3.0/](https://creativecommons.org/licenses/by-sa/3.0/)
