**Řízení vnější paměti**

**K čemu složí vnější paměť**

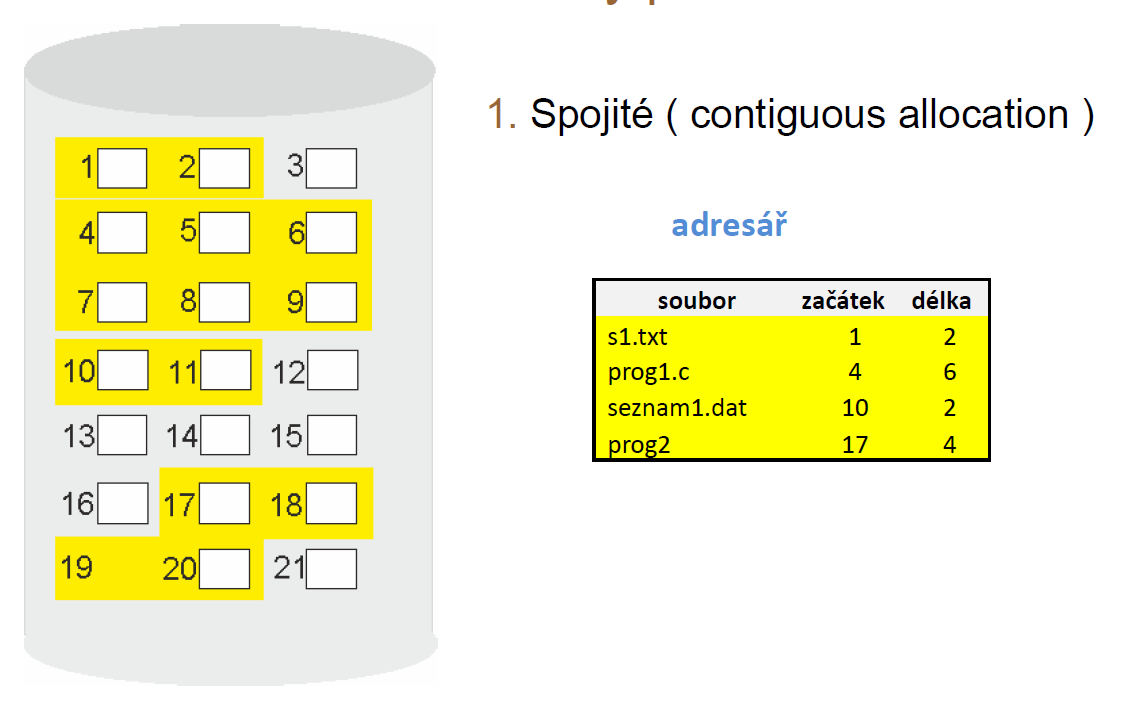
* představuje v architektuře počítače paměť určenou k trvalému ukládání informací (programů a dat)
* její obsah se vypnutím počítače neztrácí
* K vnější paměti nemá procesor počítače zpravidla přímý přístup

**Charakteristika HDD**

* Pevný disk
* je elektromechanické zařízení pro záznam a čtení adresovatelných dat, paměť o velké kapacitě a s pomalejším přístupem než operační paměť RAM
* používá se k dočasnému nebo trvalému uchovávání většího množství dat
* Ukládání a čtení probíhá pomocí magnetické indukce

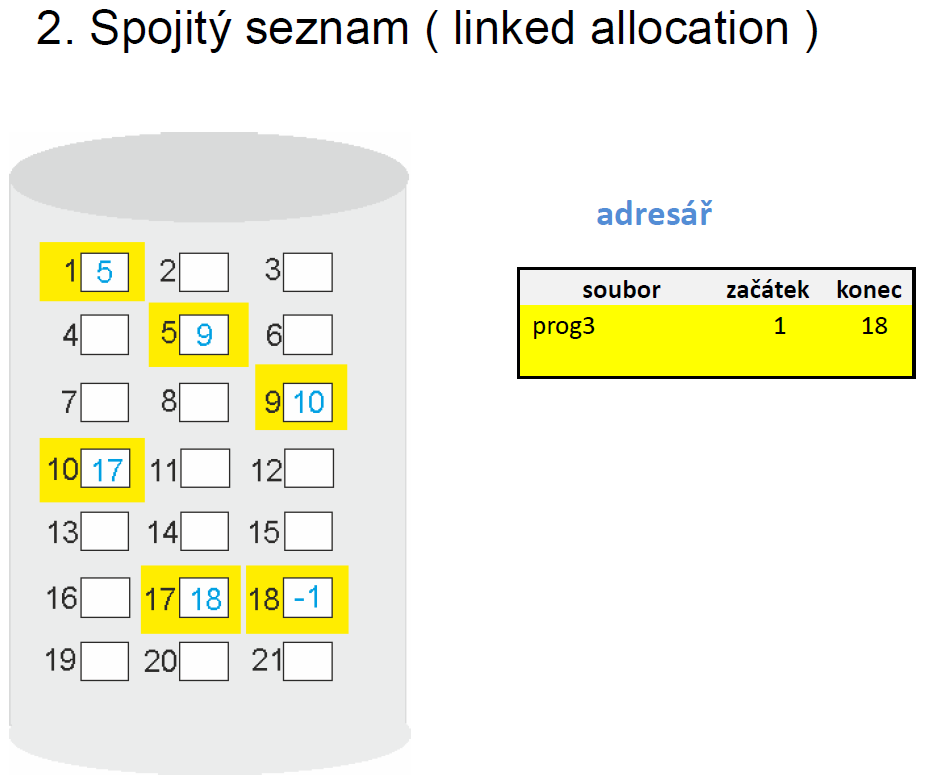
Metody přídělování místa na disku

Spojité přídělování



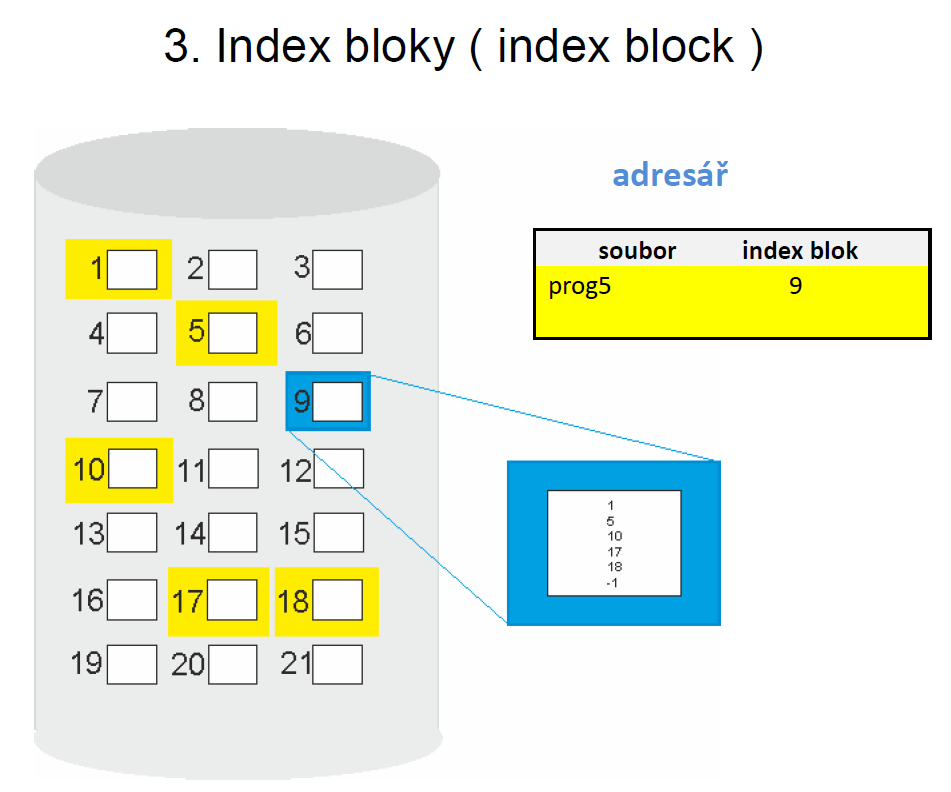
* Místo musí být v kuse volné -> sytém musí dopředu znát velikost souboru
* Pokud nenajde místo je nutné provést setřásání paměti -> dá dokupy soubor (naskládá) je tak aby bylo možné uložit soubor
* Přístup je možný jak sekvenčně (postupně – od začátku na konec) tak i přímým přístupem
* Minimální pohyb hlaviček (rychlejší procházení souborem)
* Při rozšíření (zvětšení souborů) 4 metody umístění souborů
  + First fit
  + Last fit
  + Best fit
  + Worst fit

Spojitý seznam



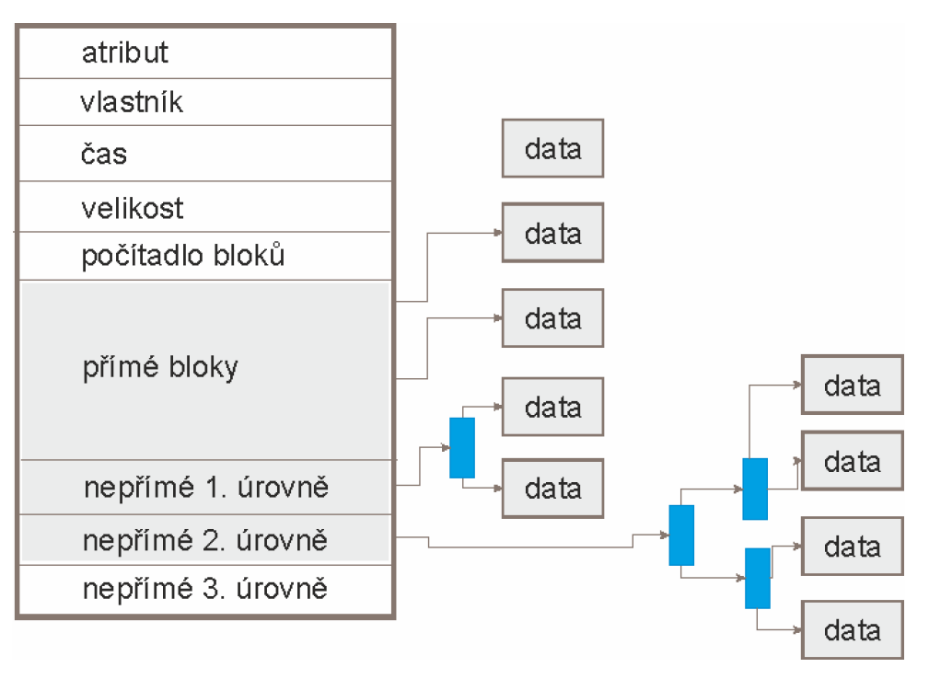
* Odstraňuje nutnost souvislého prostoru (může být různě po disku)
* Snaha OS aby to bylo nejblíž k sobě (minimalizace pohybu hlaviček)
* Stačí znát pouze adresu 1.bloku ty následující adresy jsou schovány v konkrétním bloku
* Pouze sekvenční přístup
* Potlačuje externí fragmentace (vnitřní stále zůstává)
  + Vnější - jev, kdy alokační bloky dat (tzv. clustery) jednoho souboru nejsou uloženy na médiu souvisle a toto roztroušení jejich částí na různých místech datového média pak může způsobovat zpomalování operací (čtení, zápis) při přístupu k těmto datům. Odstranění problému vnější fragmentace se označuje jako defragmentace.
  + Vnitřní - stav kdy zůstávají nevyužity části clusterů. Tomu lze zabránit subalokací clusterů (angl. block suballocation) kterou podporují mnohé současné souborové systémy.
* -1 v bloku označuje konec souboru

Indexová alokace



* Indexy všech bloků patřících souborů jsou umístěny pohromadě -> v tzv. indexovém bloku
* Každý soubor má svůj indexový blok
* Je možný jak přímý tak sekvenční přístup
* Při otevření souboru se indexový blok nahraje do OP
* Složitější na realizaci
* Využíváno v unixových systémech (iuzel-inode)
  + iuzel
    - neobsahuje jméno souborů, které zná adresářový soubor
    - obsahuje metadata (např. přístupová práva) a přímé nebo nepřímé odkazy
* je možná interní fragmentace
  + nevyužití celého bloku
  + nevyužití posledního alokačního bloku
* potlačení externí fragmentace
* snaha o co nejmenší indexový blok -> jak ale udržet seznam všech bloků systému?

1. Spojová struktura
   * 1 či více bloků za sebou
   * Na konci odkaz
   * PŘ.: blok velikosti 2kB, každý ukazatel zabere 4B -> ukazuje na 512 míst -> 512 \*2 = 1024kB
2. Víceúřovňový index
   * 1 hlavní indexový blok -> obsahuje pouze seznam ukazatelů na další indexové bloky
   * PŘ.: blok velikost 2kB ukazatel = 4B, 15 ukazatelů (12x prímý)
3. Kombinace předchozích



* + - * Blok velikosti 2048 B
      * Ukazatel 4 B
      * Celkem ukazatelů 15(12x přímý)
      * přímé: 12 \* 2048 = 24kB
      * 1. nepřímé: 512 \* 2048 = 1MB
      * 2. nepřímé: 512\*512 \* 2048 = 512MB
      * 3. nepřímé: 512\*512\*512\*2048=256GB

**Plánovací metody přístupu na disk**

* Čas přístupu na disk se dělí na 3 části

1. Seek
   * Přesun hlavy na požadovanou stopu či cilindr a k tomu způsobuje největší zpomalení
2. Latency
   * Otočení disku na začátek požadovaného sektoru
3. Transfer
   * Přesun dat z nebo na disk

* U víceúlohových systémů přicházejí požadavky na disk rychleji než je možné je vyřizovat

FCFS

* [Obrazek](ODKAZY/fcfs%20(2).jpg)
* First come first serve
* Nejjednodušší, jednoduchý na naprogramování
* Vyřizuje požadavky tak jak přichází
* Není optimální -> pomalý

SSTF

* [Obrazek](ODKAZY/sstf.jpg)
* Shortest seek time first
* Naplánován je požadavek s nejmenším relativním pohybem hlavy
* Nerovnoměrná obsluha -> není optimální
* Rychlá metoda (pro krátké vzdálenosti)

SCAN

* [Obrazek](ODKAZY/scan.jpg)
* Pohyb hlav jedním směrem a následně druhým
* Pokud je požadavek před hlavou, zpracuje ho, pokud přijde až za hlavu musí počkat až pojede zpět

C-SCAN

* [Obrazek](ODKAZY/cscan.jpg)
* Obsluhuje jenom v jednom směru
* Před hlavičkou je obslouží, za ní musí počkat až se zrotuje
* Přesun se nepočítá (z 0 do 200)

LOOK

* [obrazek](ODKAZY/look.jpg)
* Na rozdíl od scan nejezdí až do konce

C-LOOK

* [obrazek](ODKAZY/clook.jpg)

**HDD vs SSD**

SSD

* Solid-state drive
* neobsahuje pohyblivé součásti. Data ukládá na flash paměť
* VÝHODY
  + Tišší
  + Rychlejší
  + Úspornější
  + Mechanicky odolnější
* NEVÝHODY
  + Menší kapacita
  + Větší náklady

Defragmentace

* SSD
  + Sekvenční čtení SSD jednotek činí fragmentaci zanedbatelnou. Defragmentace by dalšími zápisy jen působila únavu NAND buněk s omezenou životností.
* HDD
  + Soubory, především veliké, se na HDD při častých zápisech průběžně fragmentují. Pravidelná defragmentace je potřeba pro dosažení optimálních výkonů.