

Efektivita využití vláken, vláken A, \dots, D RR

$A \dots C$ Ready / Running $D = \text{blocked}$

přepnutí kontextu = 10 ms časové kvantum = 90 ms

$$\text{Efektivita} = \frac{\text{kvantum}}{\text{kvantum} + \text{přepnutí}} = \frac{90}{100} = 90\%$$

$$\text{Odcas} = 3 * (\text{přepnutí} + \text{časové kvantum})$$

event, ostatní vlákna + přepínání, lock

Strategie pro dosažení tohoto cíle může být následující

- **Priorita se zvýší a časové kvantum sníží**
 - ★ pokud vlákno v posledním běhu nevyužilo celé své časové kvantum (vlákno orientované na V/V),
 - ★ pokud vlákno dlouho čeká na CPU (hrozí problém hladovění vlákna).
- **Priorita se sníží a časové kvantum se zvýší**
 - ★ pokud vlákno v posledním běhu využilo celé své časové kvantum (vlákno orientované na CPU).

Výkonnostní parametry

- **Cache hit count** (n_h): počet případů, kdy data byla ve skryté paměti,
- **Hit time** (t_h): čas přístupu k datům ve skryté paměti,
- **Cache miss count** (n_m): počet případů, kdy data nebyla ve skryté paměti,
- **Miss penalty** (t_m): čas přístupu k datům ve zdroji dat (data source),
- **Cache reference**: celkový počet přístupů k datům $n_r = n_h + n_m$,
- **Cache Hit Ratio**: $r_h = \frac{n_h}{n_r} = \frac{n_h}{n_h + n_m}$,
- **Average Access Time**: $t_{avg} = t_h + (1 - r_h) \times t_m$.

$$L1 \text{ cache } t_h = 1 \text{ ns}$$

$$\text{Hlavní paměť } t_m = 50 \text{ ns}$$

$$\text{Hit Ratio} = 0.8$$

$$T_{avg} = t_h + (1 - r_h) \times t_m$$

$$= \underbrace{1}_{\text{Přístup do cache}} + \underbrace{(1 - 0.8) \times 50}_{\text{Přístupy do Ram}}$$

Přístup do cache

Problém s fragmentací fyzické paměti

- ▶ **Externí fragmentace:** po určitém čase je volná paměť reprezentována příliš malými oblastmi, do kterých se již nevejdou nově vznikající procesy.
- ▶ **Interní fragmentace:** VAS obsahuje volnou paměť, do které se může rozvíjet halda a zásobníky vláken.

● Příklad

- ▶ Na 32-bitovém CPU, které podporuje pouze 4KB stránky/rámce, je nainstalovaný 32-bitový OS, ve kterém běží 32-bitový proces. Systém používá dvouúrovňovou tabulku stránek a indexy do jednotlivých tabulek mají stejnou velikost.
- ▶ Předpokládáme, že proces bude alokovat ve svém VAS pouze následující datové struktury
 - ★ TEXT a DATA: 4 MB na nejnižších virtuálních adresách,
 - ★ halda: 4 MB na následujících virtuálních adresách,
 - ★ zásobník: 4 MB na nejvyšších virtuálních adresách.

● Jaká bude struktura virtuální a fyzické adresy?

- ▶ Virtuální adresa (32 bitů): level 1 index (10 bitů) + level 2 index (10 bitů) + offset (12 bitů).
- ▶ Fyzická adresa (32 bitů): číslo rámce (20 bitů) + offset (12 bitů).

● Kolik řádek bude mít top level tabulka? $\Rightarrow 2^{10}$ řádek.

● Kolik místa zabírá jedna řádka?

- ▶ Číslo rámce (20 bitů) + kontrolní bit (P) \Rightarrow zaokrouhlíme ~ 32 bitů = 4 B.

● Kolik řádek bude mít tabulka druhé úrovně? $\Rightarrow 2^{10}$ řádek.

● Kolik místa zabírá jedna řádka?

- ▶ Číslo rámce (20 bitů) + kontrolní bity (P...) \Rightarrow zaokrouhlíme ~ 32 bitů = 4 B.

● Kolik místa zabírají všechny tabulky pokud na systému běží 2^7 podobných procesů?

- ▶ $2^7 \times [1 \times (2^{10} \times 4) + 3 \times (2^{10} \times 4)] \text{ B} = 2^7 \times 16 \text{ KB} = 2 \text{ MB}$.

$$1 \times \text{top level} + 3 \times 2.\text{level}$$

↓

Alokujeme 12 MB, jedna tabulka 2 úrovně po 2 x 4KB = 4MiB

$$12/4 = 3$$

Náhled stránek - Optimální \Rightarrow Náhled tu co bude použit 2x co největší dobu

$$\text{Stránky} = \underline{2}, \underline{3}, \underline{2}, \underline{1}, \underline{5}, \underline{2}, \underline{4}, \underline{5}, \underline{3}, \underline{2}, \underline{5}, \underline{2}$$

řádky a, b, c

a	2		2			2	4			2		2
b		3							3			
c				1	5			5			5	
	x	x		x	x		x			x		

6x page Fault

Náhradní stránka - NRU (Not Recently Used)

2 bity R - reference bit, M - modified bit dle nejnovějšího RM vyhodíně

	w			reset			reset			reset			w w		
	2	3	2				1	5					2	5	2
a	Stránka	2		2			2			2			2		2
	R	1		1	0		0			1			0	1	1
	M	0		1	1		1			1			1	1	1
b	str		3		3		5	5		4		3	3		
	R		1		0		1	0		1		1	0		
	M		0		0		0	0		0		0	0		
c	str					1		1			5		5		5
	R					1		0			1		0		1
	M					0		0			0		0		1
		x	x			x	x			x	x	x			

=> 7x Page Fault

Náhradní stránka FIFO

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

a	2		2		5			5	3				
b		3			2				2	5			
c				1			4				2		
	x	x		x	x	x	x		x		x	x	

=> 9 x Page Fault, boha vyhovuje první uložení, Ne neposledy použitou

Náhrada Stránky LRU (Least Recently used)

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

a	2		2			2			3			
b		3		5			5			5		
c				1		4				2		2
	x	x		x	x		x		x	x		

=> 7 výpadků

Náhrada Stránky - Clock

- počítas utvoříš na první vložení, při vložení R na 1, pokud utvoříš na $R \neq 0$

$R=0$, posuň

nahrad, posuň

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2
R

a	Stránky	2		2		5			5	3			
	R	1		1		1			1	1			
b	Stránky		3			3	2			2	2	2	2
	R		1			0	1			0	1	0	1
c	Stránky				1	1		4		4		5	
	R				1	0		1		0		1	
clock		a	a	a	a	b	c	a	a	b		a	
		x	x		x	x	x		x			x	

=> 8x Page Fault

Nehms Story - Aging

- R - reference bit, n-bitový čítač C, nastaven na součet 1 \Rightarrow pro 8 bitů $= 7$

Periodicity: $C_1 = 2$

: nejvyšší bit C nastavit na 1

$$\therefore R = 0$$

- höhersteigende Struktur s. höchstem C

Číslo měření obsazení řádku v průběhu času:

Číslo stránky	2	3	2		1	5		2	4	5	3		2	5	2
Čas	1	4	9	10	15	17	20	21	22	25	27	30	31	32	37

	R				R				R							
a	Students	2		2	2		5	5		4	5	3	3			
	R	1		1	0		1	0		1	1	1	0			
	C	7		7	7		7	7		7	7	7	7			
b	Students		3		3			3	2				2	2		2
	R		1		0			0	1				0	1		1
	C		7		7			3	7				7	7		7
c	Students					1		1					1		5	
	R					1		0					0		1	
	C					7		7					3		7	
		X	X			X	X	X	X	X	X	X			X	

9x Page Fault

- Mějme disk s těmito parametry

- ▶ HDD má pouze jeden povrch, který se otáčí rychlostí 10000 rpm.
- ▶ Velikost sektoru 512 B.
- ▶ Každá stopa má 320 sektorů.
- ▶ Průměrný čas vystavení hlaviček (seek time) je 10 ms.
- ▶ Vystavení hlaviček nad sousední stopu (track-to-track seek time) je 1 ms.

2560 slots

$$RPS = \frac{10000}{60} \quad \text{1 tot spóviedení} = \frac{1}{\underbrace{\frac{10000}{60}}_{\text{dobu doby}}} / 2 = 3 \text{ ms}$$

$$1_{\text{setLen}} = 10 + 3 + 6 = 19$$

↑
↑
set
steps

$$2-8 \text{ sektoy} = 7 + \underset{\substack{\uparrow \\ \text{sek}}}{(1+3+6)} = 70 \quad \Rightarrow 89 \text{ ms}$$

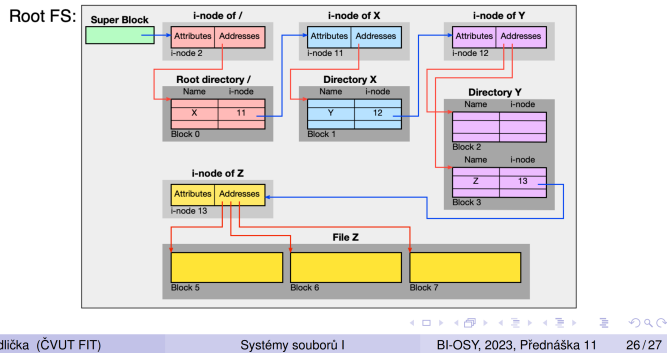
Náhodní po distu

Нојетни једнобо сејторо = $10 + 3 + \frac{6}{326}$

$$\text{Altum} = \left(13 + \frac{6}{320}\right) \times 2560 = 33,3285$$

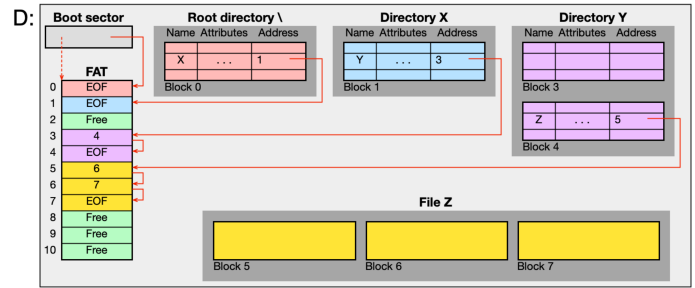
● Přístup k souborům/adresářům

- Super blok obsahuje informaci, kde začínají struktury pro správu volného prostoru, tabulka i-nodů a datové bloky.
- Po připojení UFS se do paměti načte i-node kořenového adresáře (i-node číslo 2).
- Pokud chceme např. zobrazit obsah souboru /X/Y/Z, pak musíme načíst z disku příslušné i-nody a datové bloky.



● Přístup k adresářům/souborům

- Boot sektor obsahuje informaci, kde začíná FAT, kopie FAT a obsah kořenového adresáře (Root directory).
- Při připojení FS se načte do hlavní paměti celé/část FAT.
- Pokud chceme např. zobrazit obsah souboru D:\X\Y\Z, pak musíme načíst z disku příslušné datové bloky.



Zpsst soubor 781 9264 B

$$\text{počet cylindrů} = 781\,9264 / (4096 * 512) = 0,12 \dots$$

$$\text{počet sektorů} = 1909$$

$$\text{přičtení jednoho sektoru} = \frac{6}{512}$$

$$C_{\text{celkem}} = 1909 * (3 + 9 + \frac{6}{512}) = 22,9095$$

$$\text{tps} = \frac{10000}{60}$$

$$\text{lat spojení} = \frac{1}{\frac{10000}{60}} / 2 = 3 \text{ ms}$$

① Remove -tf

1) Kolik i-nodes musím načíst? Počet adresářů v cestě + 1

2) Kolik i-nodes se zapíše? Pouze počet Dir souboru => 1

3) Kolik datových bloků se načte?

- počet Adresářů + počet dat. bloků potřebných k uložení souboru (1., 2. úroveň bloky)

! musíme samostatně dat. bloky souboru

4) Kolik dat. bloků se zapíše? 1 dat. blok v počtu možného

5) Kolik se píše Alokační/Dealozační

1x i-node, počet datových bloků souboru, počet dat. bloků pro tabulku (1., 2. úroveň)

② Symbolic link

- 1) Kolik i-nodes se načte? $\text{musíme pouze jméno linku} = 2 \text{ argument} \rightarrow \text{počet odkazů? } 2. \text{ arg}$
- 2) Kolik i-nodes se zapíše? $2 \rightarrow \text{počet dir a nový link}$
- 3) Kolik dat bloků se načte? $\text{počet odkazů? } 2. \text{ arg}$
- 4) Kolik dat bloků se zapíše? $\text{blok pro cesta v linku} + \text{počet dir}$
- 5) Opusne alokujeme bloky + i-nodes? $\text{i-node linku, dat blok linku}$

③ Hard Link

- 1) Kolik i-nodes se načte? $\text{Musíme dojít ke konce odkazování, můžeme i-node, dojít do cíle} \Rightarrow \text{počet Adm obrac Arg}$
- 2) Kolik i-nodes se zapíše? $\text{ex. soubor, počet dir nového (RefCount, více nového)}$
 $+ 1$
 \uparrow
 i-node souboru
 $\text{musíme zvednout RefCount}$
- 3) Kolik dat bloků se načte? $\text{oba odkazy (1...1)}$
- 4) Kolik dat bloků se zapíše? $\text{pouze jeden v pořadí}$
- 5) Opusne alokujeme bloky + i-nodes? \bigcirc

④ Copy

cp /x/y /z/a
cc krm

- 1) Kolik i-nodes se načte? $\text{Adresy argumentů} + \text{soubor co kopírujeme}$
- 2) Kolik i-nodes se zapíše? $\text{počet} + \text{nový soubor}$
- 3) Kolik dat bloků se načte? $\text{Adresy argumentů} + \text{soubor (soubor dat + tabulky 1., 2. úrovně)}$
- 4) Kolik dat bloků se zapíše? $\text{soubor (...) + počet}$
- 5) Opusne alokujeme bloky + i-nodes? $1 \times \text{i-node} + \text{soubor}$

⑤ Move

mv /x/y /z/y
cc krm

- 1) Kolik i-nodes se načte? soubor arg
- 2) Kolik i-nodes se zapíše? $\text{pouze dva parametry} \Rightarrow 2 \quad (\text{na i-node souboru se pouze zkopíruje odkaz})$
- 3) Kolik dat bloků se načte? soubor arg
- 4) Kolik dat bloků se zapíše?
- 5) Opusne alokujeme bloky + i-nodes?

6) Mame do jiného FS (je to spíše tabulky cp)

1) Kolik i-nodes se načte? součet avg + 1

2) Kolik i-nodes se zspíše? 2 x percent + nový i-node

3) Kolik dat bloků se načte? součet avg + celý soubor (data, tabulky)

4) Kolik dat bloků se zspíše? celý soubor + 2 dat bloky adresič?

5) Operace alokace bloky + i-nodes? 2 + celý soubor + 2

\uparrow \uparrow
 Alloc/dealloc Alloc, dealloc i-node
 soubor

Máme k dispozici pevný disk s následujícími parametry:

- Kapacita: 1 TiB
- Velikost sektoru: 4096 B
- Počet hlav: 2
- Počet cylindrů: 32768
- Průměrný počet sektorů na cylindru: 8192
- Rychlost: 10000 rpm
- Průměrná doba vystavení hlaviček: 9 ms
- Doba vystavení hlaviček na sousední cylindru: 3 ms

1. Jak dlouho bude trvat přečíst soubor o velikosti 7819264 B z disku, pokud je jeho obsah uložen na jednotlivých sektorech náhodně umístěných po disku?

Čas zadejte v ms, tolerance 1%.

$$\text{tps} = \frac{10000}{60}$$

$$\text{počet sektorů} = 7819264 / 4096 = 1909$$

$$\text{jedna otáčka} = \frac{1}{\frac{10000}{60}} = 6 \text{ ms}$$

$$\text{celkem} = 1909 * (3 + \frac{6}{8192} + 9)$$

$$\text{tot spoi.} = 3 \text{ ms}$$

...

V takové konfiguraci se provede následující příkaz shellu:

```
# rm -f /var/boot/etc/doc/local/etc/index.html
```

Uvažujte pouze efekt tohoto příkazu (tedy neuvažujte režii spojenou s vlastním natažením tohoto příkazu do paměti).

1. Kolik i-nodů se musí načíst?
2. Kolik i-nodů se musí zapsat?
3. Kolik datových bloků se musí načíst?
4. Kolik datových bloků se musí zapsat?
5. Kolik operací při alokování/uvolňování bloků a i-nodů musí být provedeno? Každou operaci alokace/uvolnění bloků/i-node započítáte zvlášť.

1) 2

2) pouze percent souboru => 1

3) 7 + 1 + 1 + 21 = 30

$$\text{bloky} = 1422467929837 / 16 \text{ KiB} = 86820552 \text{ bloku}$$

$$\text{počet adresič v bloku} = 16 \text{ KiB} / 8 = \underline{2048}$$

$$2048^2 = 4194304$$

$$2048^3 = \text{enough}$$

$$\text{Zbude } 86820552 - 12 - 2048 = 86818492$$

$$\text{plně uvolnen} + 2 \text{ uvolně} + \left\lceil \frac{86818492}{2048^2} \right\rceil = 7 + 1 + 1 + 21 = \underline{\underline{30}}$$

\uparrow
 21

4) 1 percent souboru

5) 30 + 1 + 86820552

\uparrow \uparrow \uparrow
 tabulky i-node soubor

Uživatel	Proces	Statická priorita	Počet vláken	Celková doba výpočtu [sec]
xvagner	P ₀	0	3	108
trdlicka	P ₁	26	1	54
trdlicka	P ₂	80	3	270
zdarekj	P ₃	71	1	45
zdarekj	P ₄	0	4	72
soch	P ₅	26	2	72

4 Jídel

$$T = 0 \quad c_{1,2,3} \quad P_2$$

$$c_4 \quad P_3$$

$$P_0 \text{ 45 s dobíhání } (P_3)$$

$$\text{Celkem odpracováno } 45 \cdot 4 = 180$$

$$\text{z } P_2 \text{ zbude } 270 - (180 - 45) = 135 \text{ s}$$

$$T = 45 \quad c_{1,2,3} \quad P_2$$

$$c_4 \quad P_1, P_5 (1:2)$$

$$P_0 \quad 135/3 = 45 \text{ s dobíhání } (P_2) \quad T = 90$$

$$\text{Celkem odpracováno } = 45 \cdot 4 = 180$$

$$\text{bc z } P_2 = 180 - 135 = 45$$

$$\text{z } P_1 \text{ zbude } 54 - \frac{45}{3} \cdot 1 = 39$$

$$\text{z } P_5 \text{ zbude } 72 - \frac{45}{3} \cdot 2 = 42$$

$$T = 90 \quad c_1 \quad P_1$$

$$c_{2,3} \quad P_5$$

$$c_4 \quad P_0, P_4 (3:4)$$

$$P_0 \quad 42/2 \text{ sekund slevit } (P_5) \quad T = 90 + 21 = 111$$

$$= 21$$

$$\text{z } P_1 \text{ zbude } 39 - 21 = 18$$

$$\text{z } P_0 \text{ zbude } 108 - 21 \cdot \frac{3}{7} = 99$$

$$\text{z } P_4 \text{ zbude } 72 - 21 \cdot \frac{4}{7} = 60$$

$$T = 111 \quad c_1 \quad P_1$$

$$c_{2,3,4} \quad P_0, P_4 (3:4)$$

$$P_0 \quad 18 \text{ s slevit } P_1 \quad T = 129$$

$$\text{Celkem } 3 \cdot 18 = 54$$

$$\text{z } P_0 \text{ zbude } 99 - 54 \cdot \frac{3}{7} = 75,8571$$

$$\text{z } P_4 \text{ zbude } 60 - 54 \cdot \frac{4}{7} = 29,1429$$

$$\frac{\text{celkový čas} \cdot (\text{max čísl})}{\text{počet dostupných vláken}}^{-1}$$

$$T = 129 \quad c_{1,2,3,4} \quad P_0, P_4$$

$$(3:4)$$

$$P_0 \quad \frac{29,1429 \cdot \frac{7}{3}}{4} = 12,75 \quad \text{dobíhání } (P_4) \quad T = 141,75$$

$$\text{z } P_0 \text{ zbude } 75,8571 - (12,75 \cdot 4) \cdot \frac{3}{7} = 54$$

$T = 141.75$ c_1, c_2, c_3 p_0 p_0 54 / 3 bude víc hotovo
 c_4 víc $= 130$

$T = 159.75$