

# Přerušeni – definice a postup činnosti počítače při přerušeni, bloková zařízení, DMA a jeho činnost

## Přerušeni

Přerušeni je schopnost procesoru přerušit právě vykonávanou činnost (program) a začít vykonávat jiný úkon (program). Jinak řečeno jde o stav, kdy něco žádá obsluhu (obvykle periferie počítače) nebo alespoň pozornost.

Každé zařízení má při inicializaci nastavené číslo přerušeni, kterému náleží určitý vodič PCI sběrnice nebo hodnota USB signálu či PCIe. Tento signál je vždy zaslán do *řadiče přerušeni*, který rozhoduje podle priority požadavků a s ohledem na probíhající proces, zda tento požadavek předá procesoru ke zpracování. Proces je současně informován, zda se jedná o maskovatelné nebo nemaskovatelné přerušeni.

### Dělení přerušeni – podle původu

- a) **vnější přerušeni** – asynchronní přerušeni (nečekané, neplánované); přichází z I/O zařízení (např. stiskl klávesy, tlačítka z myši apod.)
- b) **vnitřní přerušeni** – vyvolává ho sám procesor; také asynchronní (nečekané, neplánované); procesor tím dává najevo, že má nějaké problémy se zpracováním strojových instrukcí (výpadek stránky, porucha paměti, dělení nulou, jiná chyba, jež by se neměla vyskytnout)
- c) **softwarové přerušeni** – očekávané, vyplývá z podstaty zpracovávaného programu – např. nějaké systémové volání

### Dělení přerušeni – podle maskovatelnosti

- a) **maskovatelné** – přerušeni pro potřeby ovladačů, např. přenos po síti
- b) **nemaskovatelné** – systémové; fatální chyba (výpadek zdrojů apod.); přerušeni, které se musí provést a není možné jej zamaskovat

### Postup činnosti počítače při přerušeni

1. přerušeni je speciálním čipem přijato (řadičem přerušeni, interrupt controllerem)
  - řadič vyhodnotí a rozhodne, zda přerušeni nastane nebo nenastane, případně kdy nastane
  - je-li rozhodnuto, že nastane, je o požadavku na přerušeni vyrozuměn procesor – ten dokončí právě prováděnou instrukci, uloží operaci do stacku (uloží  $IP+CS^1$  – adresu aktuální instrukce, která by se dále prováděla, kdyby přerušeni nenastalo)
2. procesor si z řadiče přerušeni převezme *číslo přerušeni* – každé přerušeni je spojeno s určitým číslem přerušeni, které popisuje určitou službu/hardwareovou funkci, která je požadována
3. procesor hledá v TVP (*tabulka vektorů přerušeni*) (TVP je úsek operační paměti) tzv. *vektor přerušeni* (adresu první instrukce obsluhy přerušeni – ke každému přerušeni náleží jeden

vektor přerušení – když procesor zjistí číslo přerušení, může zjistit vektor přerušení a vypočítat adresu –  $IP+CS^1$ , zapsat a pokračovat v přerušení)

4. po vykonání obsluhy procesor obnoví ze stacku obsahy všech registrů a nakonec i  $IP+CS^1$  – tím je obsluha přerušení ukončena a pokračuje se ve vykonávání původního procesu

## Bloková (a znaková) zařízení

- a) **bloková zařízení** – komunikace s nimi probíhá po blocích (systémem definovaný počet B, zpravidla 1 kB); chceme-li přečíst nebo zapsat (byť jen jediný bajt) musíme přečíst nebo zapsat celý blok<sup>2</sup>; typickým blokovým zařízením je disk
- b) **znaková zařízení** (bytová) – komunikace s nimi probíhá po znacích (bajtech); každým zápisem i čtením se přenáší jeden znak; typickým blokovým zařízením je např. klávesnice

## DMA (Direct Memory Access)

Přímý přístup do paměti blokového zařízení bez účasti procesoru. Protože data neprocházejí skrze procesor, I/O zařízení komunikují s operační pamětí rychleji – je dosaženo vyššího výkonu.

DMA je zajišťováno hardwarovým řadičem – procesorem. Komunikace řadiče s I/O zařízením se nazývá *kanál*. Kanálů je 8, jsou 8/16bitové, některé jsou k volnému užití, jiné určené pro daná zařízení (DRAM refresh, HDD, CD-ROM, ...). Dvě zařízení nesmějí používat stejný DMA kanál.

### Průběh přenosu

1. před přenosem musí procesor v DMA nastavit číslo kanálu, adresy, délku bloku dat a směr přenosu; toto naprogramování DMA trvá cca mikrosekundy (na rozdíl od HDD, kde trvá vyhledávání milisekundy)
2. po nastavení je povolena činnost DMA, řadič je však spuštěn až po *přerušení* od pomalejší periferie udávající připravenost zařízení k přenosu
3. DMA oznámí procesoru signálem *HOLD*, že bude provádět přenos a čeká na potvrzení od procesoru
4. procesor ukončí přenosy na sběrnících, odpojí se a odpoví DMA signálem *HOLDA* (hold accepted), čímž umožní DMA řízení přenosu, a přechází do stavu *WAIT* (čekání) – v této době může procesor dále provádět výpočty, ale jen do doby, než opět potřebuje sběrnici
5. DMA po ukončení komunikace předá řízení sběrnici zpět procesoru
6. procesor zjistí, zda přenos skončil v pořádku nebo s chybou a pokračuje ve své práci

Procesor by celý proces vyřídil softwarově (potřeboval by instrukce), DMA vše řídí hardwarově (vše se řídí čítačem – v jednom čítači jsou nastaveny adresy na adresové sběrnici, ve druhém čítači je počet nepřenesených bytů – přenáší se tak dlouho, až je v čítači 0). Vše je řízeno pulzem bez instrukcí.

Funkci DMA někdy přebírají specializované procesory pro I/O nazývané PIO. Do příchodu DMA šlo o jediný způsob komunikace na tomto rozhraní.

---

<sup>1</sup>  $IP+CS$  = instruction pointer + code segment

<sup>2</sup> na rozdíl od bytových zařízení se využívá buffer – např. při čtení se blok načte do bufferu a čtení jednotlivých znaků pak již probíhá pouze v rámci bufferu – k zařízení tak není potřeba již přistupovat

## **Ultra DMA**

Rychlost DMA dosahuje až 100 MB/s. Pro každý blok se navíc používá kontrolní CRC (cyclical redundancy checking), a to pro každý přenášený blok.