

HARDVÉR POČÍTAČOV



Štruktúra počítačového systému

2

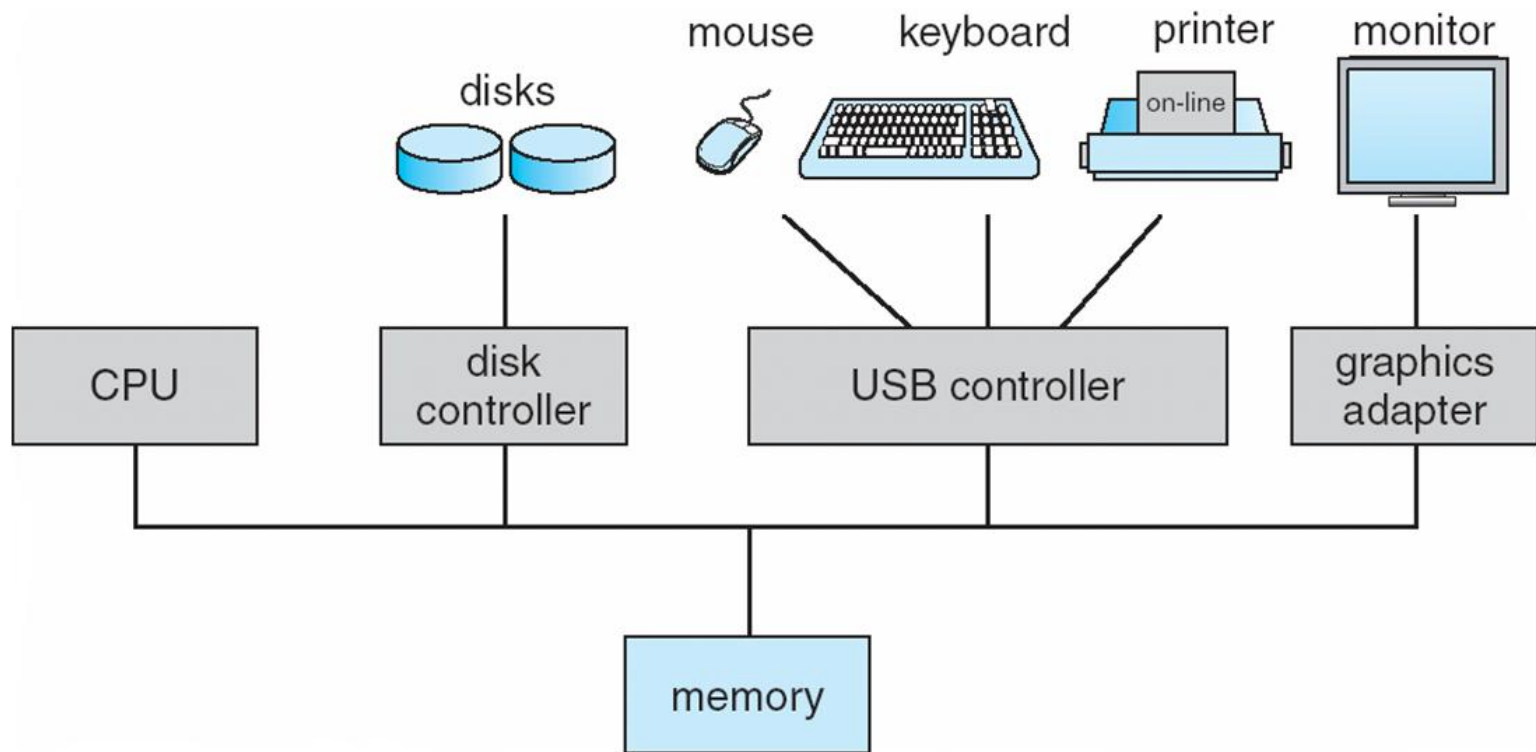
- Časti počítačového systému, ktoré spolupracujú s OS sú:
 - ▣ I/O systém
 - ▣ Pamäť – všetky typy
 - ▣ Periférne úložné zariadenia
 - ▣ Ochrana jednotlivých komponentov

Základné časti číslicového počítača

- **Procesor** - riadi činnosť počítača a vykonáva jeho hlavnú činnosť – spracovanie údajov. V súčasnosti je funkcia procesora často integrovaná do jedného integrovaného obvodu nazývaného mikroprocesor.
- **Operačná pamäť** - umožňuje ukladanie údajov a inštrukcií pre procesor
- **Vstupno/výstupný systém** - slúži pre komunikáciu s ostatnými zariadeniami počítača (vonkajšia pamäť, konzola, ...)
- **Systémová zbernica** - umožňuje komunikáciu medzi procesorom, operačnou pamäťou a modulom vstupu a výstupu

Zbernicová organizácia

4



Základné pojmy

5

□ Reprezentácia dát

Pomenovanie	Počet bitov (N)	Kardinalita (2^N)
Bajt (byte)	8	256
Slovo (word)	16	65536
Dvojslovo (double word)	32	4294967296
Štvorslovo (quad word)	64	2^{64}

Operačná pamäť

6

Operačnú pamäť reprezentuje údajová štruktúra typu pole.

Môžeme ju charakterizovať dvomi údajmi

- **šírka pamäte** udáva **počet bitov (N)** jednotlivých prvkov poľa,
 - **veľkosť pamäte** udáva jej **kapacitu** 2^N , teda koľko prvkov je schopná súčasne uchovávať.
- Jednotlivé prvky sú očíslované od 0 po $2^N - 1$.
- Pri prístupe k prvkom pamäte - **adresa**.

Processor

7

- Vyberá inštrukcie z pamäte a vykonáva ich.
ALU + riadiaca jednotka
- Registre procesora
 - ▣ **Riadiace a stavové registre**
 - **PC** (*program counter*) - počítadlo inštrukcií.
 - **Register inštrukcií IR** (*instruction register*), obsahuje operačný kód inštrukcie, ktorá sa bude práve vykonávať.
 - **Register príznakov** (*flag register*).

▣ Všeobecne prístupné registre

■ Dátové registre - akumulátor (AC)

■ Adresové registre

- **Indexový register** (*index register*) - pre adresovanie úsekov pamäte vzhľadom na nejaký počiatok
- **Ukazovateľ zásobníka** (**stack pointer**) určuje adresu vrcholu používateľského zásobníka. Menia ho inštrukcie *push* a *pop*. Je možné ho modifikovať aj priamo.
- **Segmentový register** (*segment pointer*) - pri segmentovaní. Uchováva adresu počiatku segmentu.

Registre, adresy, inštrukcie

9

▣ Registre

- 8 - 64 registrov; 4-8 bajtov

- Príklad - Registre architektúry x86

Obecné registre - AX, BX, CX, DX – 16 bitové

Obecné registre - SP, BP, SI, DI -32 bitové

Segmentové registre — CS, DS, ES, FS, GS, SS – 16 bitové

Špeciálne registre — IP a FLAGS

▣ Adresy

- 2-4 bytové adresy pamäťových miest

▣ Inštrukcie

- príklad kódu

add eax, 10 — *EAX ← EAX + 10*

add BYTE PTR [var], 10 — *add 10 to the single byte stored at memory address var*

and eax, 0fH — *clear all but the last 4 bits of EAX.*

xor edx, edx — *set the contents of EDX to zero.*

Výkonávanie inštrukcií

10

Príklad: jednoduchý 8 bitový procesor, 64 KB pamäte

Ku hodnote AC (akumulátora) pripočítame konštantu a podľa výsledku vykonáme podmienený skok

(PC- adresa inštrukcie, IR – register inštrukcií, „zero“ – register príznakov)

□ 1 krok - začiatok programu od adresy 100

Operačná pamäť		Registre procesora			
Adresa	Hodnota		Register	Hodnota pred	Hodnota po
100	64		PC	100	101
101	5		IR	xx	64
102	192		AC	79	79
103	0		„zero“	1	1
104	1				
105	78				

Vykonávanie inštrukcií pokr.

11

□ 2 krok

Dekódovanie inštrukcie - **operand** je na adrese z PC

Operačná pamäť			Registre procesora		
Adresa	Hodnota		Register	Hodnota pred	Hodnota po
100	64		PC	101	102
101	5		IR	64	64
102	192		AC	79	79
103	0		„zero“	1	1
104	1				
105	78				

Vykonávanie inštrukcií pokr.

12

□ 3 krok

Vykonanie inštrukcie – **výsledok** do AC, nastavenie **registra príznakov**

Operačná pamäť		Registre procesora			
Adresa	Hodnota		Register	Hodnota pred	Hodnota po
100	64		PC	102	102
101	5		IR	64	64
102	192		AC	79	84
103	0		„zero“	1	0
104	1				
105	78				

Vykonávanie inštrukcií

13

- 4 krok - druhý inštrukčný cyklus - podmienený skok podľa registra príznakov

Operačná pamäť			Registre procesora		
Adresa	Hodnota		Register	Hodnota pred	Hodnota po
100	64		PC	102	256
101	5		IR	192	192
102	192		AC	84	84
103	0		„zero“	0	0
104	1				
105	78				

Vykonávanie inštrukcií

14

□ 5 krok – podmienený skok

Operačná pamäť			Registre procesora		
Adresa	Hodnota		Register	Hodnota pred	Hodnota po
100	64		PC	102	256
101	5		IR	192	192
102	192		AC	84	84
103	0		„zero“	0	0
104	1				
105	78				

$$\text{Adresa} = 256 * \text{vyšší_bajt} + \text{nižší_bajt} = 256 * 1 + 0$$

Viacjadrové (multi-core) procesory

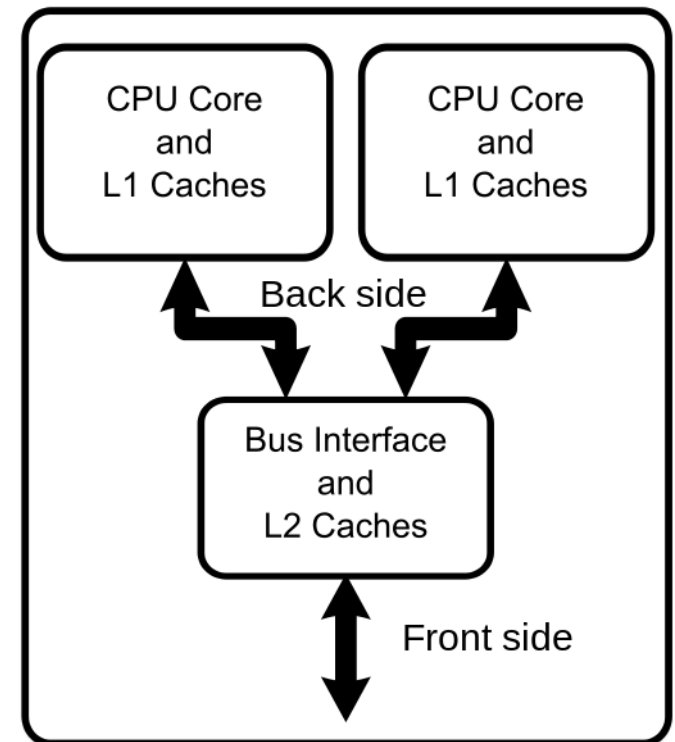
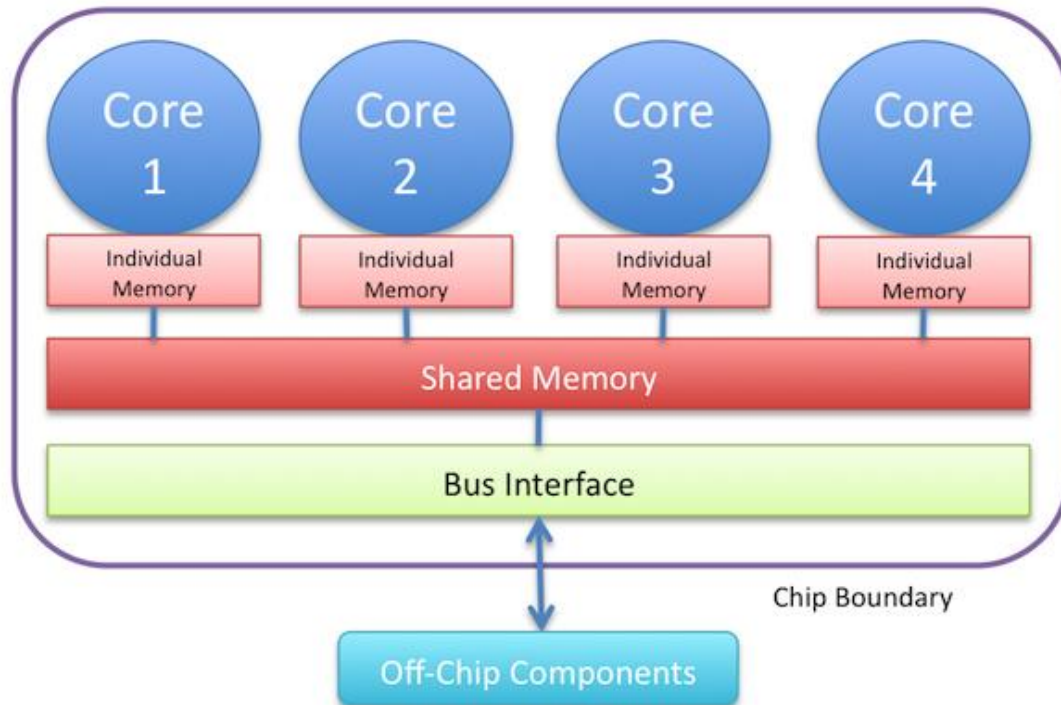
15

- Pozostávajú z 2 alebo viac nezávislých jadier
- Nachádzajú sa obyčajne na jednom integrovanom obvode
- Dvojjadrový, štvorjadrový, viacjadrový procesor
- Jadra môžu byť prepojené **tesne** alebo **vol'ne**. Napr. jadra môžu ale nemusia **zdieľať cache pamäte** a môžu implementovať **zasielanie správ** alebo zdieľanú pamäť medzi jadrami.
- Schémy prepojenia jadier: zbernica, kruh, 2 rozmerná mriežka (mesh) alebo súradnicové prepojenie
- V **homogénnych** viacjadrových systémoch všetky jadra **sú homogénne**
- V **heterogénnych** - **nie sú homogénne**

Viacjadrové (multi-core) procesory

16

Multi-core Processor



Amdahalov zákon

17

Vyjadruje aké maximálne zrýchlenie môžeme očakávať ak časť $(1-P)$ z programu sa dá vykonať len sériovo a zvyšná časť programu (P) - paralelne

Zrýchlenie **S** pri použití **N** procesorov,
za predpokladu, že časť **P** z programu je možné
vykonať **paralelne**

$$S(N) = \frac{1}{(1 - P) + \frac{P}{N}}$$

Viacjadrové (multi-core) procesory

18

□ Výhody

- Blízkosť viacerých CPU jadier na jednom chipe dovoľuje výmenu signálov s vyššou rýchlosťou a s vyššou kvalitou, ako v opačnom prípade .
- Najväčšie zrýchlenie – procesy, ktoré intenzívne využívajú CPU.
- Potrebujú menej energie ako rovnaký počet vzájomne prepojených procesorov
- Zdieľajú obvody ako napr. L2 cache a interface k FSB (front side bus).

Viacjadrové (multi-core) procesory

19

Nevýhody

- Potrebná dodatočná podpora zo strany OS pre maximalizáciu využitia výpočtovej kapacity.
- **Závislosť výkonu na využití vlákien v aplikáciách.**
- Ťažšie sa chladia kvôli vysokej hustote.
- Výkon procesora nie je jediným obmedzením pre výkon systému:
 - ▣ šírka pásma pri spojení s pamäťou cez systémovú zbernicu je ďalším obmedzením

Vstupno/výstupné operácie

20

- Vstupno/výstupné operácie zabezpečuje V/V systém
- Úlohou V/V systému je prenášať informácie medzi CPU alebo OP a periférnymi zariadeniami
- V/V systém pozostáva z :
 - ▣ V/V zariadení (periférie)
 - ▣ Radičov V/V zariadení
 - ▣ Softvéru - ovládače (drivery)

Vstupno/výstupné operácie pokr.

21

- Návrh V/V systému – základné problémy
 - ▣ CPU a V/V nemožno synchronizovať, dá sa len koordinovať
 - ▣ CPU je obyčajne omnoho rýchlejšie ako V/V zariadenie – V/V komunikujú s CPU asynchrónne
 - ▣ CPU – binárne kódovanie informácie,
V/V – spolupracuje s človekom – treba kódovať a dekódovať

Prerušenie

22

Zdrojom prerušenia môže byť:

- **Procesor** – napr. pri pretečení aritmetickej operácie, delení nulou a pod.
- **Časovač** – pri zabezpečení vykonávanie určenej činnosti v presne stanovenom intervale.
- **Periférne zariadenia** – oznamujú procesoru ukončenie vstupno-výstupnej operácie, resp. žiadosť o obsluhu.
- **Hardvér počítača** – napr. pri chybe parity pamäte, výpadku napájania a pod.

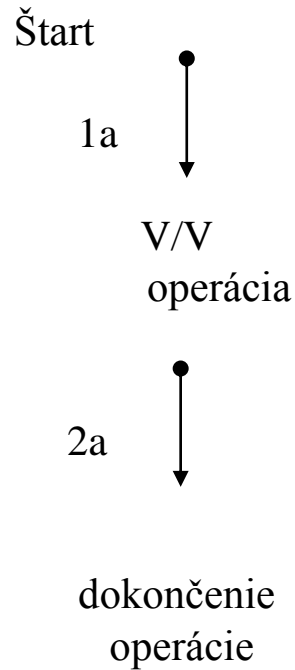
Obsluha prerušení

23

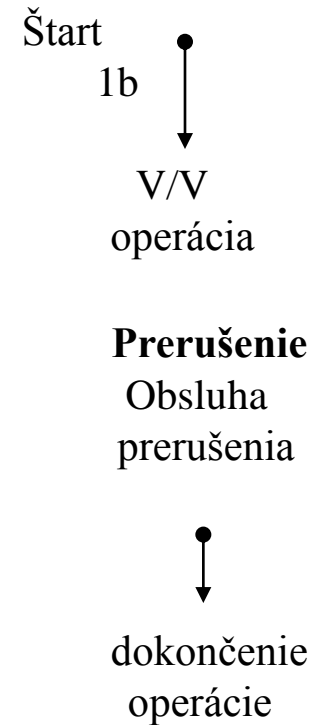
- Ked' sa vyskytne prerušenie, CPU
 - uchováva stav CPU
 - registre a čítač inštrukcie (PC)
 - vykonáva kód na adrese, na ktorej nastalo prerušenie
 - závisí od zdroja prerušenia
 - idea je nasledovná:
 - tento kód obslúži prerušenie
 - potom prerušený program môže pokračovať

Rutina pre obsluhu komunikačnej linky

24



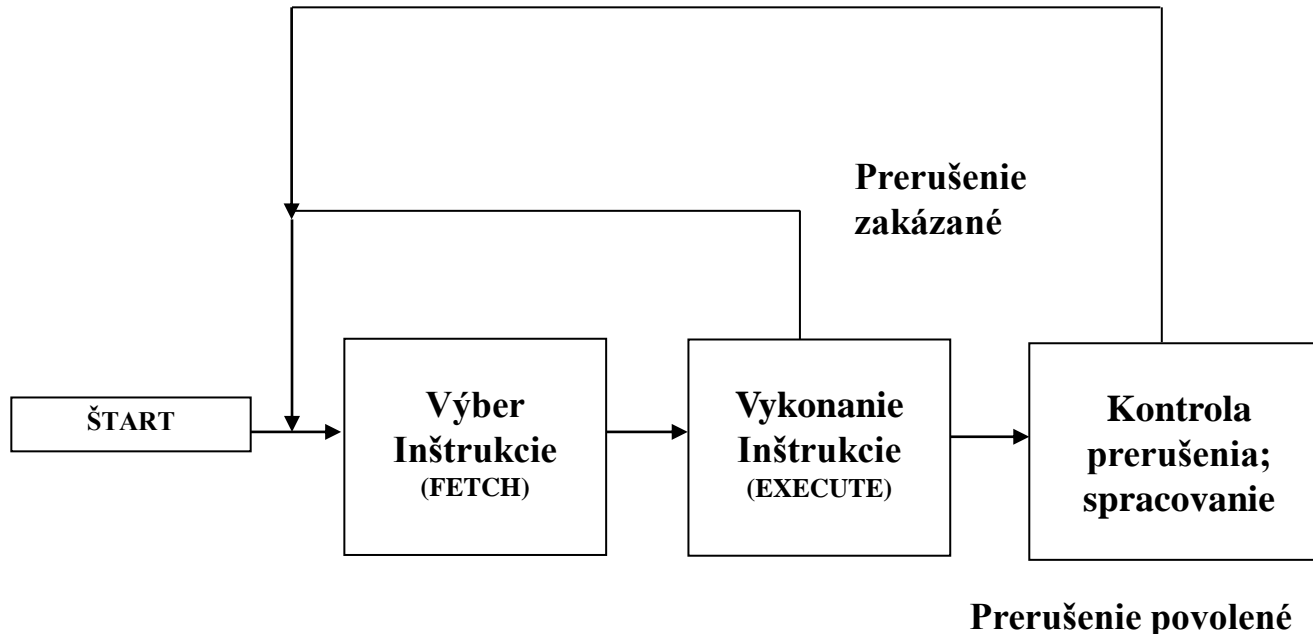
(a) bez použitia prerušenia
- pooling



(b) s použitím prerušenia

Vykonanie inštrukcie a prerušenie

25

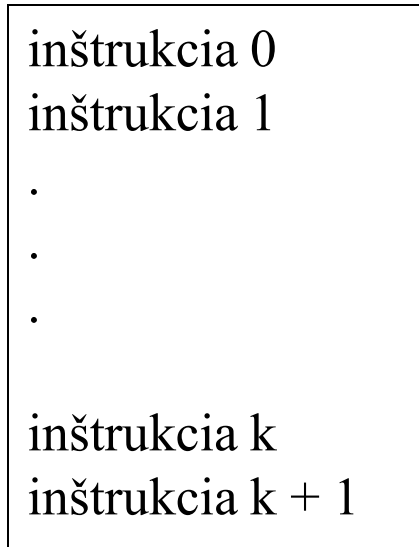


- Procesor skontroluje či nastalo prerušenie
- Ak nenastalo, zavedie ďalšiu inštrukciu
- Ak nastalo, zastaví vykonávanie bežiaceho programu a vykoná obslužný program prerušenia

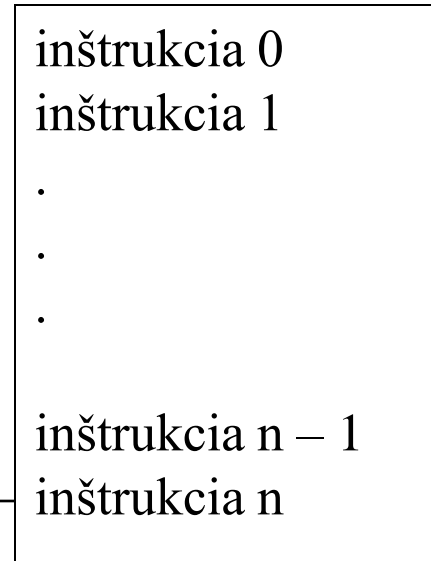
Priebeh riadenia programu v prípade vzniku požiadavky na prerušenie

26

Používateľský program



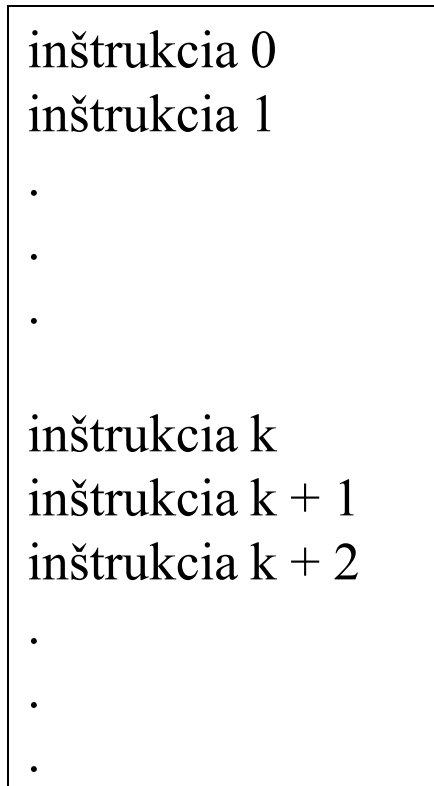
Obsluha prerušenia



Viacnásobné prerušenie

27

Používateľský program



inštrukcia 0
inštrukcia 1

·
·
·

inštrukcia n – 1
inštrukcia n

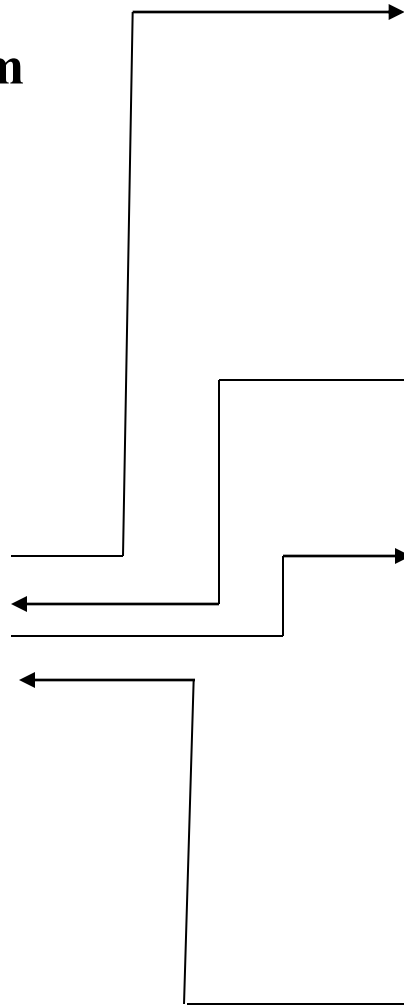
**Obsluha
prerušenia 1**

inštrukcia 0
inštrukcia 1

·
·
·

inštrukcia m – 1
inštrukcia m

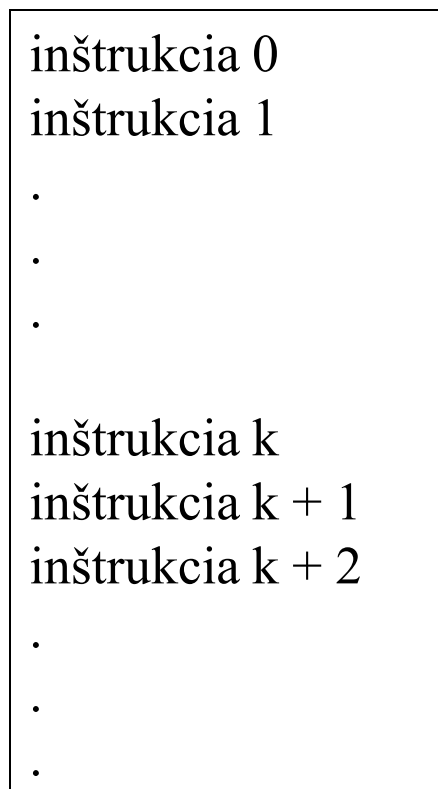
**Obsluha
prerušenia 2**



Vzniku viacerých požiadaviek na prerušenie, spracovanie podľa priorít

28

Používateľský program

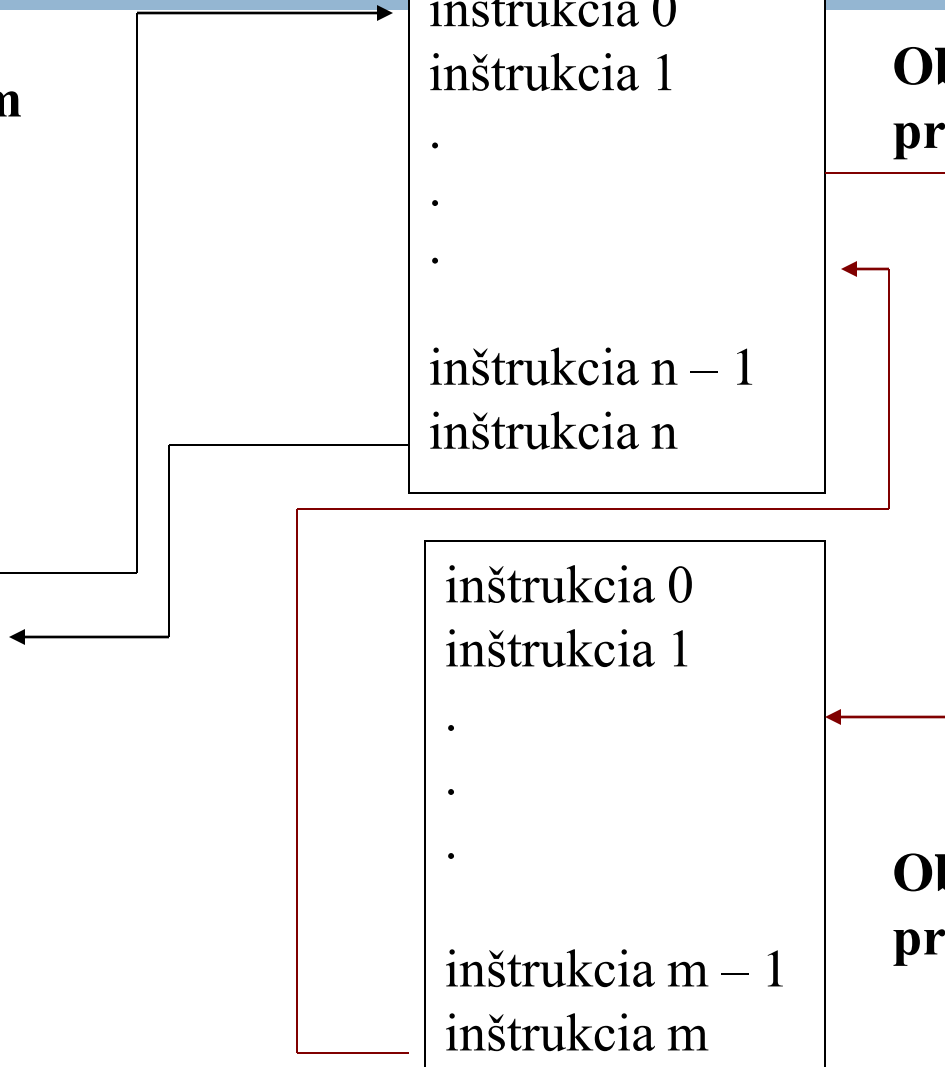


inštrukcia 0
inštrukcia 1
.
.
.
inštrukcia n - 1
inštrukcia n

Obsluha prerušenia 1

inštrukcia 0
inštrukcia 1
.
.
.
inštrukcia m - 1
inštrukcia m

Obsluha prerušenia 2



Vstupno-výstupné operácie

29

Procesor a periférne zariadenie si vymieňajú informácie rôzneho charakteru:

- **riadiace informácie**, napr. požiadavka previnúť magnetickú pásku na začiatok, určenie smeru nasledujúcej V/V operácie, a pod.,
- **stavové informácie**, napr. stav tlačiarne (pripravená, obsadená, chyba),
- **dáta**, s ktorými sa má V/V operácia uskutočniť

Riadenie vstupno-výstupných operácií

30

Programovo riadený vstup a výstup

Neustála kontrola stavového registra periférneho zariadenia

V/V riadený prerušením

prerušenie umožňuje procesoru sa venovať inej práci kým nastane prerušenie – efektívnejšie

Priamy prístup do pamäte (DMA)

Priamy prístup do pamäte - DMA

31

- V/V riadený prerušením nepostačuje na obsluhu periférií, ktoré prenášajú veľké objemy dát medzi pamäťou a periférnymi zariadeniami
- Využíva sa špecializovaný obvod pre priamy prístup k pamäti - direct memory access (DMA), je riadený pomocou radiča DMA
- Podstata DMA
 - ▣ CPU inicializuje DMA kanál, potom prenos riadi radič bez účasti CPU
 - ▣ Tým sa radovo zvýši prenosová rýchlosť
 - ▣ Prenáša sa veľký blok údajov v jednej súvislej operácii
 - ▣ Počas DMA prenosu môže dôjsť ku kolízii (radič DMA a CPU môžu súčasne potrebovať zbernicu, alebo pristupovať k pamäti)

DMA operácie

32

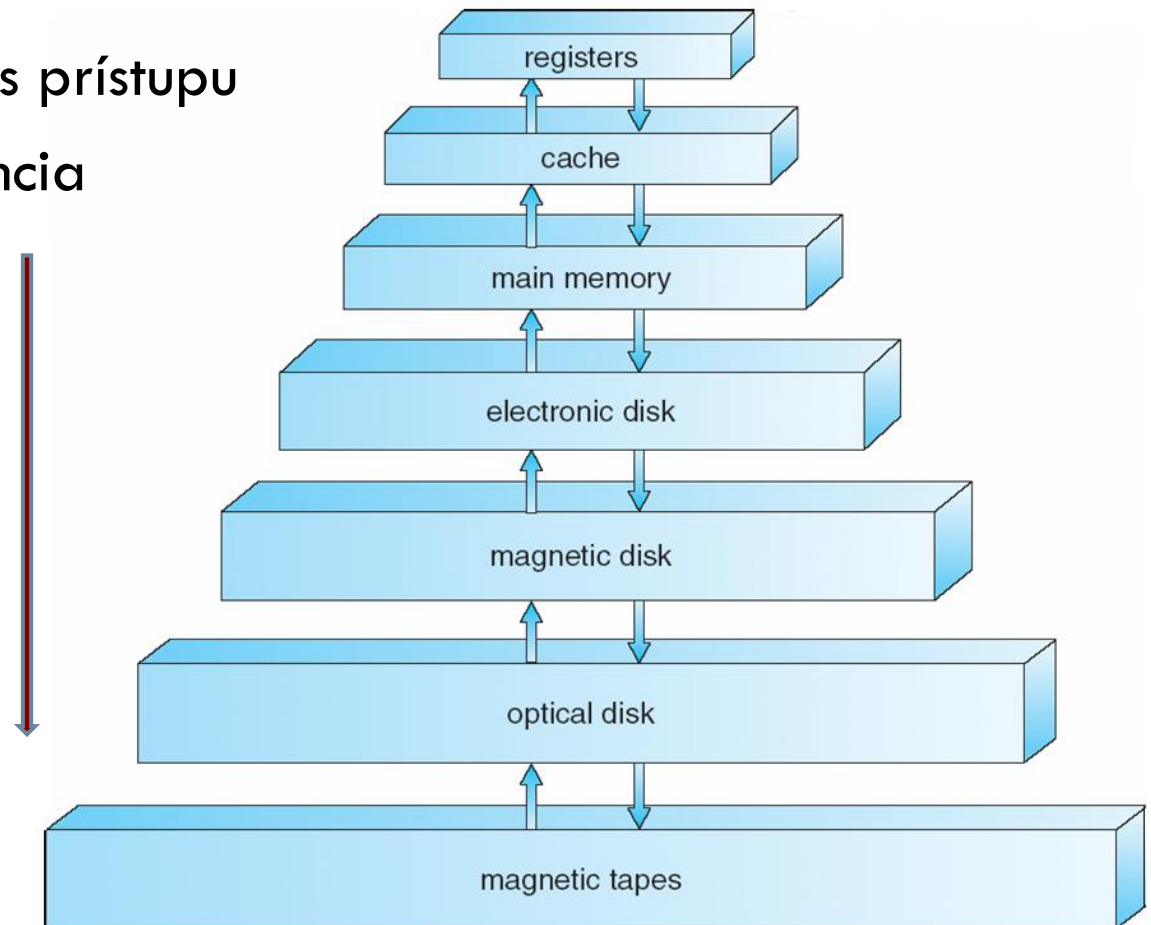
- DMA radič
 - ▣ pristupuje k periférnemu zariadeniu cez jeho radič
 - ▣ číta/zapisuje celé bloky dát
 - ▣ číta/zapisuje ich do/z pamäte
- Prerušená sú
 - ▣ na úrovni blokov dát
 - ▣ menej časté a preto
 - CPU môže urobiť veľa práce medzi prerušeniami.

Hierarchia pamäti

33

Smerom dolu

- rastie kapacita a čas prístupu
- klesá cena a frekvencia prístupu



Cache pamäť

34

- **Malá rýchla pamäť**, obsahujúca posledne použité dáta,
 - ▣ najčastejšie medzi procesorom a pamäťou, ale tiež používaná pre lokálne kópie sieťových dát atď.
 - ▣ pokúša sa zvýšiť výkon tým, že všetky informácie sú prítomné v pamäti
 - ▣ pri pokuse o čítaní z hlavnej pamäte sa najskôr skontroluje či daný údaj nie je v cache-pamäti
- Mechanizmus rozhodovania pre pohyb informácie medzi vrstvami je veľmi podstatný a riadi ho nejaký **nahrádzovací algoritmus**

Operačná pamäť a I/O

35

- Obyčajne **len** CPU môže pristupovať priamo k adresám pamäte
- Ostatné pamäte - cez radiče
- V niektorých systémoch
 - ▣ registre radiča sú mapované do OP (RAM)
 - ▣ výsledok – uniformná práca s perifériami
 - môžu sa obslúžiť cez procedúry správy pamäte
 - adresovanie je len v OP
 - prístup cez DMA, obsluha prerušení, ...

Pásky

36

- Veľmi podobné audio páskam
- Sekvenčný prístup : read/write hlava
- Ked' je nastavená pozícia – prenos je veľmi rýchly
- Je stála : back-up súborov
 - ▣ Magnetická búrka ju môže zmazať
- Rozmery a hustoty sú rôzne
 - ▣ teraz 8mm páska má najväčšiu hustotu a na ňu sa zmestí 5Gb dát (350-stôp)

- Zvláštny typ periférneho zariadenia
- Používajú sa na počítanie uplynulého (systémového) času
- Pre
 - ▣ prerušenie zastavených procesov
 - ▣ prepínanie kontextu v time-sharing systémoch
- Prístup k ním je obyčajné privilegovaný

□ Potreba ochrany

▣ ak užívateľský proces má prístup k

- pamäte procesov iných užívateľov
- k I/O zariadení iných užívateľov

▣ môže ich poškodiť

- vedome
- nevedome

Možné porušenia ochrany

39

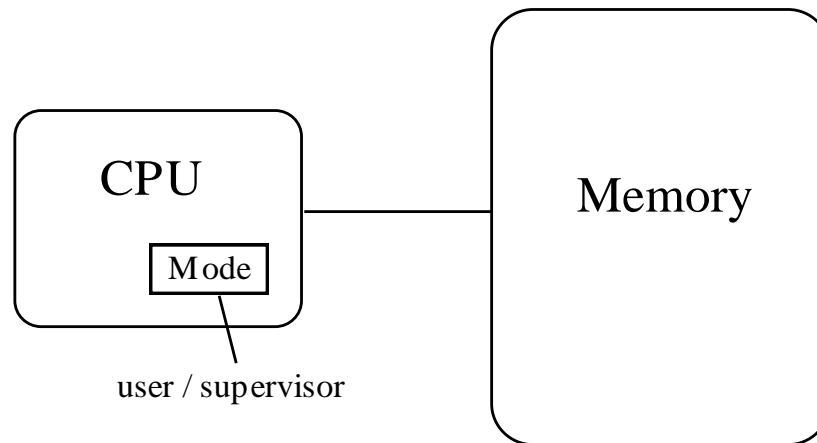
- Užívateľský proces priamo pristupuje k pamäť iného užívateľského procesu
- Užívateľský proces intervenuje do I/O iného užívateľského procesu cez zásah do I/O ovládača a nepriamo poškodzuje dáta
- Užívateľský proces vstúpi do OS a prepisuje iné užívateľské procesy alebo samotný OS
 - MS-DOS a Macintosh OS to dovoľujú
- Zastaviť chod OS – napr. cez nekonečnú slučku

Režim užívateľa/supervisora

40

Prepínanie režimov na základe zmeny 1 bitu

Musí byť poskytnutá hardvérom



Privilegované inštrukcie

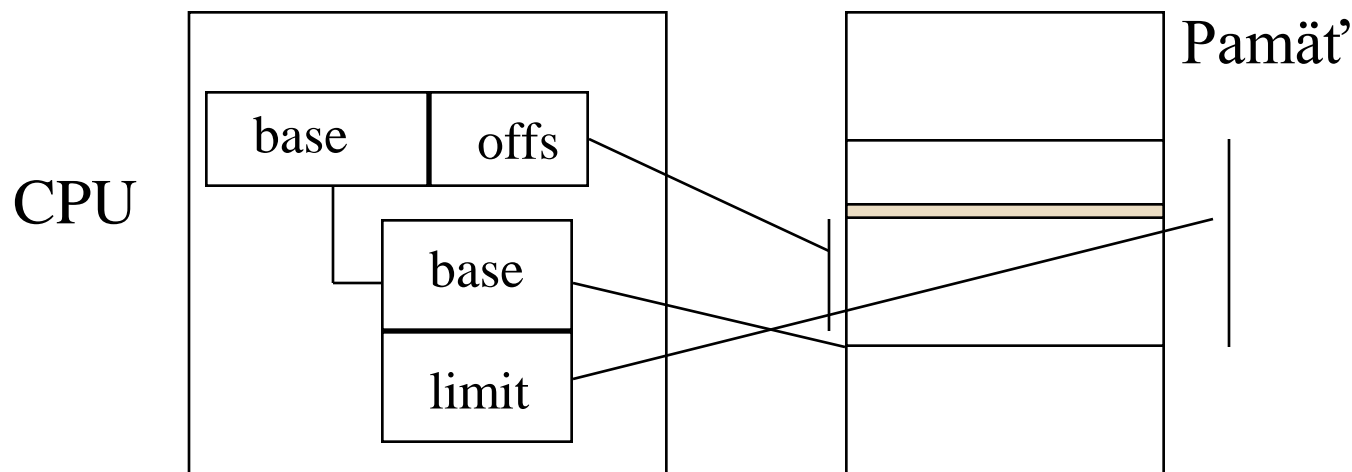
41

- Privilegované inštrukcie sú inštrukcie, ktoré sa vykonávajú v režime supervizora a riadia
 - ▣ I/O, vrátane riadení prerušení
 - výnimkou sú inštrukcie, ktoré generujú prerušenie a môžu vzniknúť aj pri mapovaní pamäťových adries
 - ▣ správu pamäte
- HW podpora, rozhodujúca pre výkon a atomickosť (nedeliteľnosť)

Ochrana pamäte

42

- Závisí od techniky správy pamäte - segmentácia alebo stránkovanie
- ▣ Napr. ak pamäť je rozdelená na segmenty
 - pre každý segment a pre každý užívateľský proces je
 - *base address* – počiatočná adresa, *offset* – požadovaná adresa, *limit* - rozsah platných adries



Ochrana pamäte

43

- ▣ Ak pamäť je rozdelená na stránky
 - každá stránka je prístupná cez tabuľku stránok procesu
 - samotná technika správy pamäte zabezpečuje ochranu