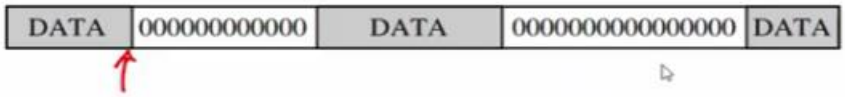
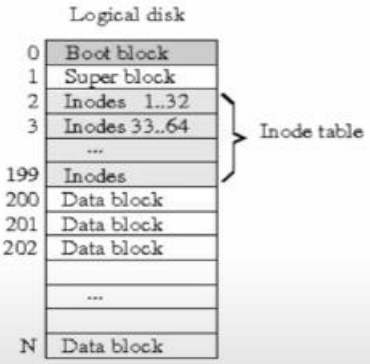
❖ Sparse files – "řídké soubory":

- Vznikají nastavením pozice za konec souboru a zápisem.
- Bloky, do kterých se nezapisovalo nejsou alokovány a nezabírají diskový prostor. Při čtení se považují za vynulované.
- Někdy také „hole punching": mazání prostoru uvnitř souboru (např. `fallocate`).



Klasický UNIXový systém souborů (FS)

boot blok	pro zavedení systému při startu
super blok	informace o souborovém systému (typ, velikost, počet i-uzlů, volné místo, volné i-uzly, kořenový adresář, UUID, ...)
tabulka i-uzlů	tabulka s popisy souborů
datové bloky	data souborů a bloky pro nepřímé odkazy



❖ Modifikace základního rozložení FS v navazujících souborových systémech:

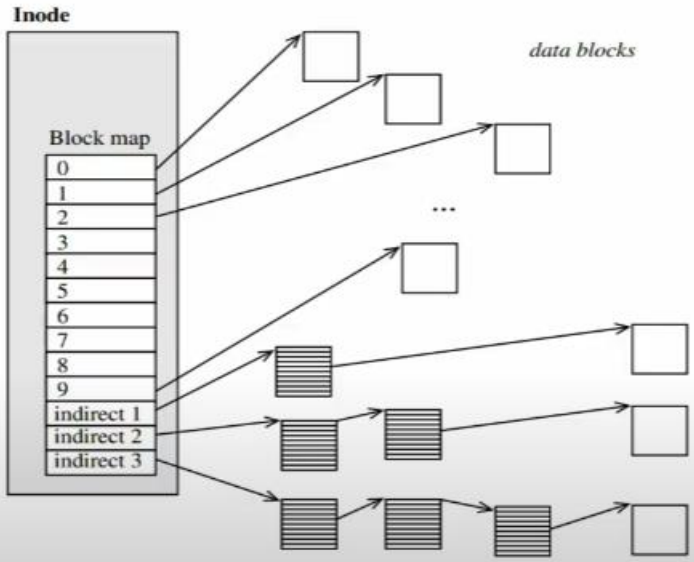
- Disk rozdělen do skupin bloků.
- Každá skupina má své i-uzly a datové bloky a také svůj popis volných bloků: lepší lokalita.
- Superblok se základními informacemi o souborovém systému je rovněž uložen vícenásobně.

i-uzel

❖ Základní datová struktura popisující soubor v UNIX-ových souborových systémech.

- obsahuje metadata, ve speciálních případech i data (např. symbolický odkaz),
- jiné souborové systémy mívají analogické struktury: např. záznam v MFT u NTFS.

❖ U FS, ext2, ext3 (více modifikováno ext4, btrfs, ...):

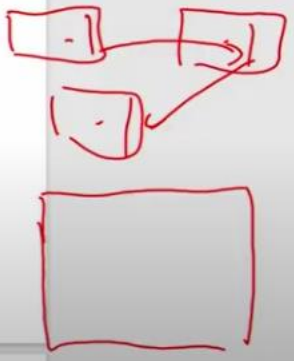


Jiné způsoby organizace souborů

- Kontinuální uložení: jedna spojitá posloupnost na disku.
  - Problémy se zvětšováním souborů díky externí fragmentaci nebo obsazení prostoru hned za koncem souboru.

- Zřetězené seznamy bloků: každý datový blok obsahuje kromě dat odkaz na další blok (nebo příznak konce souboru).
  - Při přístupu k náhodným blokům či ke konci souboru (změna velikosti) nutno projít celý soubor.
  - Chyba kdekoliv na disku může způsobit ztrátu velkého objemu dat (rozpojení seznamu).

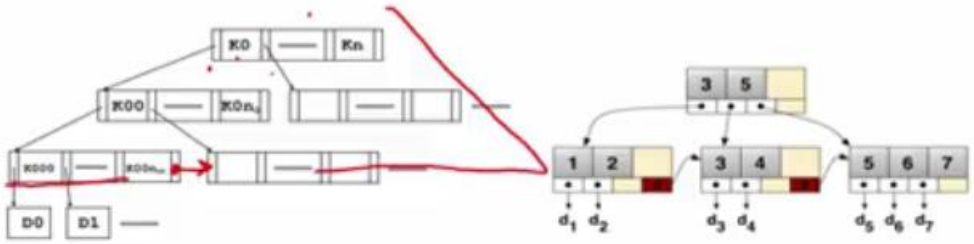
- FAT (File Allocation Table): seznamy uložené ve speciální oblasti disku. Na začátku disku je (pro vyšší spolehlivost zdvojená) tabulka FAT, která má položku pro každý blok. Do této tabulky vedou odkazy z adresářů. Položky tabulky mohou být zřetězeny do seznamů, příp. označeny jako volné či chybné.
  - Opět vznikají problémy s náhodným přístupem.



Jiné způsoby organizace souborů

❖ B+ stromy:

- Vnitřní uzly obsahují sekvenci $link_0, key_0, link_1, key_1, ..., link_n, key_n, link_{n+1}$, kde $key_i < key_{i+1}$ pro $0 \leq i < n$. Hledáme-li záznam s klíčem k, pokračujeme $link_0$, je-li $k < key_0$; jinak $link_i$, $1 \leq i \leq n$, je-li $key_{i-1} \leq k < key_i$; jinak užijeme key_{n+1}.
- Listy mají podobnou strukturu. Je-li $key_i = k$ pro nějaké $0 \leq i \leq n$, $link_i$ odkazuje na hledaný záznam. Jinak hledaný záznam neexistuje.
- Poslední odkaz $link_{n+1}$ v listech je užit k odkazu na následující listový uzel pro urychlení lineárního průchodu indexovanými daty.



Jiné způsoby organizace souborů

❖ B+ stromy:

- Strom zůstává výškově vyvážený.
- Limity zaplnění pro uzly s m odkazy (tedy klíči key_0 až key_{m-2}): sólo kořen 1 až $m-1$, kořen 2 až m, vnitřní uzel $\lceil m/2 \rceil$ až m, list $\lceil m/2 \rceil - 1$ až $m-1$.
- Vkládá se na listové úrovni. Dojde-li k přeplnění, list se rozštěpí a přidá se nový odkaz do nadřazeného vnitřního uzlu. Při přeplnění se pokračuje směrem ke kořeni. Nakonec může být přidán nový kořen.
- Ruší se od listové úrovně. Při nenaplnění minimální kapacity, pokus o přerozdělení mezi sourozenci (potomky předka uzlu, ve kterém se ruší odkaz). Nestačí-li, sourozenci se spojí a ruší se jeden odkaz na nadřazené úrovni. Nutno upravit klíče. Rušení může pokračovat směrem ke kořeni. Nakonec může jedna úroveň ubýt.

❖ B+ stromy a jejich různé varianty jsou použity pro popis diskového prostoru přiděleného souborům v různých souborových systémech:

- XFS, JFS, ZFS, Btrfs, APFS, ReFS, ...,
- omezená analogie v podobě tzv. stromů extentů v ext4, podobně i v NTFS.

Otevření souboru pro čtení

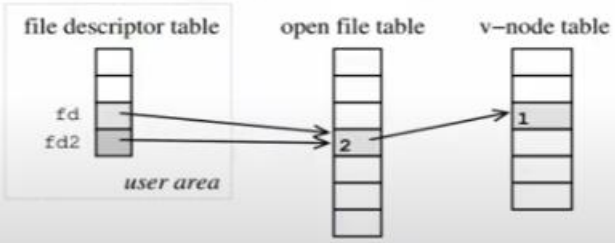
```
fd = open("/dir/file", O_WRONLY | O_CREAT | O_EXCL);
```

- V případě, že soubor ještě nebyl otevřen:
  1. Vyhodnotí cestu a nalezne číslo i-uzlu: postupně načítá i-uzly adresářů a obsah těchto adresářů, aby se dostal k číslům i-uzlů pod-adresářů či hledaného souboru – čísla i-uzlů pro některá jména mohou samozřejmě být ve speciálních vyrovnávacích pamětech, tzv. d-entry cache.
  2. V systémové tabulce aktivních i-uzlů vyhradí novou položku a načte do ní i-uzel. Vzniká rozšířená paměťová kopie i-uzlu: v-uzel.
  3. V systémové tabulce otevřených souborů vyhradí novou položku a naplní ji:
     - odkazem na položku tabulky v-uzlů,
     - režimem otevření,
     - pozicí v souboru (0),
     - čítačem počtu referencí na tuto položku (1).
  4. V poli deskriptorů souborů v záznamu o procesu v jádře nebo v tzv. uživatelské oblasti procesu vyhradí novou položku (první volná) a naplní ji odkazem na položku v tabulce otevřených souborů.

Duplikace deskriptoru souboru

```
fd2 = dup(fd);
fd2 = dup2(fd,newfd);
```

- Postup při duplikaci deskriptoru:
  1. Kontrola platnosti `fd`.
  2. Kopíruje danou položku v tabulce deskriptorů do první volné položky (`dup`) nebo do zadané položky (`dup2`). Je-li deskriptor `newfd` otevřen, `dup2` ho automaticky uzavře.
  3. Zvýší počitadlo odkazů v odpovídající položce tabulky otevřených souborů.
  4. Funkce vrací index nové položky nebo -1 při chybě.



- Poznámka: Použití pro přesměrování stdin/stdout.

Adresářové soubory

- Adresáře se liší od běžných souborů:
  - vytváří se voláním `mkdir` (vytvoří položky . a ..),
  - mohou být otevřeny voláním `opendir`,
  - mohou být čteny voláním `readdir`,
  - mohou být uzavřeny voláním `closedir`,
  - modifikaci je možné provést pouze vytvářením a rušením souborů v adresáři (`creat`, `link`, `unlink`, ...).

- Poznámka: Adresáře nelze číst/zapisovat po bajtech!

- Příklad obsahu adresáře:

32577	.
2	..
2361782	Archiv
1058839	Mail
1661377	tmp

Blokové a znakové speciální soubory

- Představují rozhraní k blokovým/znakovým zařízením, buď fyzickým či virtuálním – disky, logické disky, terminály, myš, paměť, ...
  - Lze vytvořit pomocí `mknod`.
  - Normálně řeší přímo jádro či různí démoni (např. `udev`, `devd`).

- Jádro mapuje běžné souborové operace (`open`, `read`, ...) nad blokovými a znakovými speciálními soubory na odpovídající podprogramy tyto operace implementující nad příslušným zařízením prostřednictvím dvou tabulek:
  - tabulky znakových zařízení a
  - tabulky blokových zařízení.

- Zmíněné tabulky obsahují ukazatele na funkce implementující příslušné operace v ovladačích příslušných zařízení.

- Ovladač (*device driver*) je sada podprogramů pro řízení určitého *typu* zařízení.