

Memory management

Andrea Číková
Martin Osovský

Mechanizmy pre prácu s pamäťou

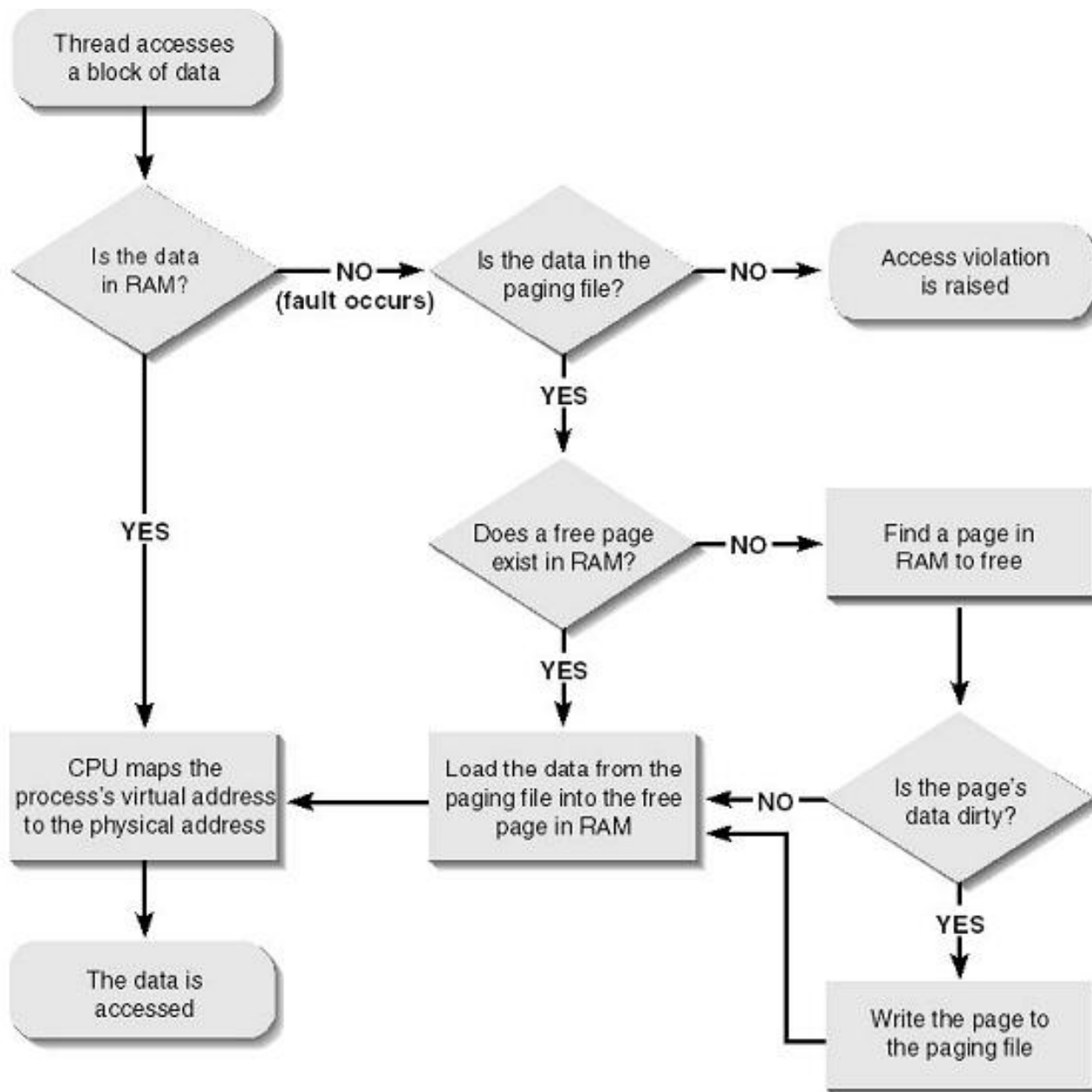
- **Virtuálna pamäť** – vhodná na správu veľkých polí objektov a štruktúr
- ***Memory-mapped files*** – vhodné na správu obsiahlych prúdov dát
- **Halda (*heaps*)** – vhodné na správu veľkého množstva malých objektov

Virtuálna pamäť – adresový priestor procesu

- Každý proces má svoj vlastný adresový priestor – vlákno môže pristupovať iba k adresovému priestoru svojho procesu
- 32-bitový proces = 4GB adr. priestor
- 64-bitový proces = 16EB (exabytes) adr. priestor
- Jedná sa o **virtuálny** adresový priestor, nie fyzickú pamäť!!

Free, reserved, committed,...

- Časť adresového priestoru po vytvorení je voľná (**free**)
- Pre použitie časti tohto priestoru musíte alokovať oblasť (*region*) volaním funkcie *VirtualAlloc* – daná časť je potom rezervovaná (**reserved**); *veľkosť oblasti násobkom veľkosti stránky*
- Pre použitie rezervovanej oblasti musíte alokovať fyzickú pamäť volaním funkcie *VirtualAlloc* – **committing**
- Uvoľnenie fyzickej pamäte – *VirtualFree* – **decommitting**
- Uvoľnenie rezervovanej oblasti – *VirtualFree* - **releasing**




Funkcia *VirtualAlloc*

```
PVOID VirtualAlloc(  
    PVOID address,  
    SIZE_T regionSize,  
    DWORD allocationType,  
    DWORD protection);
```


allocationType:

- MEM_RESERVE – rezervuje oblasť v adr. priestore
- MEM_COMMIT – pridelí oblasti fyzickú pamäť
- MEM_RESERVE | MEM_COMMIT – rezervuje oblasť a pridelí jej fyzickú pamäť v jednom kroku


Zásobník vlákna

Memory Address	State of Page
0x080FF000	Top of stack: committed page
0x080FE000	Committed page with guard protection attribute flag
0x080FD000	Reserved page
	
0x08003000	Reserved page
0x08002000	Reserved page
0x08001000	Reserved page
0x08000000	Bottom of stack: reserved page

Zásobník vlákna

Memory Address	State of Page
0x080FF000	Top of stack: committed page
0x080FE000	Committed page
0x080FD000	Committed page
	
0x08003000	Committed page
0x08002000	Committed page with guard protection attribute flag
0x08001000	Reserved page
0x08000000	Bottom of stack: reserved page

Zásobník vlákna

Memory Address	State of Page
0x080FF000	Top of stack: committed page
0x080FE000	Committed page
0x080FD000	Committed page
	
0x08003000	Committed page
0x08002000	Committed page
0x08001000	Committed page
0x08000000	Bottom of stack: reserved page

Memory-mapped files

- EXE a DLL súbory namapované do pamäte
- Dátové súbory namapované do pamäte
- Zdieľanie dát medzi procesmi pomocou *memory-mapped* súborov

Namapovanie súborov

- Vytvoriť objekt súboru – *CreateFile*
- Vytvoriť *file-mapping* objekt – *CreateFileMapping*
- Vytvoriť *view* (namapovať *file-mapping* objekt) – *MapViewOfFile*
- Práca s namapovaným súborom
- Odstránenie namapovania – *UnmapViewOfFile*
- Zatvorenie objektu súboru a *file-mapping* objektu

Halda

- Halda je oblasť rezervovaného adresového priestoru.
- Vhodná pre správu veľkého množstva malých objektov.
- Sériový prístup
- Implicitná halda procesu – `GetProcessHeap()`
- Dodatočné haldy procesu - `CreateHeap`

Dôvody pre vytvorenie dodatočnej haldy

- Ochrana komponentov
- Efektívnejšia správa pamäte
- Lokálny prístup
- Rýchle uvoľnenie pamäte

Otázky?

Ďakujem za pozornosť😊