

Systemy souborů

Milan Radojčić

SSPŠaG – Operační systémy

19. 10. 2026



- Na HDD se můžeme dívat jako na pole bloků^{<1>}.

Čím se budeme zabývat

- Jak můžeme z arraye bloků udělat systém souborů?
 - Jaké informace navíc musíme na disk ukládat?
- Jak tohle všechno dělat efektivně?

^{<1>}Většinou o velikosti 4KB.



- Soubory obsahují nějaká data.
 - Každý soubor má svoje *low-level jméno* – **inode^{<2>} number**.
- Složky obsahují soubory nebo jiné složky.
 - Složky mají také *inode number*.
 - Obsahem složky je **mapping: název/high-level jméno – inode**.
- Inode number unikátně identifikuje soubor/složku v rámci filesystemu.

^{<2>}Zkratka pro index node.



Název	Inode num.
home	4
opt	20
lib	7
lib64	7
...	...

Kořenová složka

Název	Inode num.
mrkev7	42

Inode 4

Název	Inode num.
.bash_history	123
.bashrc	321
Documents	333
...	...

Inode 42

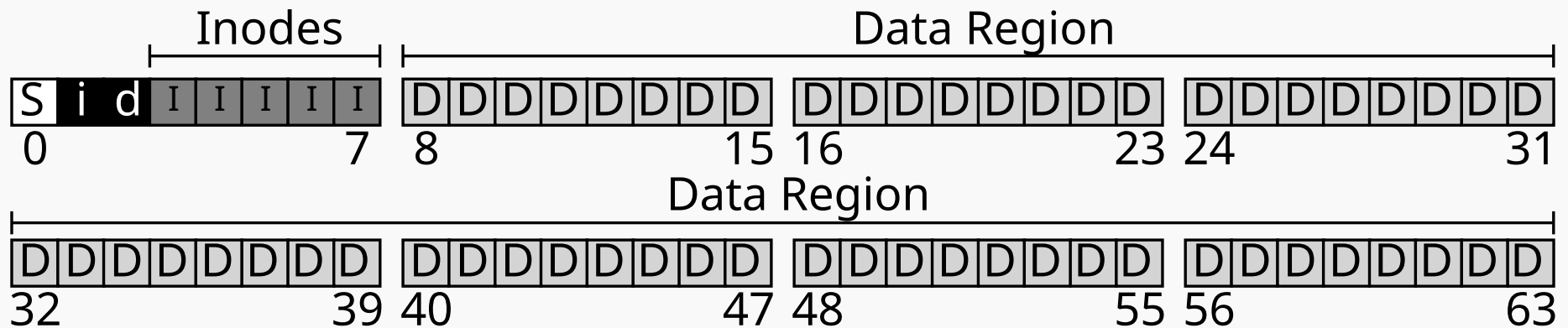


- Ukážeme si **příklad jednoduchého FS**.
- Pro každou složku/soubor budeme potřebovat alokovat bloky pro
 1. samotná data^{<3>} a
 2. metadata (inode number, čas vytvoření/přístupu, přístupová práva, ukazatele na datové bloky, ...).
- Mimo dat a metadat budeme muset i trackovat, které bloky jsou *alokované* a které ne.
 - Můžeme např. použít jednoduchou *bitmapu*.

^{<3>} U složky pro mapování název – inode.



Základní struktura



- s je superblok – obsahuje informace o FS.
- i a d jsou bitmapy pro inode a data sekce.
- I je pole inodů – inode number udává offset v tomto poli.



- Pro nějaký zadaný inode number jsme schopni získat daný inode.
 - Jakým způsobem ale *odkazovat z inode na datové bloky*?



Datové bloky: ukazatele

- Nejjednodušší je použít **ukazatele** (*pointer*).
- Např. můžeme udělat, že každý inode souboru bude obsahovat 12 ukazatelů na datové bloky.

- Jaké limitace vidíte u tohoto přístupu?
- Napadá vás jak je odstranit?



Datové bloky: ukazatele

- Pro rozšíření tohoto přístupu přidáme **nepřímé ukazatele** (*indirect pointer*).
 - Nepřímé ukazatele *ukazují na blok s přímými ukazateli*.
- Každý inode bude kromě 12 přímých ukazatelů obsahovat i *jeden nepřímý*.

- Jaké limitace *stále* vidíte u tohoto přístupu?
- Napadá vás jak je odstranit?



- Chceme ještě rozšířit kapacitu, takže přidáme **dvojitě nepřímé ukazatele** (*double indirect pointer*).
 - Dvojitě nepřímé ukazatele ukazují na blok s nepřímými ukazateli.
- Dál bychom mohli přidat i *trojitě nepřímé ukazatele*.

12 přímých, 1 nepřímý, 1 dvojitě nepřímý a 1 trojitě nepřímý ukazatel používá *Unix File System* (UFS) a *ext3*.



Datové bloky: jiné přístupy

- Jiný způsob jak odkazovat je pomocí **rozsahů** (*extents*).
 - Rozsah je vlastně jen pointer + velikost/počet bloků.
 - Tento přístup používá *ext4*.
- Další cesta je vytvořit **spojový seznam** (*linked list*) bloků.
 - Spoják můžeme implementovat pomocí tabulky, která na disku žije mimo inode a datové bloky.
 - V inode máme jen index prvního bloku v tabulce – tenhle **index je zároveň i adresa bloku**.
 - Na daném indexu v tabulce pak najdeme další blok (pokud soubor pokračuje) nebo nějakou informaci, že tady končí^{<4>}.
 - Anglicky se tomuto říká *file allocation table* (FAT).

^{<4>}Na tohle můžeme třeba vyhradit číslo bloku 0.



Přístup k souboru: čtení

	data bitmap	inode bitmap	root inode	foo inode	bar inode	root data	foo data	bar data [0]	bar data [1]	bar data [2]
open(bar)			read	read	read	read	read			
read()					read			read		
read()					write				read	
read()					read					
read()					write					read
					read					
					write					

Průběh čtení souboru /foo/bar, čas plyne směrem dolů.



Přístup k souboru: vytvoření a čtení

	data bitmap	inode bitmap	root inode	foo inode	bar inode	root data	foo data	bar data [0]	bar data [1]	bar data [2]
create (/foo/bar)		read write	read	read	read write	read	read write			
write()	read write				read write			write		
write()	read write				read write			write		
write()	read write				read write			write		

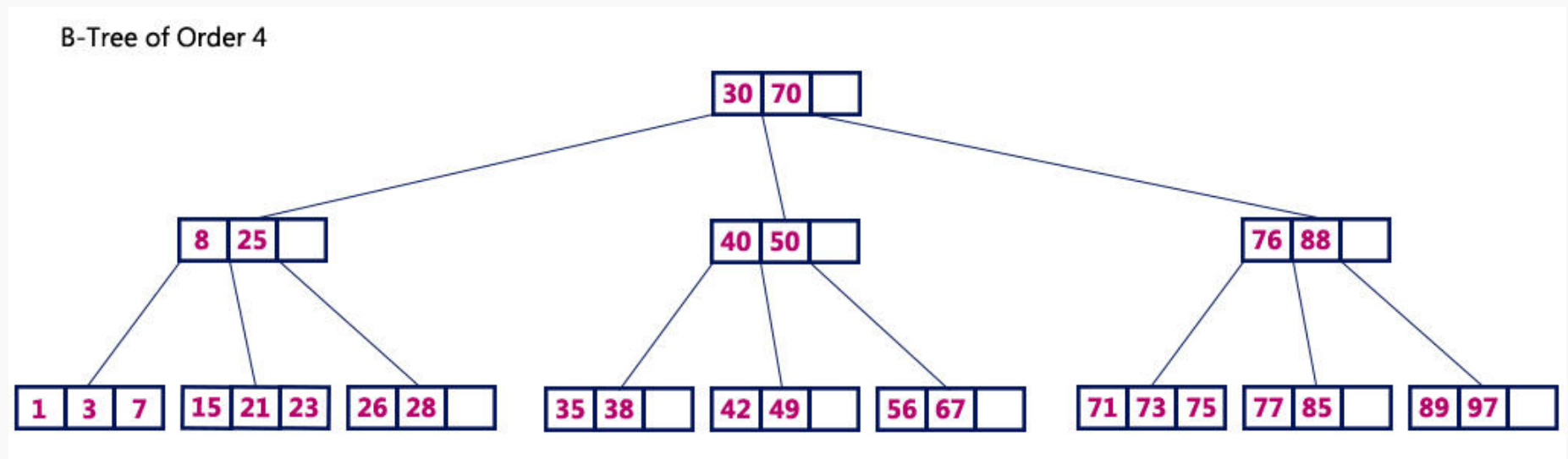
Průběh vytvoření a zápisu do souboru /foo/bar, čas plyne směrem dolů.



- Velký počet čtení a zápisů na předchozích diagramech můžeme samozřejmě snížit **cachováním**.
- Ze dvou různých složek můžeme *odkazovat na stejný inode number*.
 - Tomuto se říká **hard link**.
 - Nutně musí být jen na soubor ve stejném FS.
- Jiný způsob jak odkázat na soubor je *pomocí cesty*.
 - Tomuto se říká **symbolic link** (symlink).



- Jak chceme ukládat seznamy souborů ve složkách?
 - Když je budeme ukládat do pole, tak musíme při vyhledávání projít všechny prvky.
 - Nabízí se použít **B stromy** (*B trees*).



- Kapitoly *Files and Directories*^{<5>} a *File System Implementation*^{<6>} z *Operating Systems: Three Easy Pieces*.

^{<5>} <https://pages.cs.wisc.edu/~remzi/OSTEP/file-intro.pdf>

^{<6>} <https://pages.cs.wisc.edu/~remzi/OSTEP/file-implementation.pdf>

