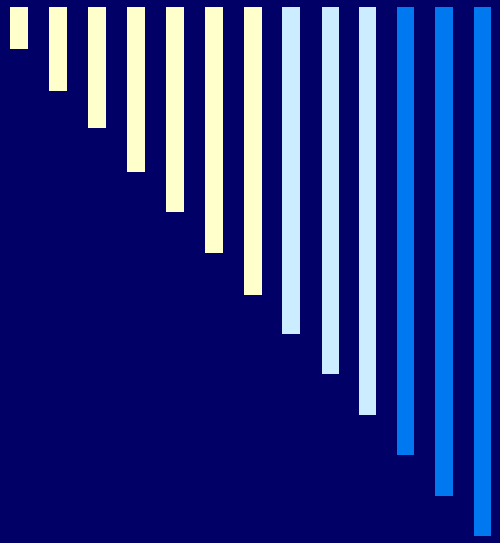
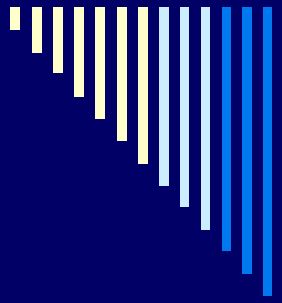


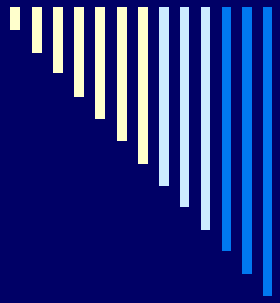
12.
Správa souborů

ZOS 2006, L. Pešička



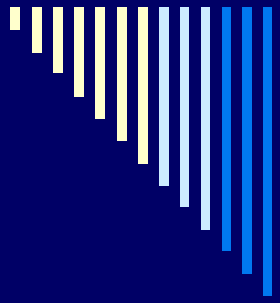
Informace – 2. zápočtový test

- Látka z přednášek do 9. týdne
  - Požadované znalosti
    - Meziprocesová komunikace
    - Synchronizace
    - Základ MM
    - Řešení konkrétních příkladů v BACI
- **12.12.2006** (úterý) od **18:30** v **EP130**



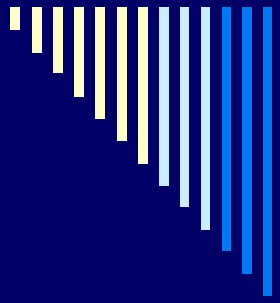
Souborové systémy

- potřeba aplikací trvale uchovávat data
- hlavní požadavky
  - možnost uložit velké množství dat
  - informace zachována i po ukončení procesu
  - data přístupná více procesům
- společné problémy při přístupu k zařízení
  - alokace prostoru na disku
  - pojmenování dat
  - ochrana dat před neoprávněným přístupem
  - zotavení po havárii (výpadek napájení)



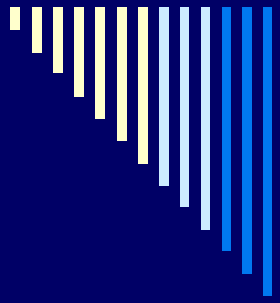
Soubor

- OS pro přístup k mediim poskytuje abstrakci od fyzických vlastností média – soubor
- soubor = pojmenovaná množina souvisejících informací
- souborový systém (file system, fs)
  - konvence pro ukládání a přístup k souborům
    - datové struktury a algoritmy
  - část OS, poskytuje mechanismus pro ukládání a přístup k datům, implementuje danou konvenci



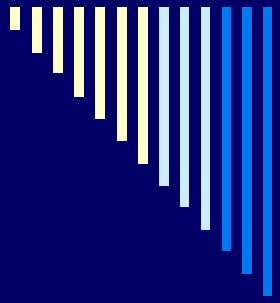
File system

- Současné OS – implementují více fs
  - kompatibilita (starší verze, ostatní OS)
- Windows XP
  - základní je NTFS
  - ostatní: FAT12, FAT16, FAT32, ISO 9660 (CD-ROM)
- Linux
  - ext2, ext3, ReiserFS, JFS, XFS
  - ostatní: FAT12 až 32, ISO 9660, Minix, VxFS, OS/2 HPFS, SysV fs, UFS, NTFS read-only



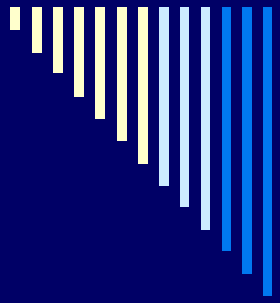
Historický vývoj

- první systémy
  - vstup – děrné štítky, výstup – tiskárna
  - soubor = množina děrných štítků
- později magnetické pásky
  - vstup i výstup – pásky
  - soubor = množina záznamů na magnetické pásce



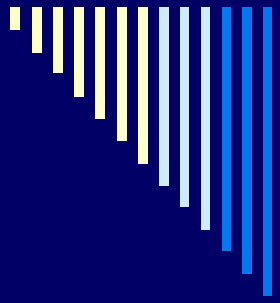
Historický vývoj - pokračování

- nyní – data na magnetických a optických discích
  - ISO 2382-4:1987
  - soubor – pojmenovaná množina záznamů, které lze zpracovávat jako celek
  - záznam – strukturovaný datový objekt tvořený konečným počtem pojmenovaných položek



Uživatelské rozhraní fs

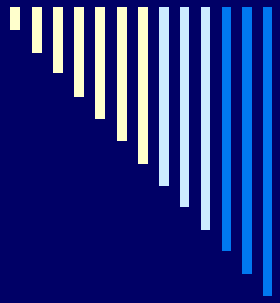
- vlastnosti fs z pohledu uživatele
  - konvence pro pojmenování souborů
  - vnitřní struktura souboru
  - typy souborů
  - způsob přístupu
  - atributy a přístupová práva
  - služby OS pro práci se soubory



Konvence pro pojmenování souborů

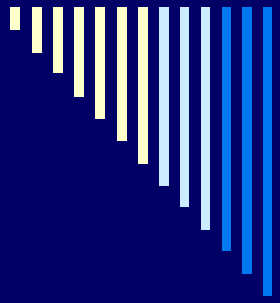
- vytvoření souboru – proces určuje jméno souboru
- různá pravidla pro vytváření jmen – různé OS
- Windows NT, XP x Unix a Linux

- rozlišuje systém malá a velká písmena?
  - Win32API nerozlišuje: ahoj, Ahoj, AHOJ stejná
  - UNIX rozlišuje: ahoj, Ahoj, AHOJ rozdílná jména



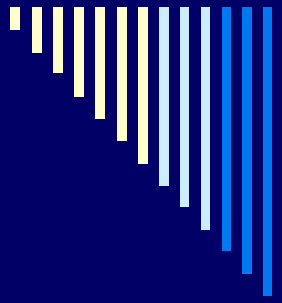
Pojmenování souborů

- jaká může být délka názvu souboru?
  - WinNT 256 znaků NTFS
  - UNIX obvykle alespoň 256 znaků (dle typu fs)
- množina znaků?
  - všechny běžné – názvy písmena a číslice
  - WinNT – znaková sada UNICODE
    - βετα – legální jméno souboru
  - Linux – všechny 8bitové znaky kromě / a char(0)



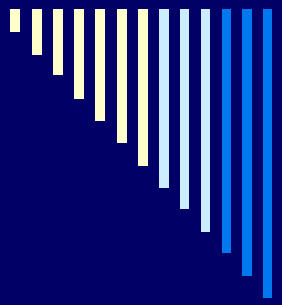
Pojmenování souborů

- přípony?
  - MS DOS – jméno souboru 8 znaků + 3 znaky přípona
  - WinNT, Unix – i více přípon
- další omezení?
  - WinNT – mezera nesmí být první a poslední znak



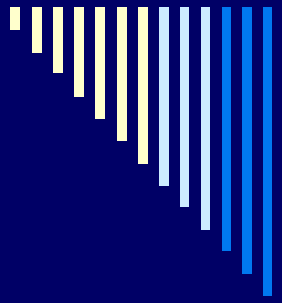
Typy souborů

- OS podporují více typů souborů
- obyčejné soubory – převážně dále rozebírány
  - data zapsaná aplikacemi
  - obvykle rozlišení textové x binární
  - textové - řádky textu ukončené CR (MAC), LF(UNIX), nebo CR+LF (MS DOS, Windows)
  - binární – všechny ostatní
  - OS rozumí struktuře spustitelných souborů



Typy souborů

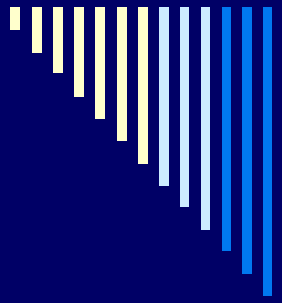
- adresáře
  - systémové soubory, udržují strukturu fs
- Linux , UNIX ještě:
  - znakové speciální soubory
  - blokové speciální soubory
    - rozhraní pro I/O zařízení, /dev/lp0 – tiskárna
  - pojmenované roury
    - pro komunikaci mezi procesy
  - symbolické odkazy



Vnitřní struktura (obyčejného) souboru

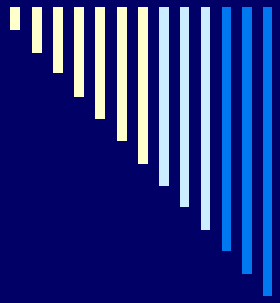
- 3 časté způsoby
  - nestrukturovaná posloupnost bytů
  - posloupnost záznamů
  - strom záznamů

- nestrukturovaná posloupnost bytů
  - OS obsah souboru nezajímá, interpretace je na aplikacích
  - maximální flexibilita
    - programy mohou strukturovat, jak chtějí



Vnitřní struktura (obyčejného) souboru – pokrač.

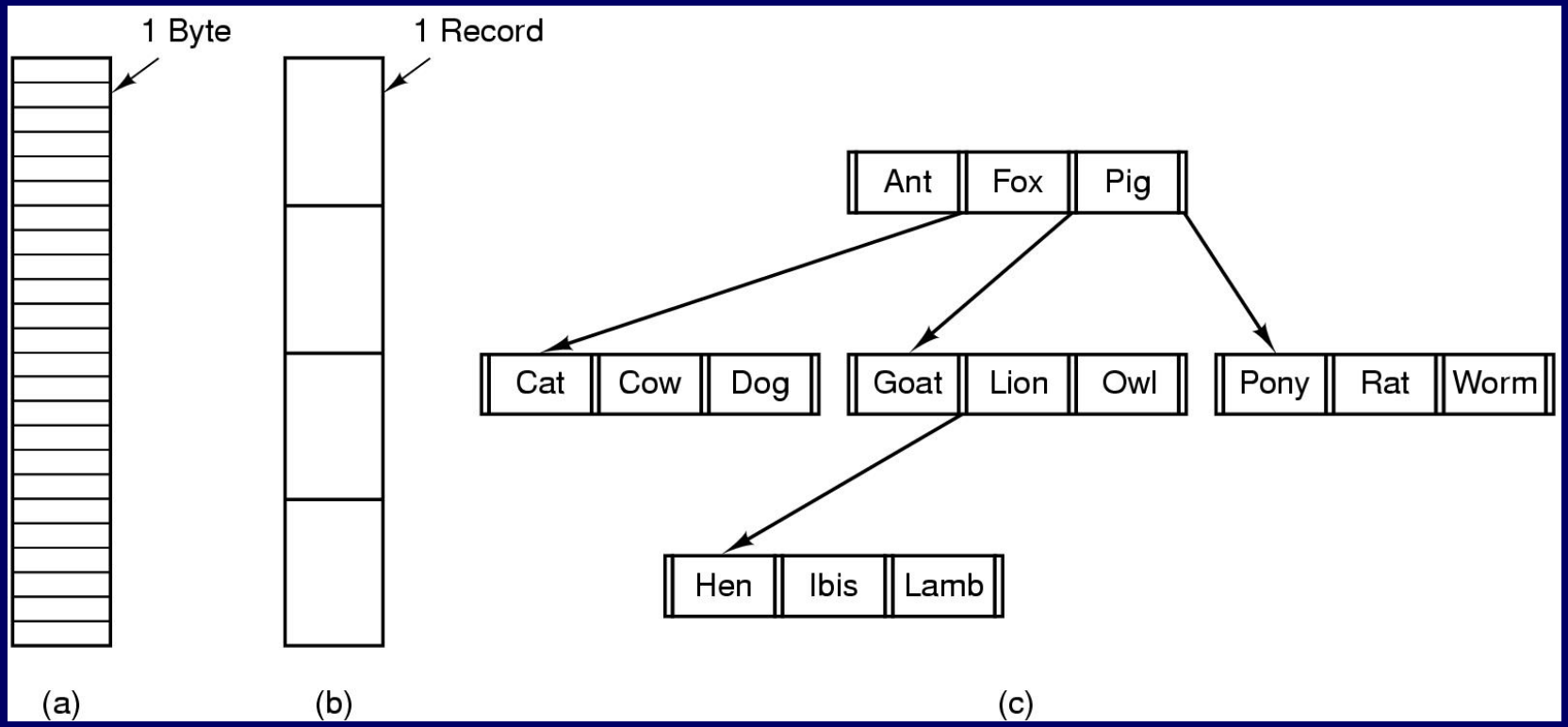
- posloupnost záznamů pevné délky
  - každý záznam má vnitřní strukturu
  - operace čtení –vrátí záznam, zápis – změní / přidá záznam
  - v historických systémech
  - záznamy 80 znaků obsahovaly obraz děrných štítků
  - v současných systémech se téměř nepoužívá

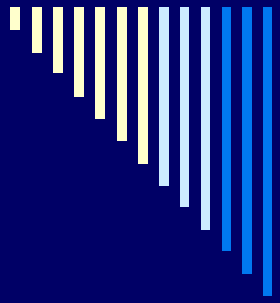


Vnitřní struktura (obyčejného) souboru – pokrač.

- strom záznamů
  - záznamy nemusejí mít stejnou délku
  - záznam obsahuje pole klíč (na pevné pozici v záznamu)
  - záznamy seřazeny podle klíče, aby bylo možné vyhledat záznam s požadovaným klíčem
  - mainframy pro komerční zpracování dat

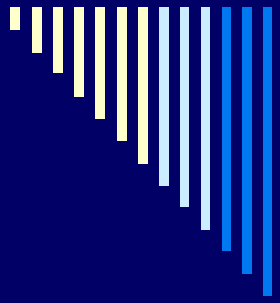
Posloupnost bajtů, záznamů, strom





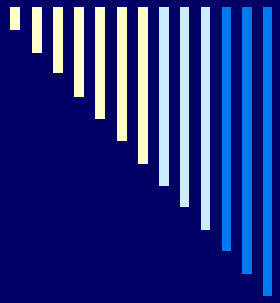
Způsob přístupu k souboru

- sekvenční přístup
  - procesy mohou číst data pouze v pořadí, v jakém jsou uloženy v souboru
  - tj. od prvního záznamu, nemohou přeskakovat
  - možnost přetočit a číst opět od začátku, rewind()
  - v prvních OS, kde data na magnetických páskách



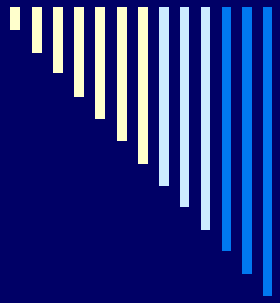
Způsob přístupu k souboru

- přímý přístup (random access file)
  - čtení v libovolném pořadí nebo podle klíče
  - přímý přístup je nutný např. pro databáze
  - určení začátku čtení
    - každá operace určuje pozici
    - OS udržuje pozici čtení / zápisu, novou pozici lze nastavit speciální operací „seek“



Způsob přístupu k souboru

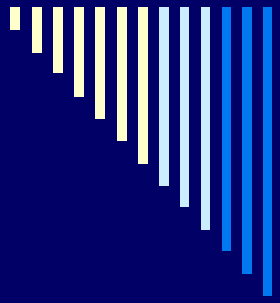
- v některých OS pro mainframy – při vytvoření souboru se určilo, zda je sekvenční nebo s přímým přístupem
  - OS mohl používat rozdílné strategie uložení souboru
- všechny současné OS – soubory s přímým přístupem



Atributy

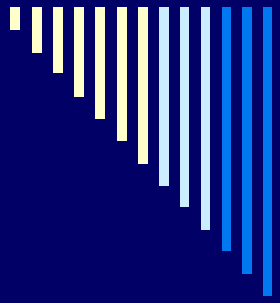
- informace sdružená se souborem
- některé atributy interpretuje OS, jiné systémové programy a aplikace
- významně se liší mezi jednotlivými OS

- ochrana souboru
  - kdo je vlastníkem, množina přístupových práv, heslo, ...



Atributy - pokračování

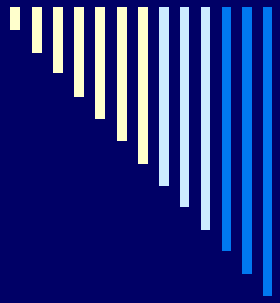
- příznaky
  - určují vlastnosti souboru
  - hidden – neobjeví se při výpisu
  - archive – soubor nebyl zálohován
  - temporary – soubor bude automaticky zrušen
  - read-only, text/binary, random access
- přístup k záznamu pomocí klíče
  - délka záznamu, pozice a délka klíče
- velikost, datum vytvoření, poslední modifikace, poslední přístup



Služby OS pro práci se soubory

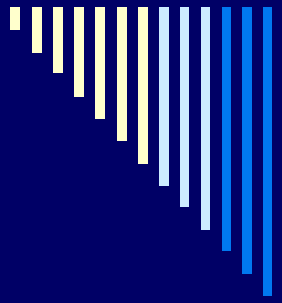
- většina současných – základní model dle UNIXu

- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Seek



Několik jednoduchých pravidel

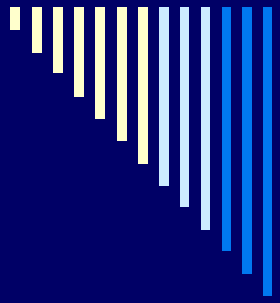
- veškerý I/O prováděn pouze pomocí souborů
  - obyčejné soubory – data, spustitelné programy
  - zařízení – disky, tiskárny
  - se všemi typy zacházení pomocí stejných služeb systému
- obyčejný soubor – uspořádaná posloupnost bytů
  - význam znají pouze programy, které s ním pracují
  - interní struktura souboru OS nezajímá
- jeden typ souboru – seznam souborů – adresář
  - adresář je také soubor
  - soubory a adresáře koncepčně umístěny v adresáři



Několik jednoduchých pravidel, poznámky

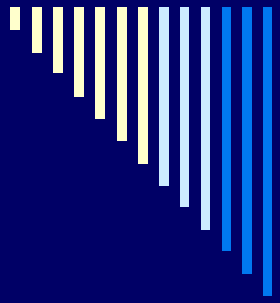
- speciální soubory – pro přístup k zařízením
  - DOS – PRN:, COM1:

- Poznámka – před příchodem UNIXu toto samozřejné nebylo
- většina systémů před UNIXem – samostatné služby pro čtení / zápis terminálu, zápis na tiskárnu, čtení / zápis do souboru
- mnoho systémů před i po UNIXu – mnoho různých druhů souborů s různou strukturou a metodami přístupu



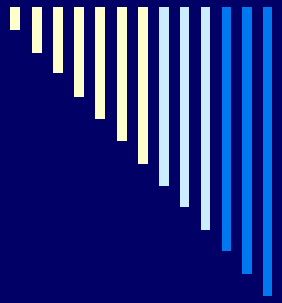
Poznámky

- systémy poskytovaly „více služeb“ x model podle UNIXu – podstatně menší složitost
- téměř všechny moderní systémy základní rysy modelu převzaly



Základní služby pro práci se soubory

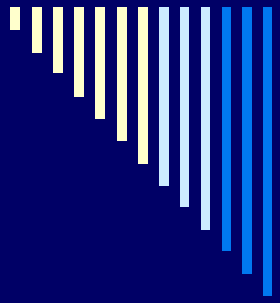
- otevření souboru
  - než s ním začneme pracovat
  - úspěšné – vrátí služba pro otevření souboru – popisovač souboru (file descriptor) – malé celé číslo
  - popisovač souboru používáme v dalších službách
    - čtení apod.



Základní služby pro práci se soubory

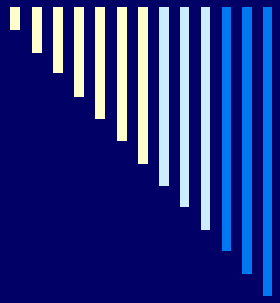
- otevření souboru:
- fd = open (jmeno, způsob)
- jméno – řetězec pojmenovávající soubor
- způsob – pouze pro čtení, zápis, obojí
- fd – vrácený popisovač souboru

- otevření souboru nalezne informace o souboru na disku a vytvoří pro soubor potřebné datové struktury
- popisovač souboru – index to tabulky souborů uvnitř OS



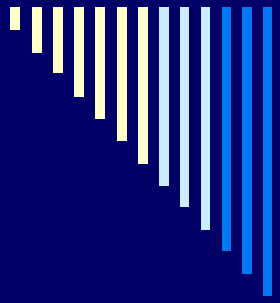
Základní služby pro práci se soubory

- vytvoření souboru:
- fd=creat(jméno, práva)
- vytvoří nový soubor s daným jménem a otevře pro zápis
- pokud soubor existoval – zkrátí na nulovou délku
- fd – vrácený popisovač souboru



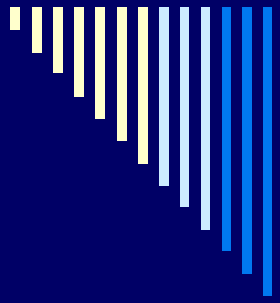
Základní služby pro práci se soubory

- operace čtení ze souboru:
- read(fd, buffer, počet_bytů)
- přečte počet_bytů ze souboru fd do bufferu
- může přečíst méně – zbývá v souboru méně
- přečte 0 bytů – konec souboru



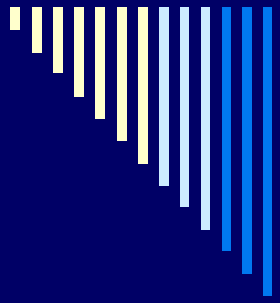
Základní služby pro práci se soubory

- operace zápisu do souboru
- write (fd, buffer, počet_bytů)
- význam parametrů jako u read
- uprostřed – přepíše, konec – prodlouží
- read i write vrací počet skutečně zpracovaných bytů
- read() a write() – jediné operace pro čtení a zápis
- samy o sobě poskytují sekvenční přístup k souboru



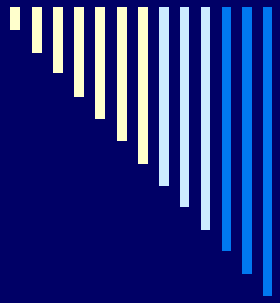
Základní služby pro práci se soubory

- nastavení pozice v souboru:
- lseek (fd, offset, odkud)
- nastaví offset příští čtené/zapisované slabiky souboru
- odkud
  - od začátku souboru
  - od konce souboru (záporný offset)
  - od aktuální pozice
- poskytuje přímý přístup k souboru



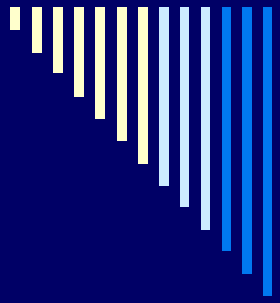
Základní služby pro práci se soubory

- zavření souboru
- close (fd)
- uvolní datové struktury alokované OS pro soubor



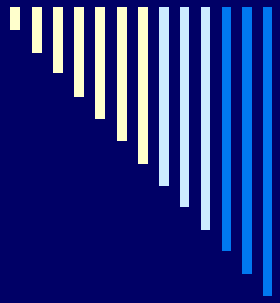
Příklad použití rozhraní – kopírování souboru

```
int src, dst, in;
src = open(”puvodni”, O_RDONLY);          /* otevreni zdrojoveho */
dst = creat(”novy”, MODE);                /* vytvoreni ciloveho */
while (1)
{
in = read(src, buffer, sizeof(buffer));    /* cteme */
if (in == 0)                              /* konec souboru? */
  {
     close(src);                          /* zavreme soubory */
     close(dst);
     return;                              /* ukončení */
  }
write(dst, buffer, in);                   /* zapiseme prectena data */
}
```



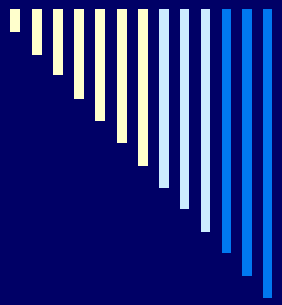
Další služby pro práci se soubory

- změna přístupových práv, zamykání, ...
- závislé na konkrétních mechanismech ochrany
- např. UNIX
  - zamykání fcntl (fd, cmd)
  - zjištění informací o souboru (typ, příst. práva, velikost)
  - stat (file_name, buf), fstat (fd, buf)



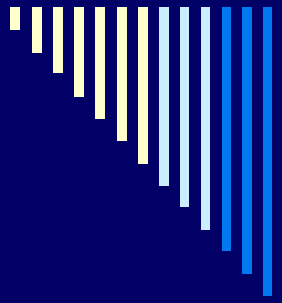
Paměťově mapované soubory

- někdy se může zdát open/read/write/close nepohodlné
- možnost mapování souboru do adresního prostoru procesu
- služby systému mmap(), munmap()
- mapovat je možné i jen část souboru



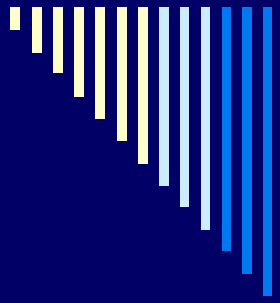
Paměťově mapované soubory - příklad

- délka stránky 4KB
- soubor délky 64KB
- chceme mapovat do adresního prostoru od 512KB
- 512 * 1024 = 524 288 .. od této adresy mapujeme
- 0 až 4KB souboru bude mapováno na 512KB – 516KB
- čtení z 524 288 čte byte 0 souboru atd.



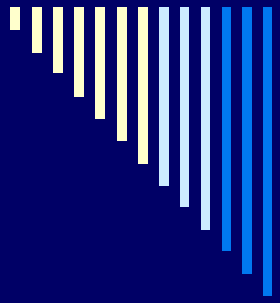
Implementace paměťově mapovaných souborů

- OS použije soubor jako odkládací prostor (swapping area) pro určenou část virtuálního adresního prostoru
- čtení / zápis na adr. 524 288 způsobí výpadek stránky
- do rámce se načte obsah první stránky souboru
- pokud je modifikovaná stránka vyhozena (nedostatek volných rámců), zapíše se do souboru
- po skončení práce se souborem se zapíší všechny modifikované stránky



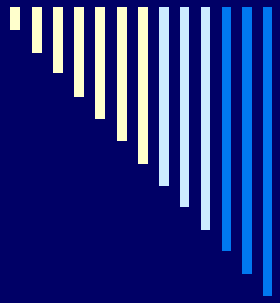
Problémy pam. map. souborů

- není známa přesná velikost souboru, nejmenší jednotka je stránka
- problém nekonzistence pohledů na soubor, pokud je zároveň mapován a zároveň se k němu přistupuje klasickým způsobem



Adresářová struktura

- jedna oblast (partition) obsahuje jeden fs
- fs – 2 součásti:
  - množina souborů, obsahujících data
  - adresářová struktura – udržuje informace o všech souborech v daném fs
- adresář překládá jméno souboru na informace o souboru (umístění, velikost, typ ...)



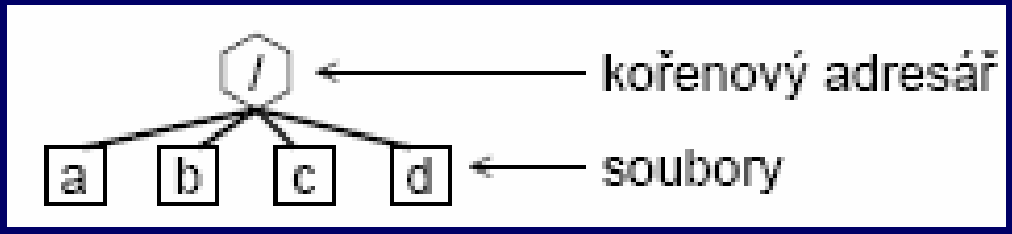
Základní požadavky na adresář

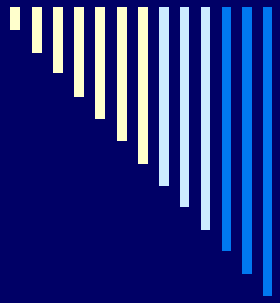
- procházení souborovým systémem (cd)
- výpis adresáře (ls)
- vytvoření a zrušení souboru
- přejmenování souboru

- dále schémata logické struktury adresářů
- odpovídá historickému vývoji OS

Schémata logické struktury adresářů

- jednoúrovňový adresář
- původní verze MS DOSu
- všechny soubory jsou v jediném adresáři
- všechny soubory musejí mít jedinečná jména
- problém zejména pokud více uživatelů

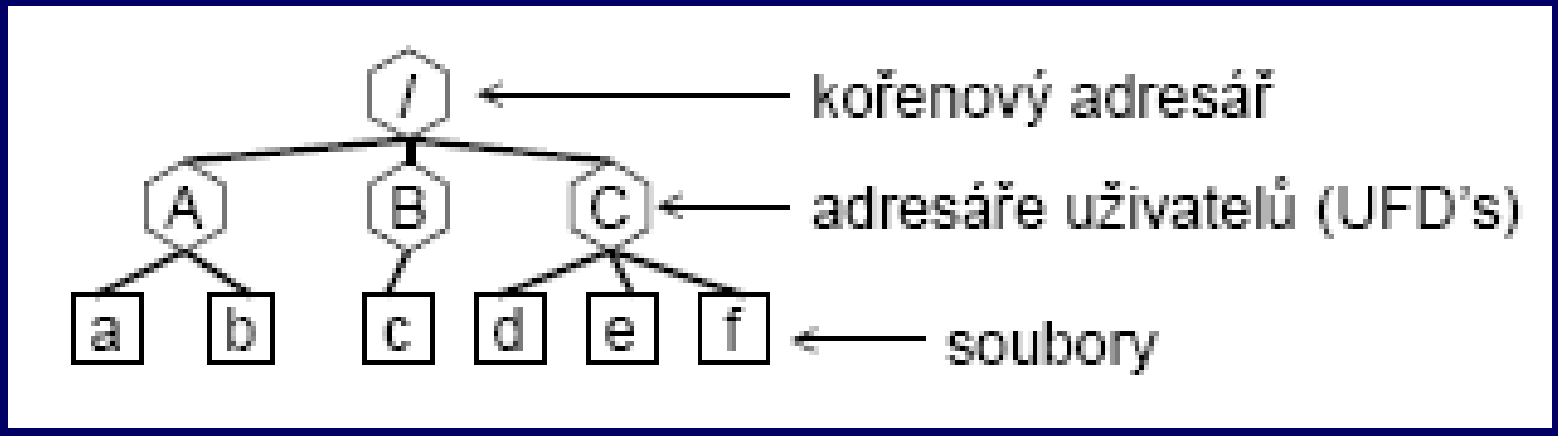


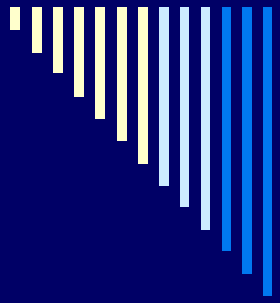


Dvouúrovňový adresář

- adresář pro každého uživatele (User File Directory, UFD)
- OS prohledává pouze UFD , nebo pokud specifikováno adresář jiného uživatele [user] file
- systémové příkazy – spustitelné soubory – speciální adresář
  - příkaz se hledá v adresáři uživatele
  - pokud zde není, vyhledá se v systémovém adresáři

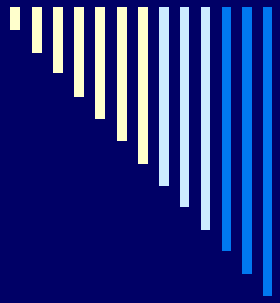
Dvouúrovňový adresář – pokr.





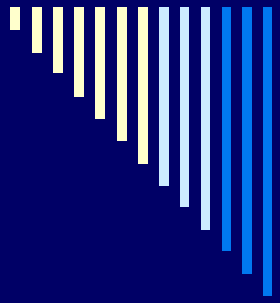
Adresářový strom

- zobecnění předchozího
- dnes nejčastější, MS DOS, Windows NT
- adresář – množina souborů a adresářů
- souborový systém začíná kořenovým adresářem „/“
- MS DOS „\“, znak / se používal pro volby
- cesta k souboru – jméno v open, creat
  - absolutní
  - relativní



Cesta k souboru

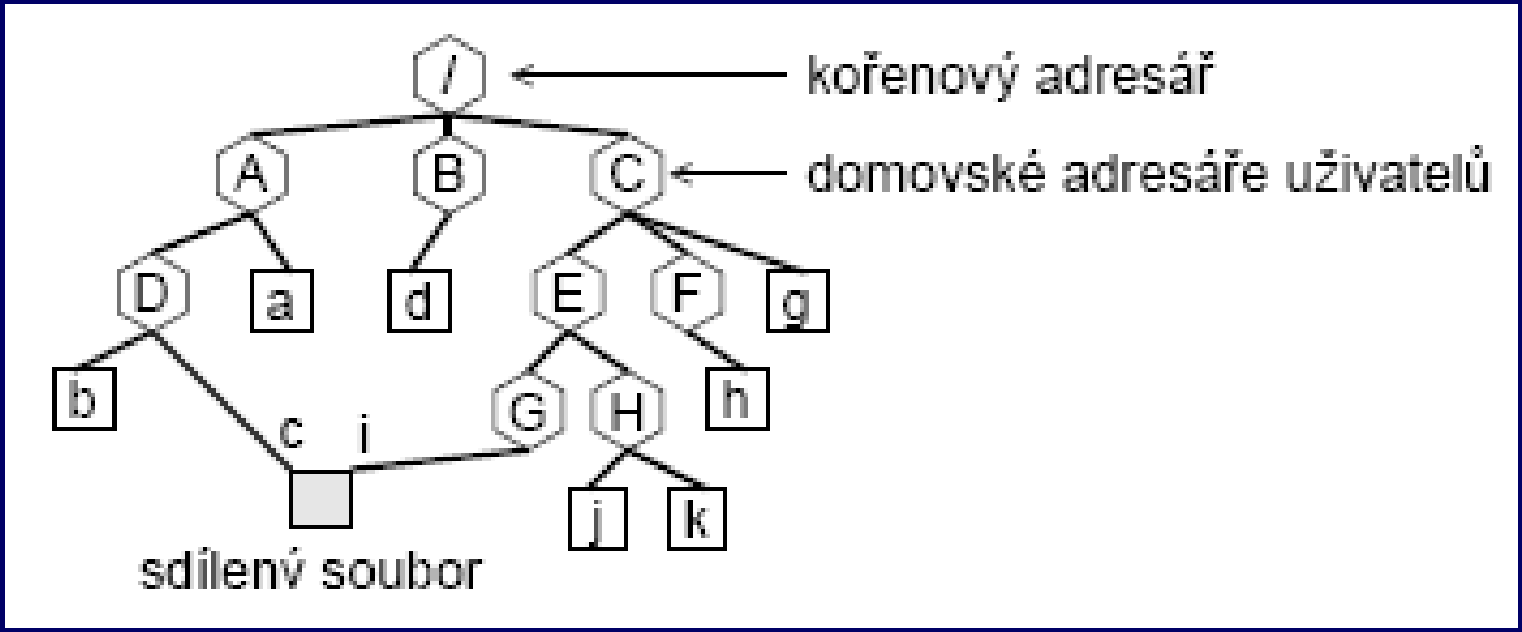
- absolutní
  - kořenový adresář a adresáře, kudy je třeba projít, název souboru
  - oddělovače adresářů – znak „/“
  - např. /home/user/data/v1/data12.txt
- relativní
  - aplikace většinou přistupují k souborům v jednom adresáři
  - defaultní prefix = pracovní adresář
  - cesta nezačíná znakem /
  - př. data12.txt data/v1/data12.txt

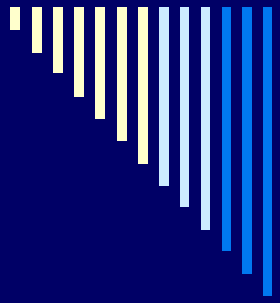


Acyklický graf adresářů

- např. týmová spolupráce na určitém projektu
- sdílení společného souboru nebo podadresáře
  - stejný soubor (adr.) může být viděn ve dvou různých adresářích
- flexibilnější než strom, komplikovanější
- rušení souborů / adresářů – kdy můžeme zrušit?
  - se souborem sdružen počet odkazů na soubor z adresářů
  - každé remove sníží o 1, 0 = není odkazován
- jak zajistit aby byl graf acyklický?
  - algoritmy pro zjištění, drahé pro fs

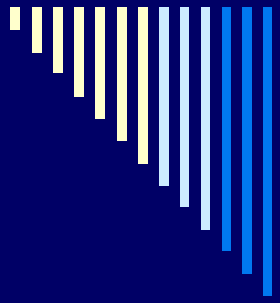
Acyklický graf adresářů





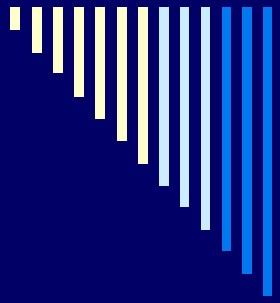
Obecný graf adresářů

- obtížné zajistit, aby graf byl acyklický
- prohledávání grafu
  - omezení počtu prošlých adresářů (Linux)
- rušení souboru
  - pokud cyklus, může být počet odkazů > 0 i když je soubor již není přístupný
  - garbage collection – projít celý fs, označit všechny přístupné soubory; zrušit nepřístupné; drahé, zřídka používáno



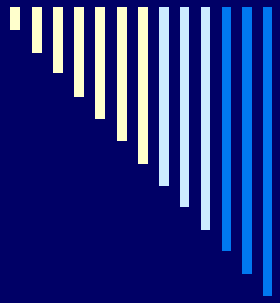
Nejčastější použití

- nejčastěji adresářový strom (MS DOS)
- UNIX od původních verzí acyklický graf
  hard links – sdílení pouze souborů – nemohou vzniknout cykly



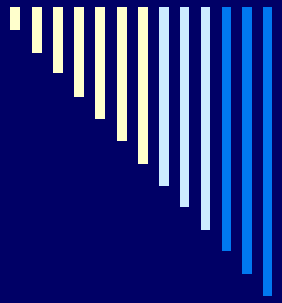
Základní služby pro práci s adresáři

- téměř všechny systémy dle UNIXu
- pracovní adresář – služby:
- chdir (adresář)
  - nastavení pracovního adresáře
- getcwd (buffer, počet_znaků)
  - zjištění pracovního adresáře



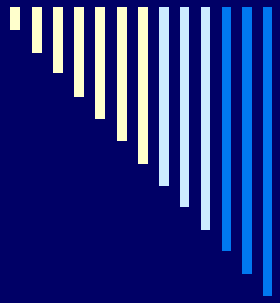
Práce s adresářovou strukturou

- vytváření a rušení adresářů
  - mkdir (adresář, přístupová_práva)
  - rmdir (adresář) – musí být prázdný
- zrušení souboru
  - remove (jméno_souboru)
- přejmenování souboru
  - rename (jméno_souboru, nové_jméno)
  - provádí také přesun mezi adresáři



Práce s adresářovou strukturou

- čtení adresářů – UNIX / POSIX
  - DIRp = opendir (adresář)
    - otevře adresář
  - položka = readdir (DIRp)
    - čte jednotlivé položky adresáře
  - closedir (DIRp)
    - zavře adresář
  - stat (jméno_souboru, statbuf)
    - info o souboru, viz man 2 stat
- př. DOS: findfirst / findnext

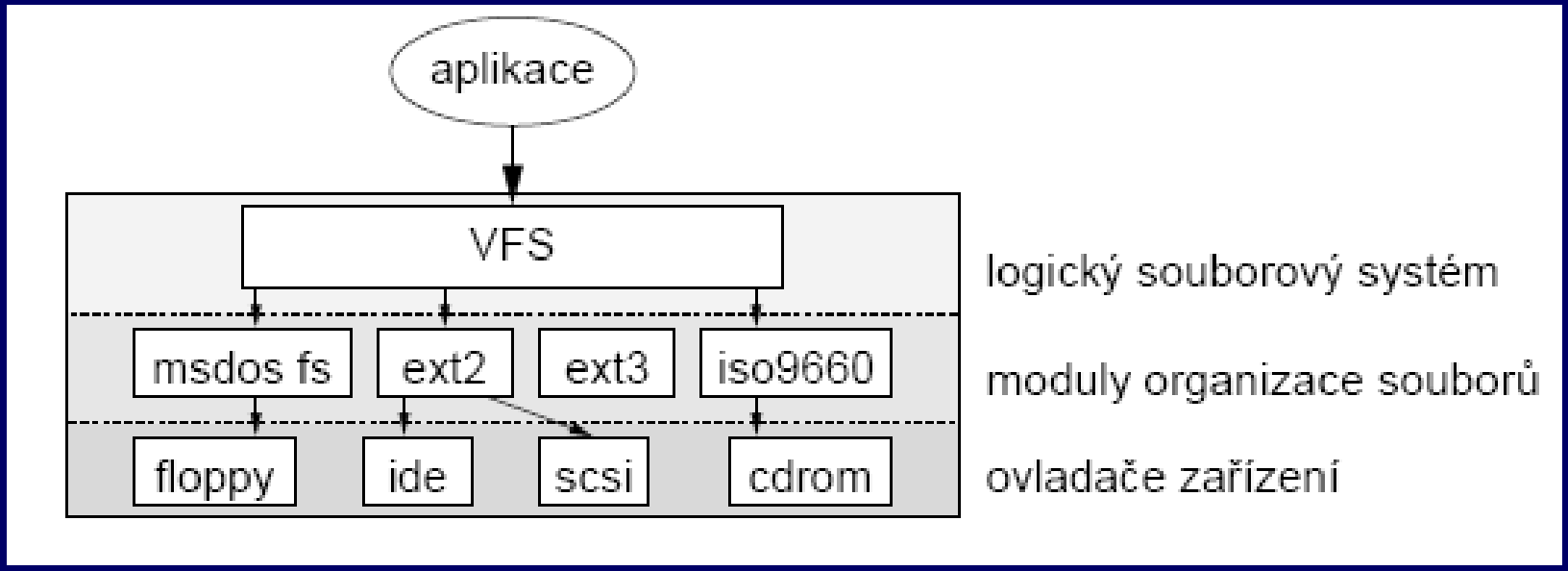


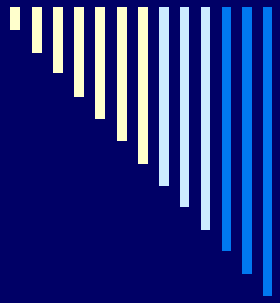
Implementace souborových systémů

- algoritmy a datové struktury pro implementaci fs
- jak bude fs vypadat z pohledu uživatele

- 3 vrstvová implementace souborů

3 vrstvy implementace

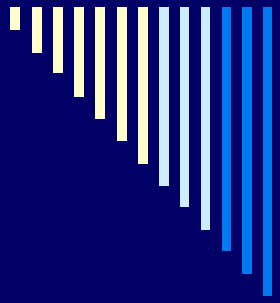




1. Logický souborový systém

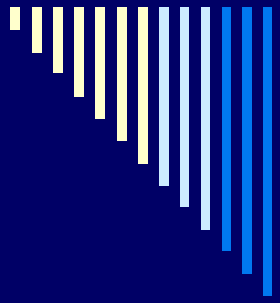
- VFS (virtual file system)
  - volán aplikacemi
  - rozhraní s moduly organizace souborů
  - kód společný pro všechny typy fs

  - převádí jméno souboru na info o souboru
  - udržuje info o otevřeném souboru
    - mód – čtení / zápis
    - pozice
  - ochrana a bezpečnost (ověřování příst.práv)



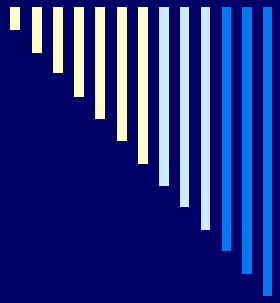
2. Modul organizace souborů

- implementuje konkrétní fs
- čte / zapisuje datové bloky souboru
  - datové bloky souboru bloky 0 .. N-1
  - převod číslo bloku na diskovou adresu
  - volání ovladače pro čtení-zápis bloku
- správa volného prostoru, alokace volných bloků
- údržba datových struktur fs
  - defragmentace



3. Ovladače zařízení

- nejnižší úroveň
- zpracovává požadavky
  - př. přečti logický blok 1234 ze zařízení 3

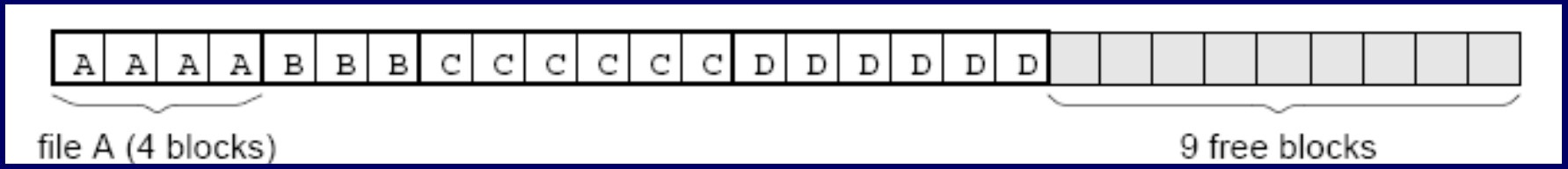


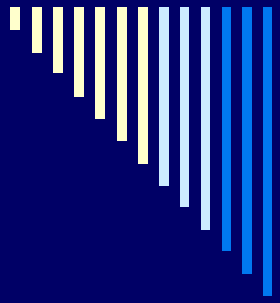
Implementace souborů

- příklady realizace
- nejdůležitějšíotázka
  - které datové bloky patří kterému souboru
- file system
  - umístěn v nějaké oblasti (disk partition)
  - bloky oblasti očíslovány od 0

Kontinuální alokace

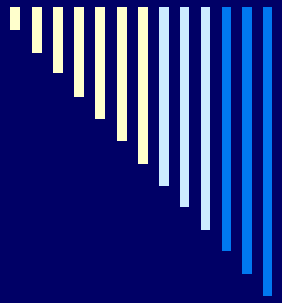
- nejjednodušší
  - soubor jako posloupnost diskových bloků
- př. blok 1KB, soubor A 4KB – 4 bloky





Kontinuální alokace

- soubor
  - číslo prvního bloku
  - celkový počet bloků souboru (např. v adresáři)
- rychlé čtení – bloky za sebou
- problém
  - soubory vznikají, zanikají, mění velikost
  - na začátku sériový zápis
  - po zaplnění využít volné místo po zrušených souborech
  - vhodná díra – potřebujeme info o konečné délce souboru

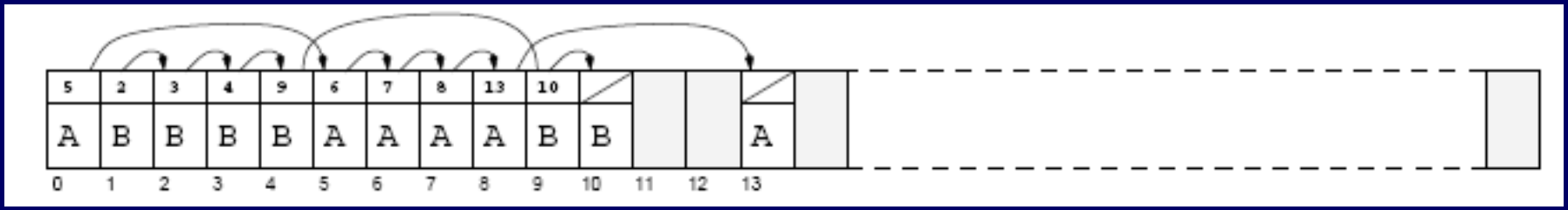


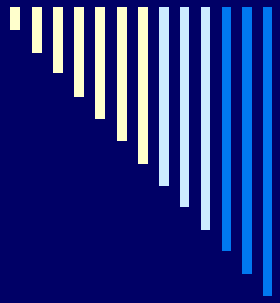
Kontinuální alokace

- read-only a write-once média
- např. základní verze ISO 9660 pro CD ROM

Seznam diskových bloků

- diskové bloky svázané do seznamu
  - nebude vnější fragmentace
- na začátku bloku – odkaz na další blok souboru
- zbytek bloku – obsahuje data souboru
- přístup – stačí znát pouze číslo prvního bloku souboru (např. z adresáře)





Seznam diskových bloků

- sekvenční čtení – ok
- přímý přístup – simulován sekvenčním – pomalé

- velikost dat v bloku není mocnina 2
  - část zabrána odkazem na další soubor