

Přerušení

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.

Proč přerušení v CPU ? – kvůli zvýšení paralelizace (aby molo více zařízení jet současně)

Proč zvýšení paralelizace? Kvůli šetření času, rozměr čipu, rozměr desky, jediná cesta ke zvýšení výkonu MCU)MIPS)

V čipu ALU je elektronické zařízení samostatně funkční

Např. v čítači v AVR assembleru si nastavíme OCR0 (overflow) a jdeme, dokud na TCNT0 nebude stejná hodnota, jakmile to dojde do tohoto bodu, přeteče to a TCNT0 jde zas od počáteční hodnoty.

Řadič přerušení - Ten určuje, v jakém pořadí budou přerušení obsloužena. Pořadí je určeno na základě *priority*, což je předem stanovené pořadí důležitosti. Jednotlivým modulům počítače, které mohou vyvolat přerušení, jsou přiřazeny rozdílné priority. Pokud procesor obsluhuje nějaké přerušení, běží na úrovni shodné s úrovní obsluhovaného přerušení. Činnost procesoru může být přerušena pouze přerušením s vyšší prioritou

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.

Vnitřní přerušení

Vnitřní přerušení 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 přerušení

Softwarové přerušení je speciální strojová instrukce. Tento typ přerušení 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 přerušení 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 přerušení (kterých je omezené množství a vstupní body lze snadno kontrolovat). Při využití privilegovaného režimu může softwarové přerušení aktivovat privilegovaný stav.

Maskování-

Proč se používá? Kvůli uložení z 2 proměnných do jednoho registru a možnosti přečtení právě pomocí maskování

Toto je avšak možné pouze tehdy, pokud obě proměnné nabývají hodnot menších než 4-bity

r16



Postup:

1) ulož a, a je 5

Ori r16,a

2) ulož b; b je 7

Swap r16

Ori r16,b

Swap r16

3) jak přečíst a z r16

MOV r17,r16

Andi r17,0b00001111 ; maskování

V r17 je a

4) jak přečíst r16 b

MOV r17,r16

And r17;0b11110000 ; maskování

Swap r17; b v r17

Priorita- dá se definovat různě (pole potřeby)

- 1) podle pořadí v tabulce vektorů přerušení
reset je na adrese 0 -> nejvyšší priorita
- 2) definovat podle potřeby systému (O. S)
např. (když si dám video do fullscreenu ,tak prioritu má větší ESC)

Jak to funguje: a) přijde požadavek o obsluhu ze zařízení

b) O. S vytvoří nový proces, s předem definovanou prioritou a uloží jej na čekací frontu

c) O.S (dispatcher) pak se třídí tyto procesy podle priority

řadič přerušení je v podstatě dispatcher – obsahuje minimálně tabulku přerušení, nějaké obsluhy a práce se zásobníkem

přerušení

jednoúrovňové-vykoná obsluhu hned

Dvojúrovňové: vytvoří proces který má zdroj obsluhy a dál ten proces jde potom na čekací frontu , a tam si to O.P setřídí podle priority ..Lépe vypsane :

- 1) vzniká žádost o přerušení
- 2) operační systém vytvoří proces, kde zdroj procesu obsahuje obsluhu přerušení ,pak tento proces uloží na čekací frontu
- 3) O.S. pak setřídí čekací frontu podle priority a udělá kontext podle potřeby

Proč ? důvod- nechceme ztratit /vzniknout žádost o přerušení

Pokud se vykonává obsluha je dlouhá, vzniká tím riziko vinutí

Takže uděláme proces, který uložíme na frontu. A vykonáme, až přijde vhodná chvíle

Kostra - Průběh přerušení

- 1) Požadavek na přerušení
- 2) Nesloučit návratovou adresu na zásobník (uložit abych se mohl vrátit tam, kde jsem skončil)
- 3) Skočit do tabulky vektorů přerušení na odpovídající lokace
- 4) V této lokaci je adresa obsluhy přerušení (ISR)
- 5) Skočit do obsluhy
- 6) Vykonání obsluhy
- 7) Obrátit návratovou adresu ze zásobníku
- 8) Vrátit se do místa, kde to bylo přerušeno

Vřeteno - SW koncept

v operačním systému existují způsoby, jak spustit programy tyto způsoby se dají klasifikovat následovně:

- 1) proces opačného systému
- 2) uživatelský proces

3) vřeteno

proces: existují následující dělení:

- 1) standartní
- 2) lehký proces

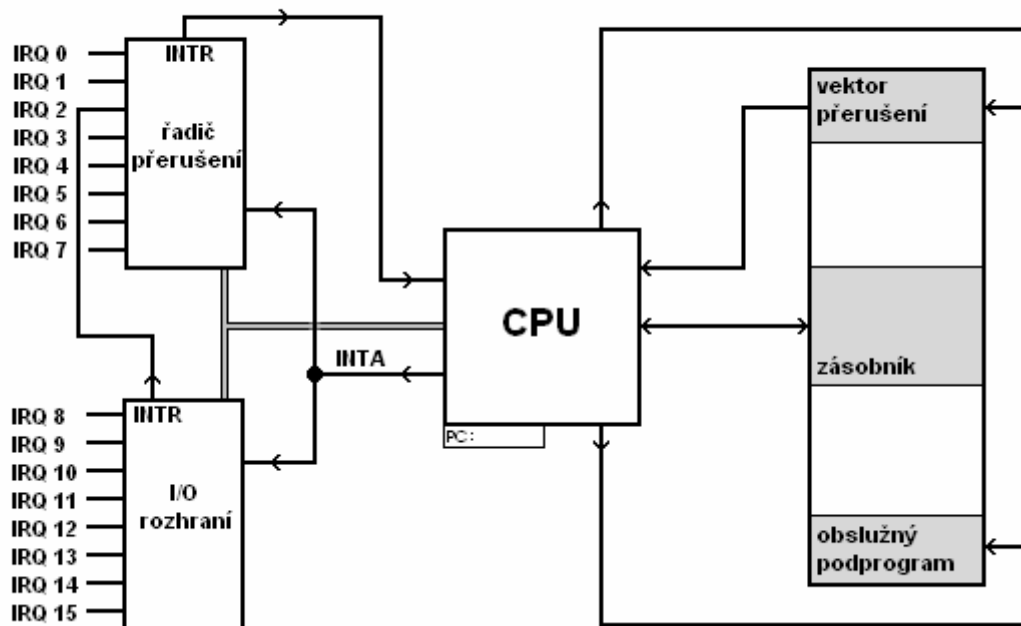
Proč standart a lehký ? protože alokace (vytvoření procesu zabere čas a zabere pamět) -> žere zdroje

Vřeteno je lehčí než „light weight“

- ➔ v podstatě jen nekonečná smyčka, která je načasovaná vřeteno
- identifikace
 - časová značka
 - vložit program do smyčky

Průběh hardwarového přerušení

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



Průběh softwarového přerušení

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

Přepnutí-

O. S. context switch – přepnutí mezi uživateli (procesy)

- přerušení (přepnutí) → podle žádosti události → RESET zdroje –
- SLEEP (power saving)
 - BOD (brown out detection)
 - manulání
 - WDT (watch dog)
 - START UP

Hashing – mapovací funkce, k jednomu vstupu přiřadí jeden výstup

Hash je program

Context switch

je operace, při níž dochází k přepnutí kontextu při plánování procesů, kdy je nutné ukončit běh jednoho procesu a vybrat jiný, kterému bude procesor následně přidělen. V preemptivních multitaskingových operačních systémech poskytuje plánovač každé úloze procesor na určitou dobu, která se nazývá time slice

time slice je tedy doba, za které je proces nechán běžet bez přerušení v preemptivní multitasking operačního systému