



KIV/ZOS – 11. CVIČENÍ

L. Pešička, verze 2011

PŘIDĚLOVÁNÍ PAMĚTI PO SEKCÍCH DLE POŽADAVKU PROCESŮ

- mějme paměť 100MB
- požadavky na alokaci:
 1. P1 20MB
 2. P2 5MB
 3. P3 10MB
 4. uvolníme P1, P2
 5. alokujeme P4 22MB
 - first fit
 - next fit
 - best fit
 - P5 20MB

všimněte si rozdílné
alokaci pro
first fit a next fit

všimněte si vzniku
malé díry,
oblasti, která bude sotva
přidělitelná nějakému
procesu

vnější fragmentace =
(reálně) nepřidělitelný
úsek paměti

PŘIDĚLOVÁNÍ PAMĚTI PO SEKCÍCH PEVNÉ VELIKOSTI

- Při startu systému určíme na kolik sekcí a jakých velikostí bude paměť rozdělena
- Např 100MB RAM na 4 sekce (50, 30, 15, 5)
- Požadavky na alokace:
 1. P1 25MB
 2. P2 44MB
 3. P3 7MB
 4. P1 a P3 skončí
 5. potřebujeme P4 35MB, co bude?



STRÁNKOVANÁ PAMĚŤ

- kolik je tabulek stránek?
- kolik je tabulek rámců?
- co je to výpadek stránky?
- může offset překročit velikost stránky/rámce?
- procesy používají VA nebo FA?
- k adresaci RAM se používá VA nebo FA?
- jak je zajištěno, aby proces nemohl sahat do paměti jiného procesu?
- kde je uložena tabulka stránek?
- k čemu slouží TLB?
- kolik adresních prostorů náš proces P1 má?



TABULKA STRÁNEK

stránky	rámce	swap
0	2	
1	4	
2	-	Disk , 4096

tabulka stránek
procesu 1

Tabulka stránek procesu říká následující:

stránka 0 procesu P1, tj. virtuální adresy $0 \cdot \text{velikost bloku} - 1$ jsou mapované jako rámec č.2 v RAM, tj. od fyzické adresy $2 \cdot \text{velikost bloku}$

stránka 2 procesu P1 není namapována v RAM, nachází se ve swapu na disku, od adresy např.:4096; při přístupu k ní nastane výpadek stránky – stránka se musí načíst na vhodné místo v paměti a pak se pokračuje dále

pozn, velikost stránky = velikost rámce



VÝPOČET FYZICKÉ ADRESY (FA)

Je dáno:

velikost stránky = velikost rámce = 4096

Spočtětě pro proces1 FA, je-li dáno VA= 5000

Využijte tabulku stránek procesu P1 z předchozího
slidu

Úkol:

Vypočítejte fyzickou adresu.



VÝPOČET FYZICKÉ ADRESY – OBECNĚ

1. Z virtuální adresy zjistíme číslo stránky a offset

$$\text{STRANKA} = \text{VA} / \text{VELIKOST_STRANKY}$$

$$\text{OFFSET} = \text{VA} \% \text{VELIKOST_STRANKY}$$

{kde / je celočíselné dělení, % zbytek po dělení}

2. Z tabulky stránek určíme příslušný rámec

$$\text{CISLO_RAMCE} = \text{TAB_STRANEK}[\text{STRANKA}]$$

{musíme využít tabulku stránek správného procesu}

{není-li stránka v RAM => výpadek stránky}

3. Vypočteme fyzickou adresu:

$$\text{FA} = \text{CISLO_RAMCE} * \text{VELIKOST_RAMCE} + \text{OFFSET}$$

VÝPOČET FYZICKÉ ADRESY – PŘÍKLAD

1. číslo stránky, offset

$$\text{číslo stránky} = 5000 / 4096 = 1$$

$$\text{offset} = 5000 \% 4096 = 904$$

2. určení rámce

$$\text{číslo rámce} = \text{tab_stranek}[1] = 4$$

3. výpočet FA

$$\text{FA} = 4 * 4096 + 904 = 17288$$



VYUŽITÍ PAMĚTI PROCESEM

- mějme virtuální adresní prostor pro proces
např. 2GB
- načrtněte, jak může být adresní prostor využit procesem (z kterých částí se proces skládá?)
 - kód
 - data
 - sdílené knihovny
 - zásobník
- pojmenujte problémy, které 1 adresní prostor může přinést



SEGMENTOVANÁ PAMĚŤ

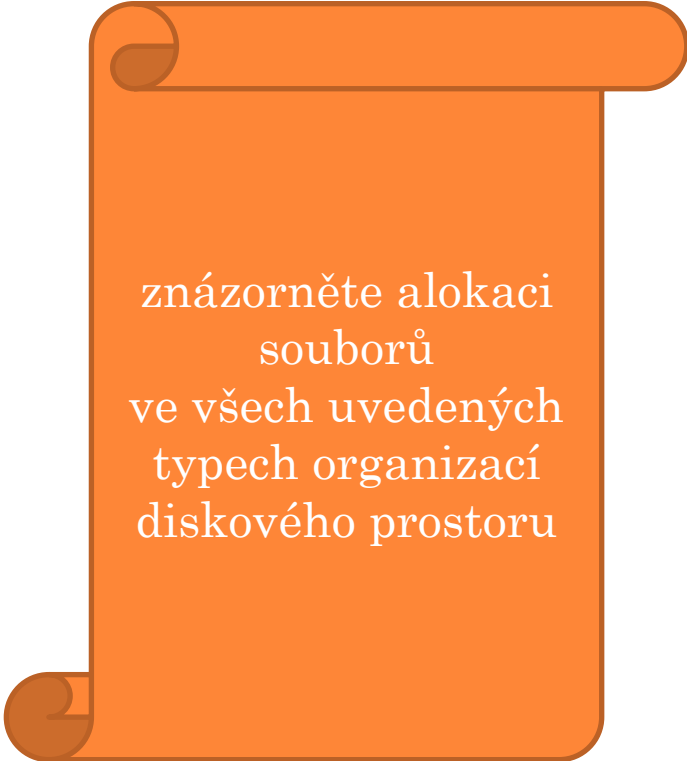
- proč může být výhodné využívat segmenty?
- kolik adresních prostorů má P1 k dispozici?
- musí se celý segment vejít do paměti?
- musí se všechny segmenty procesu vejít do paměti?
- co to znamená segmentace se stránkováním?



SOUBORY

- kontinuální alokace
- seznam diskových bloků
- FAT
- i-uzel

jsou dány tři soubory
alokační jednotka je 4KB
soubor A má velikost 16KB
soubor B má velikost 1KB
soubor C má velikost 9KB

An orange scroll graphic with a rolled-up top edge and a small circle at the bottom right.

znázorněte alokaci
souborů
ve všech uvedených
typech organizací
diskového prostoru

KONTINUÁLNÍ ALOKACE

- jaké jsou výhody?
- jaké jsou nevýhody?
- kdy tyto nevýhody nevadí a lze kontinuální alokaci použít?



SEZNAM DISKOVÝCH BLOKŮ

- část datového bloku se použije pro uložení adresy následujícího bloku
- nevýhoda
datový blok má takovou velikost (mocniny 2), aby se s ním dobře pracovalo, vešel se do různých bufferů atd..
pokud z jeho velikosti část odebereme na ukazatele, o tuto výhodu přijdeme



FAT

- seznam ukazatelů jde do samostatné tabulky
- proč se tabulka FAT často duplikuje do záložní tabulky?
- jaký má význam defragmentace disku?
- značky ve FAT
 - index následujícího bloku
 - konec souboru (označíme např. -1)
 - volný blok (označíme např. -2)



I-UZEL

- máme-li 5 souborů, kolik potřebujeme i-uzlů?
- v čem je i-uzel výhodný pro malé soubory?
- je součástí i-uzlu i název souboru?
- jsou součástí i-uzlu i unixová práva (user, group, others)?
- proč je součástí i-uzlu i počet odkazů na soubor?



NTFS

- kódování délkou běhu
- od pozice 0 např. máme:
A1, A2, A3, B1, B2, A4,A5, C1, ...
- soubor A bude popsán **fragmenty**
- fragment: index, počet bloků daného souboru
- tedy:
 - 0, 3 -- od indexu 0 tři bloky patří souboru A
 - 5,2 -- od indexu 5 dva bloky patří souboru A



KONTROLA KONZISTENCE SOUBORŮ

- kontrola, zda je diskový blok volný, nebo přiřazen právě jednomu souboru
- kontrola, zda je soubor viditelný z alespoň jednoho adresáře
- kontrola, zda počet výskytů souboru v adresářích odpovídá počtu odkazů na soubor v i-uzlu
- jak se pozná, že file systém nebyl korektně odpojen?



SPRÁVA PAMĚTI – KTERÝ RÁMEC UVOLNIT?

- plná paměť a nastane výpadek stránky
- křišťálová koule
 - potřebovali bychom se podívat do budoucnosti
- LRU
 - pohled do minulosti je dobrou aproximací
- FIFO
- NRU
- Second Chance, Clock
- Aging

