

# Organizácia a spracovanie súborov

## Fyzické vlastnosti externých médií

### a) Páska

- neadresovateľnosť záznamov
- sekvenčný pohyb => sekvenčný prístup
- buď čítanie alebo zápis
- nie je permanentný pohyb
- nízka cena na GB
- dlhá životnosť dát
- vysoká kapacita
- vhodné na dlhodobé zálohovanie dát

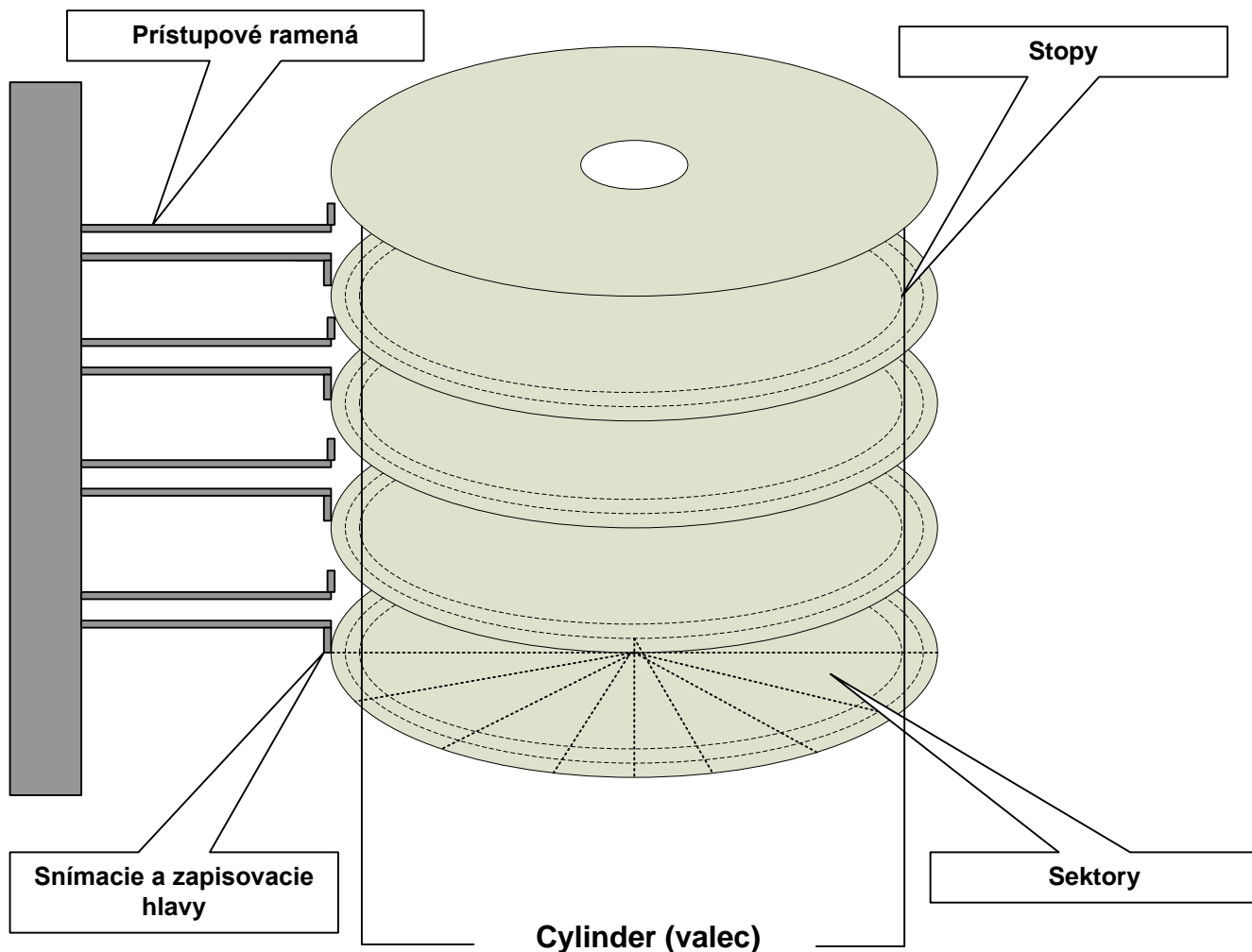
### b) SSD disk (Solid-state drive)

- neobsahuje pohyblivé časti (menej citlivý na otrasy)
- rovnaké prístupové doby k dátam
- nízka spotreba
- žiadna hlučnosť
- vyššia cena voči HDD za GB
- nižšia životnosť
- nižšia kapacita

### c) Pevný disk (HDD)

- adresovateľnosť (disková adresa): valec/hlava/sektor
- priamy prístup k adrese
- možnosť striedať čítanie a zápis
- permanentný pohyb => bez medzier
- vysoké kapacity
- nerovnomerná rýchlosť prístupu k dátam závislá aj na aktuálnom natočení disku
- vysoká životnosť

### Diskový zväzok (štylizovaný)



### Spoločné vlastnosti:

- operácie nad dátami možné iba v operačnej pamäti => prenos
- prenos (vrátane sprístupnenia) rádovo pomalší ako operácie

### Vlastnosti HDD:

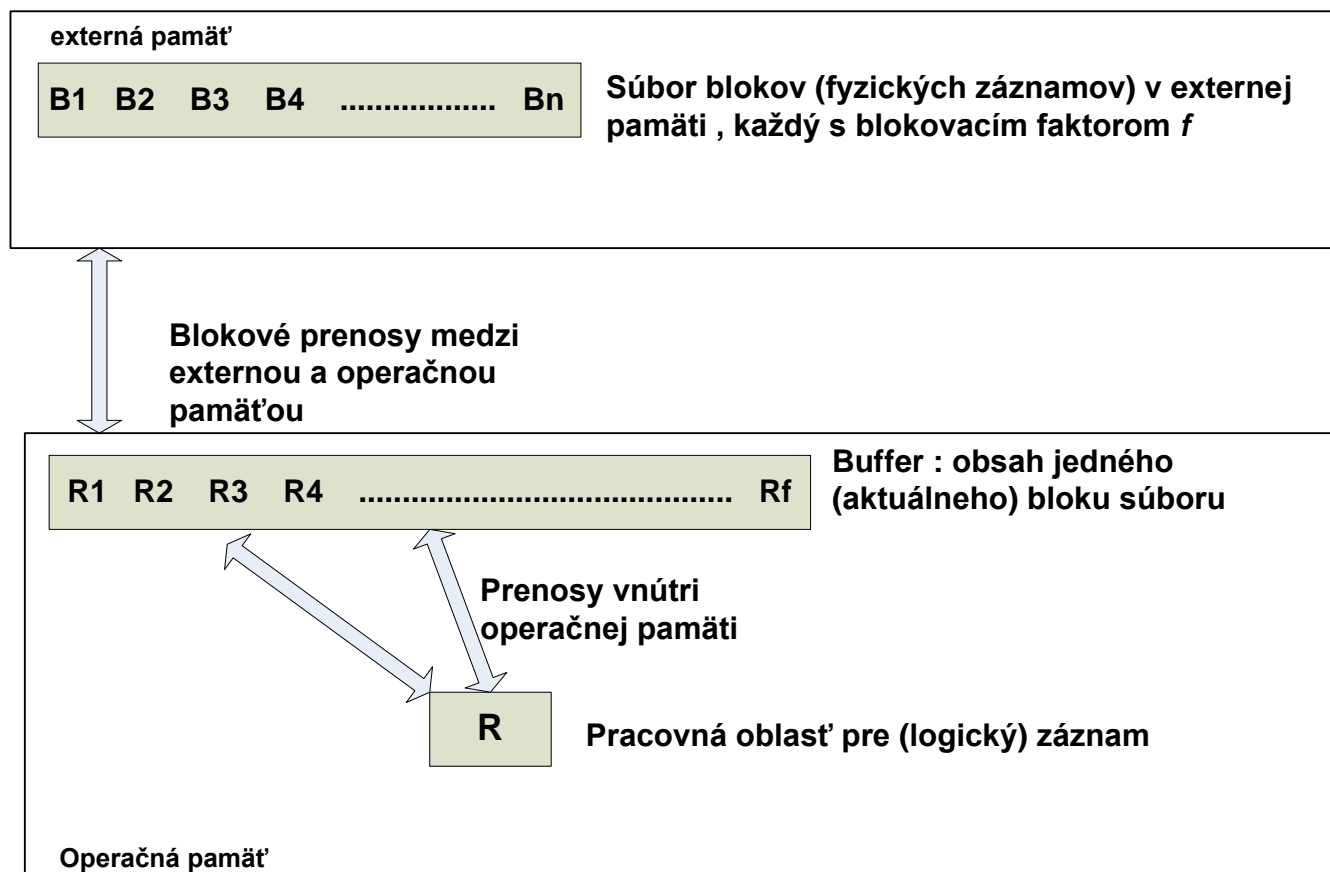
- doba (bezkontextového) prenosu súvislého bloku dát málo závislá od veľkosti bloku => výhodný prenos väčších blokov => blokovanie a bufferovanie :
  - záznamy na disku združené do blokov, počet záznamov v bloku je blokovaci faktor  $f$  (výhodné, aby veľkosť bloku = veľkosť sektora)
  - v operačnej pamäti vyhradený buffer (alebo viac) veľkosti bloku

Na HDD trvá dlhú dobu pokým sa rotačná platňa a hlavička dostanú na správnu polohu. Následne je možné prečítať/zapísať dáta.

**Prečítanie/zapísanie jediného bajtu trvá takmer rovnako dlho ako**

Materiál slúži výlučne pre študentov FRI ŽU, nie je dovolené ho upravovať, prípadne ďalej šíriť.

prečítanie/zapísanie najmensej alokovateľnej jednotky (clusteru). Nie je preto výhodné čítať dáta po malých množstvách, ale prispôbiť množstvo prečítaných dát najmensej alokovateľnej jednotke (clusteru). Zo súboru sú čítané/zapisované binárne dáta v blokoch, pričom blok obsahuje taký počet záznamov, aby sa celý zmestil na jeden cluster.



**Organizácia súboru:** sekvenčná, indexovaná, hierarchická, hešovacia...

**Spracovanie súboru** (možný prístup k záznamom):

- sekvenčný (podľa nasledovníka), priamy (podľa kľúča, podľa poradia - indexu)
- pre jednu organizáciu možné viacere druhy spracovania

# SÚBORY S PRIAMYM PRÍSTUPOM

## (VŠEOBECNE O ICH ORGANIZÁCIÍ)

- Vhodné pamäťové médium : HDD, SSD
- Priamy prístup = prístup podľa kľúča => implementácia tabuľky na disku
- Operácie:
  - Čítaj
  - Zapiš
  - Zruš – len na konci, súbor je možné skrátiť len od konca
  - Modifikuj – zmena dát uložených v súbore

**Model diskovej pamäti:** množina adresovateľných blokov pevnej veľkosti

**Súbor:** množina blokov  $B_0, B_1, B_2, \dots, B_N$

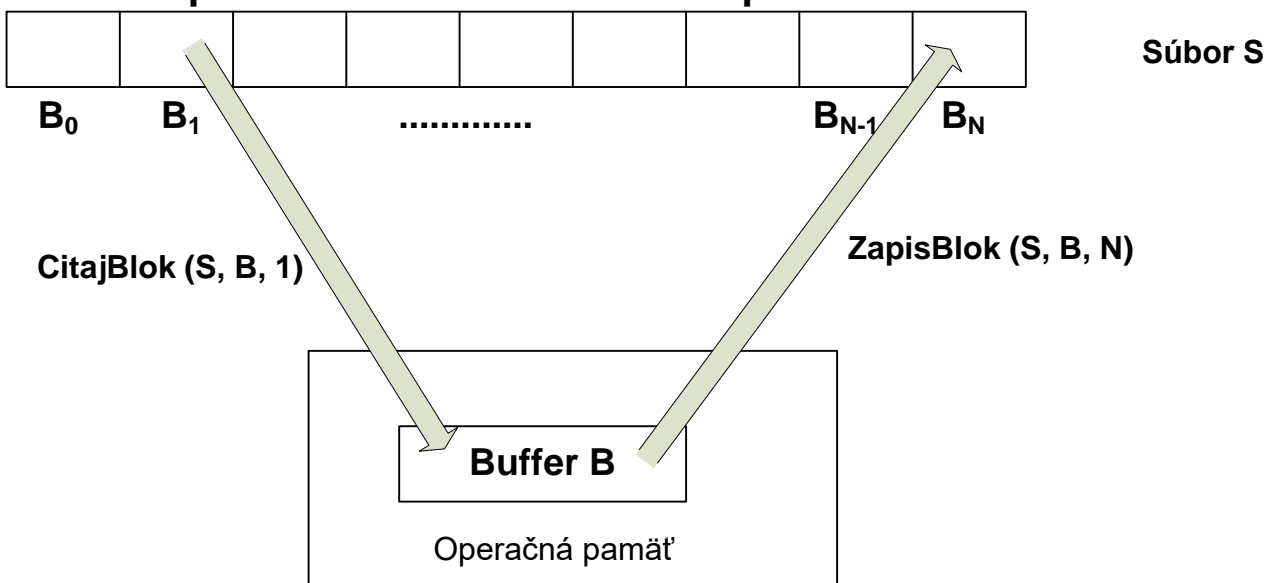
**Adresa záznamu :** { číslo bloku; posunutie v rámci bloku }

**Predpoklad:** sú vo všeobecnosti implementované operácie blokových prenosov:

**CitajBlok (Subor, Buffer (binárne dáta), PorCisloBloku (miesto v súbore))**

**ZapisBlok (Subor, Buffer (binárne dáta), PorCisloBloku (miesto v súbore))**

**Praktická realizácia je založená na binárnom súbore, kde je možné čítať a zapisovať dáta na ľubovoľnú pozíciu.**



## **Blok B<sub>0</sub> : riadiaci blok (head, label) súboru**

- informácie charakterizujúce celý súbor (iný formát ako ostatné)

### **Fakty o práci s diskovou pamäťou:**

- diskový priestor sa alokuje po blokoch, nie po bytoch,
- doba 1 diskového prenosu odpovedá asi  $10^6$  interných inštrukcií,
- doba sprístupnenia bloku rovnaká, ako doba sprístupnenia bytu,
- kritérium efektívnosti spracovania: počet blokových prenosov !!!
- vyplatí sa takmer akýkoľvek počet interných operácií, ako to ušetrí jeden blokový prenos.

### **Poznámka:**

Ak nie sú povolené referencie na záznamy súboru, tak je to súbor s nefixovanými záznamami (záznamy možno premiestňovať - typické pre utriedené verzie).

V súbore s fixovanými (pinned) záznamami sú povolené referencie na záznam (nemožno ich premiestňovať) alebo na blok (možno premiestňovať záznamy v rámci bloku). Fixované referencie na blok využívané častejšie ako fixované referencie na záznam.

### **Pridelovanie blokov na disku:**

#### **a) Dynamicky**

- ak implementované *NovyBlok*, *ZrusBlok*

#### **b) Staticky**

- na začiatku pevný počet „prázdnych“
- zavedenie evidencie prázdnych (zreťazenie alebo indikátor pre každý záznam)
- procedúry pridel', vráť

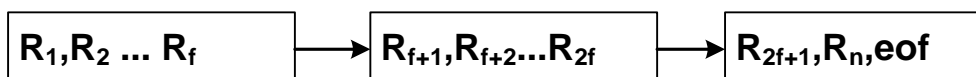
Adresovanie je na disku možné realizovať dvoma spôsobmi. Bud' vieme index bloku a na základe veľkosti bloku vieme stanoviť miesto, kde daný blok začína. Prípadne vieme presné adresné miesto, kde blok na disku začína. Pri mnohých štruktúrach je možné použiť obidva spôsoby adresácie. Pri niektorých iba jediný (napr. lineráne hešovanie umožňuje adresovať iba pomocou indexu).

**Všetky bloky jediného súboru majú rovnakú veľkosť!**

# Sekvenčný súbor

## Organizácia:

- záznamy sú uložené za sebou v ľubovoľnom poradí
- bloky môžu byť alokované
  - v súvislej oblasti (možné na páske, HDD, SSD)
  - v nesúvislej oblasti - bloky zreťazené adresami blokov (možné na HDD a SSD)
- záznamy nemusia mať definovanú kľúčovú položku (nedá sa využiť pre prístup)



## Spracovanie sekvenčného súboru na HDD a SSD (na páske s obmedzeniami):

a.)

Čítanie zo súboru (existujúceho, starého, vstupného)

*Otvor pre čítanie (S-súbor) ...* súbor prístupný pre spracovanie

*Čítaj (S, R) ...* prenos aktuálneho záznamu do premennej *R*;

*Prepíš (S, R) ...* aktuálny záznam sa prepíše záznamom *R*

*Zatvor(S) ...* súbor neprístupný pre spracovanie

*Čítaj(S, R, i) ...* možné obohatenie prístupu - podľa indexu (ako pole)

(musí byť implementované: *ČítajBlok (S, buff, b)* - blok *b* do *buff*)

Realizácia:

○ *ČítajBlok (S, buff, i div f + 1)*

○ *R = buff [Ri mod f]*

b.) Rozširovanie súboru (existujúceho, starého)

*Otvor pre rozšírenie (S) ...* súbor prístupný pre zápis na koniec

*Append (S, R) ...* zapíš *R* ako posledný v súbore

*Zatvor(S) ...* súbor neprístupný pre spracovanie

c.) Zápis do súboru (nového, neexistujúceho, výstupného)

*Otvor pre zápis (S) ...* vytvorenie nového súboru *S*; súbor prístupný pre zápis

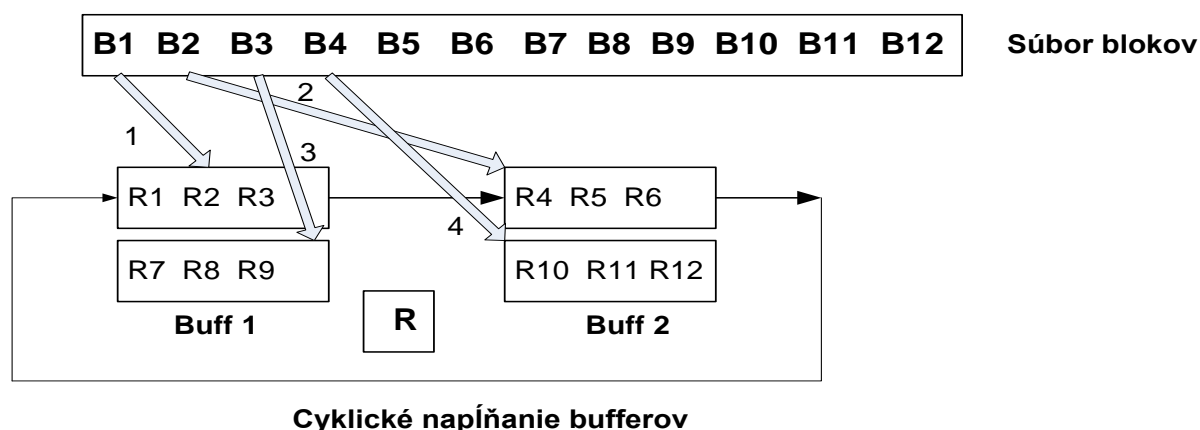
*Zapíš (S, R) ...* zapíš *R* ako ďalší (resp. prvý)

*Zatvor (S) ...* súbor neprístupný pre spracovanie

### Anticipované čítanie zo vstupného sekvenčného súboru:

- využíva, že môžeme čítať iba *nasledujúci* záznam a preto:
  - o po spracovaní posledného záznamu z buffera zahájí automaticky prenos ďalšieho bloku
  - o pri použití vhodnej veľkosti bloku a príp. viacerých bufferov možno vylúčiť čakanie na prenos po spracovaní obsahu buffera

### Príklad pre 2 buffery a $f = 3$ :



L

Logický pohľad (užívateľa) na operáciu Čítaj: R = ďalší záznam  
Fyzický pohľad (implementačný) pri opakovanom čítaní:

Otvor: Buff 1 = B1; Buff 2 = B2;

Čítaj : R = R1

Čítaj : R = R2

Čítaj : R = R3, zaháj prenos Buff 1 = B3

Čítaj : R = R4

Čítaj : R = R5

Čítaj : R = R6, zaháj prenos Buff 2 = B4

Čítaj : R = R7

... ..

### Kumulovaný zápis do výstupného sekvenčného súboru:

- analogicky ako anticipované čítanie: prenos bloku do súboru až po naakumulovaní buffera; počas prenosu zápis do ďalšieho buffera

**Dôsledky:**

- **Sekvenčný prístup môže poskytovať ďaleko rýchlejšie spracovanie v prípade, že chceme spracovať všetky záznamy súboru**
- **Výhodné ako alternatívny spôsob spracovania súborov, ktoré majú implementovaný aj priamy prístup podľa kľúča**