Univerzita Hradec Králové
Fakulta informatiky a managementu

Souborové operace

Mgr. Josef Jan Horálek, Ph.D. & Ing. Tomáš Svoboda, Ph.D.

Souborové operace

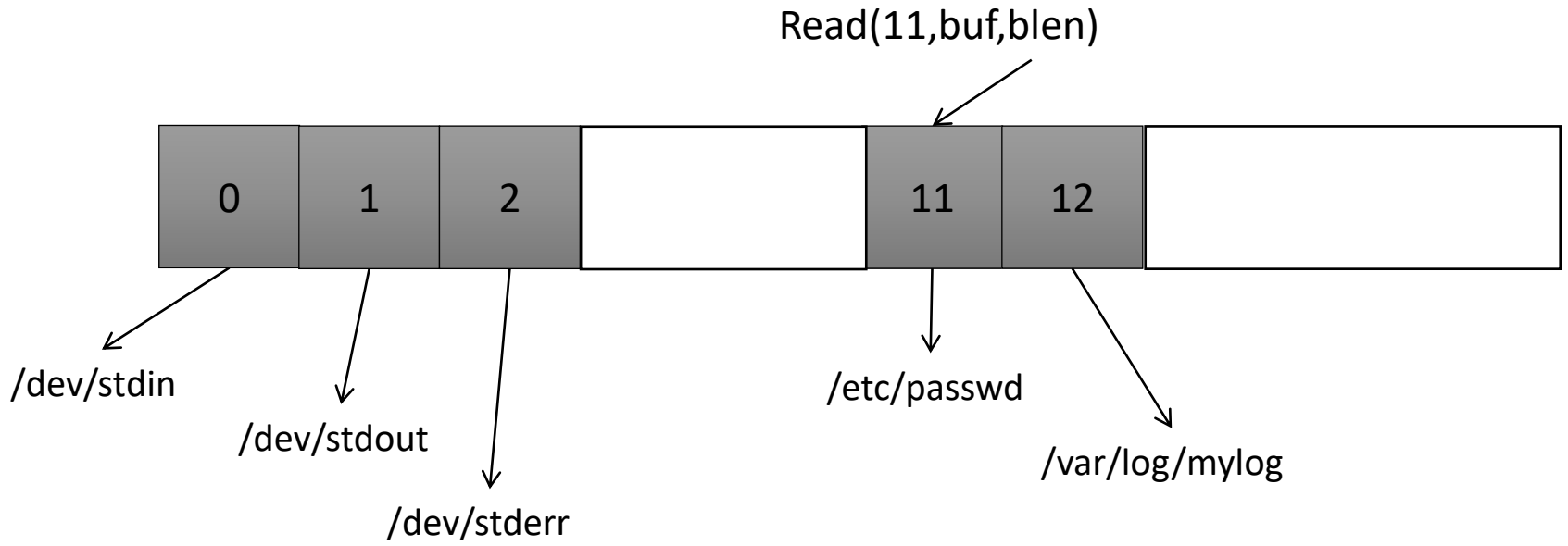
- V Unixových systémech a Linuxu platí, že skoro vše je SOUBOR
  - Častý výskyt deskriptorů souborů a souborových operací.
- Souborové operace umožňují využívat:
  - normální soubory a adresáře
  - speciální soubory (pevné a symbolické odkazy, roury a fronty, soubory řazení)
  - síťové sockety
  - další prostředky (inotify, kevent apod., složka /proc)

Deskriptor souborů (popisovač)

- Identifikován číslem (int).
- Používá se v rámci procesu pro přístup k otevřenému souboru.
- Nezáporné hodnoty pro platné deskriptory
  - -1 = neplaný deskriptor
- První tři patří standardním komunikačním kanálům
  - 0 pro standardní vstup;
  - 1 pro standardní výstup;
  - 2 pro chybový výstup;

Deskriptor souborů (popisovač)

= Vazba deskriptorů a otevřených souborů



Otevírání a zavírání souborů

- Otevření souboru:
  - jádro vytvoří potřebné datové struktury pro přístup do souboru
  - označí soubor jako otevřený daným procesem
  - proces získá deskriptor pro odkazování na soubor
  - po použití nutno soubor vždy zavřít

Otevírací operace

- Způsob otevírání závisí na druhu souboru.
- Základní operace pro otevírání souborů (normálních a speciálních) je funkce **open()**.
- Otevíraný soubor určujeme cestou (relativní nebo absolutní) a můžeme použít až dva další parametry ovlivňující způsob otevření.

```
int file_descriptor = open(„soubor.txt“, O_RDONLY);
```

Parametry

- **Režim přístupu**:
- O_RDONLY – jen ke čtení
- O_WRONLY – jen k zápisu
- O_RDWR – k zápisu i čtení
- **Režim práce se souborem**:
- O_APPEND – zápis na konec souboru
- O_NONBLOCK – neblokující operace
- O_ASYNC – asynchronní režim – lze využít pouze u rour, terminálů, soketů a FIFO!
- O_SYNC – synchronní režim

- **Režim chování funkce**:
- O_CREAT – vytvoření souboru
- O_EXCL – ochrana existujícího souboru – volání open selže pokud soubor již existuje
- O_NOLINK – otevření symbolického odkazu místo cílového souboru
- O_TRUNC – zkrácení souboru na nulovou délku

Zavření otevřeného souboru

- Každý otevřený soubor nutno zavřít.
- Proč my a ne jádro?
  - Každý otevřený soubor vyžaduje alokované prostředky v jádře (hlavně fyzickou paměť)
  - Otevřené soubory se dědí do podprocesů
  - Nebezpečí nechtěného zápisu do nezavřeného souboru
  - Počet najednou procesem otevřených souborů je omezený

Zavření otevřeného souboru

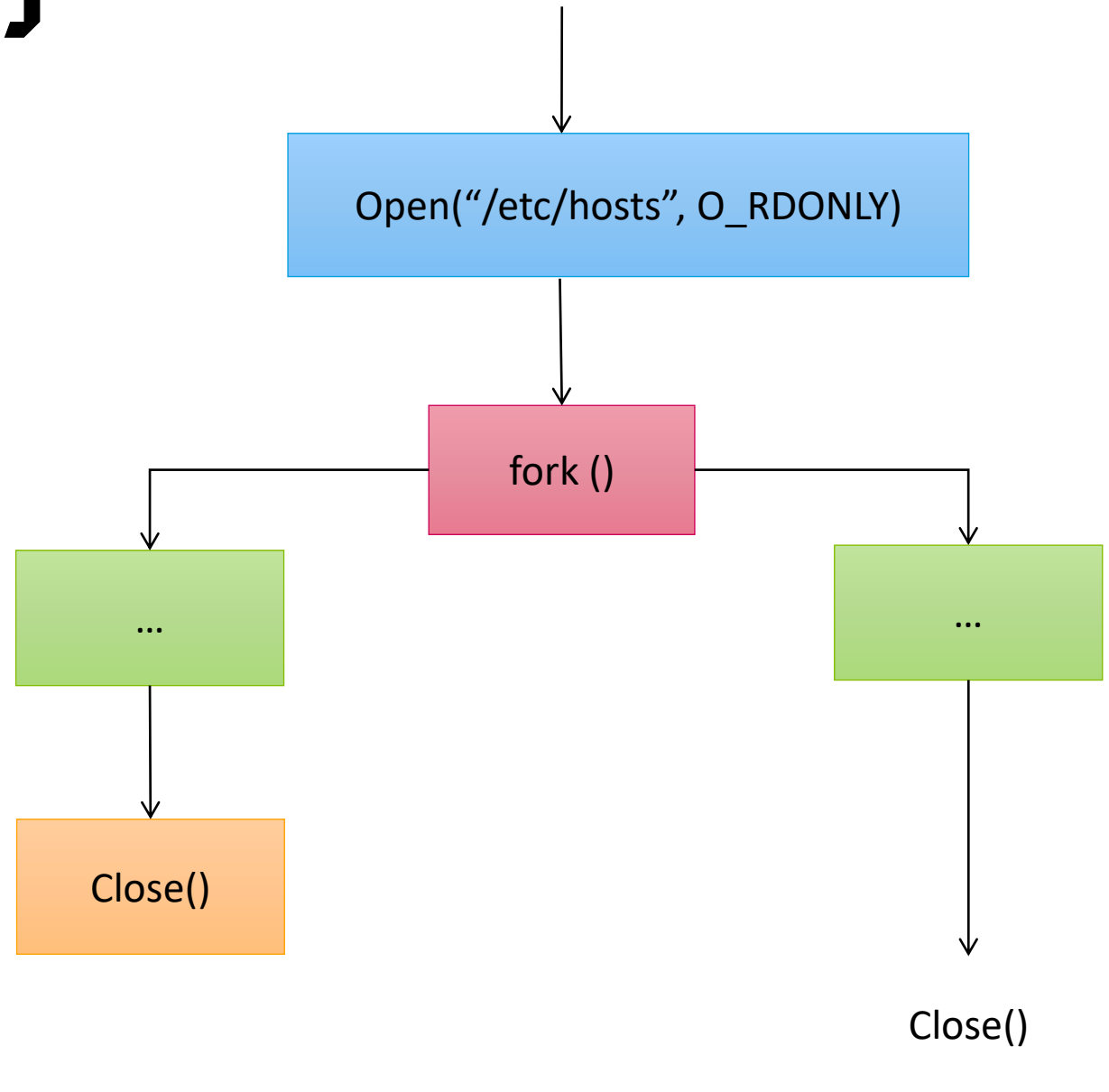
- Zavření souboru je jednoduché, volá se funkce **close(int)** – ta vyvolá stejnojmenné systémové volání.
- Jako parametr se uvádí platný deskriptor souboru.
- Funkce může selhat:
  - přerušena signálem
  - problém na zařízení nebo souborovém systému

Zavření otevřeného souboru

- Soubory otevřené v podprocesech:
  - **close()** neznamená skutečné uzavření souboru – jen sníží počet referencí na něj,
  - ovladač v jádře pak neprovede následné činnosti – ty provede až když soubor zavře poslední proces.

FIM UHK

Zavření otevřeného souboru



Otevření souboru (1 reference)

Duplikace p deskriptoru (2 reference)

Práce se souborem

Zrušení deskriptoru synchronizace (1 reference)

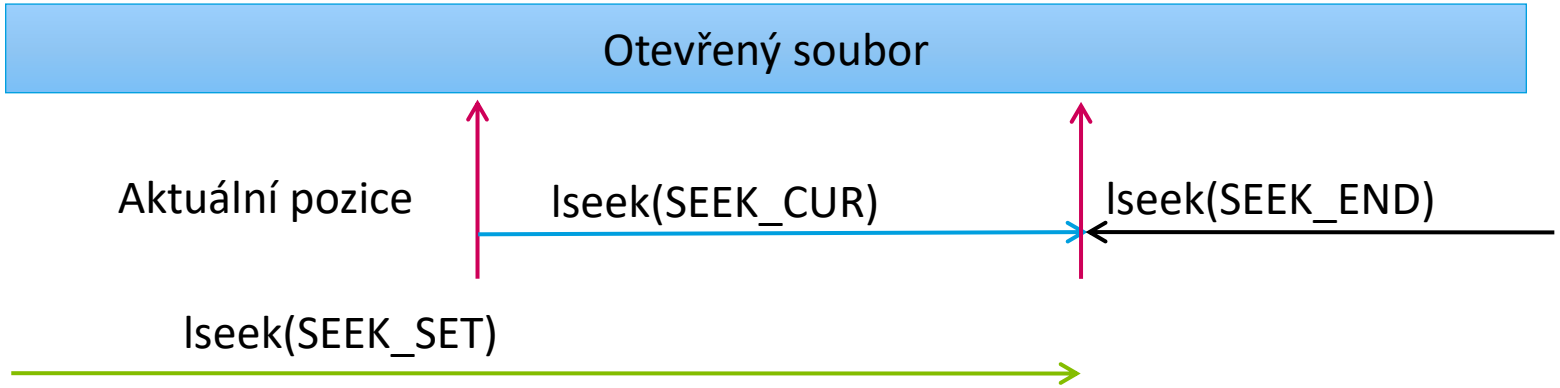
Zrušení deskriptoru synchronizace zavření souboru

Čtení a zápis

- Nejčastější operace s otevřeným souborem.
  - často zapouzdřeny do operací vyšší úrovně.
- Čtení i zápis podobné operace – pracují s částí paměti (bufferem), do kterého se načítají nebo ze kterého se zapisují data.
- Volání funkce **read()**
  - čeká, dokud nepřečte nějaká data (ne vždy celý požadovaný objem), nebo nepřijde konec souboru či není přerušeno signálem.
- Funkce **write()**
  - funguje velice podobně, jen data zapisuje. Ne vždy dojde k zápisu najednou, proto je nutné kontrolovat návratovou hodnotu a porovnávat ji s velikostí připravených dat pro zápis.

Pozice v souboru, synchronizace

- Při otevření souboru pro zápis nebo čtení je nastavena pozice na začátek souboru. Při použití O_APPEND bude nastavena na konec.
- Funkce pro nastavení aktuální pozice je **lseek()**. Je volána s uvedením deskriptoru, posunu a druhu bázové pozice.
- Volby: `SEEK_SET` – pozice od začátku; `SEEK_CUR` – relativní pozice vůči aktuální pozici; `SEEK_END` – relativně vzhledem ke konci.
- Funkce pak vrací novou absolutní pozici.



Pozice v souboru, synchronizace

- Jádro většinou nezapíše souborová data hned není opuštění funkce **write()** dostatečným potvrzením o ukončení zápisu.
- Pro spolehlivý zápis stačí zavolat funkci **sync()**, která zajistí zápis všech nezapsaných dat v celém systému.
- Není-li nutné zapisovat vše stačí použít funkci **fsync()** - volaná s deskriptorem souboru jako parametrem.

Volání sendfile

- Přenášení dat mezi otevřenými souborovými deskriptory je posloupnost operací volání **read()** a **write ()**.
  - Nejprve se kopírují do bufferu v uživatelském prostoru a následně zase do jádra.
- **Sendfile ()** osílá data přímo z jednoho souborového deskriptoru do druhého
  - Eliminace mezipřenosu do uživatelského prostoru
- Příkad

```
off_t offset = 100;
Ssize_t res = sendfile (outfd, infd, &offset, 100);
```

Volání přenese nejvýše 100 B (bajtů) dat, mezi deskriptory outfd, infd, přičemž začne přenášet od pozice 100 (proměnná offset).

Volání splice

- Rozdíl mezi **splice ()** a **sendfile ()**:
  - **splice ()** neumožňuje uvádět offset na straně, kde je roura (pipe)
  - Lze uvést další speciální příznaky pro změnu chování operace.

Zamykání souborů

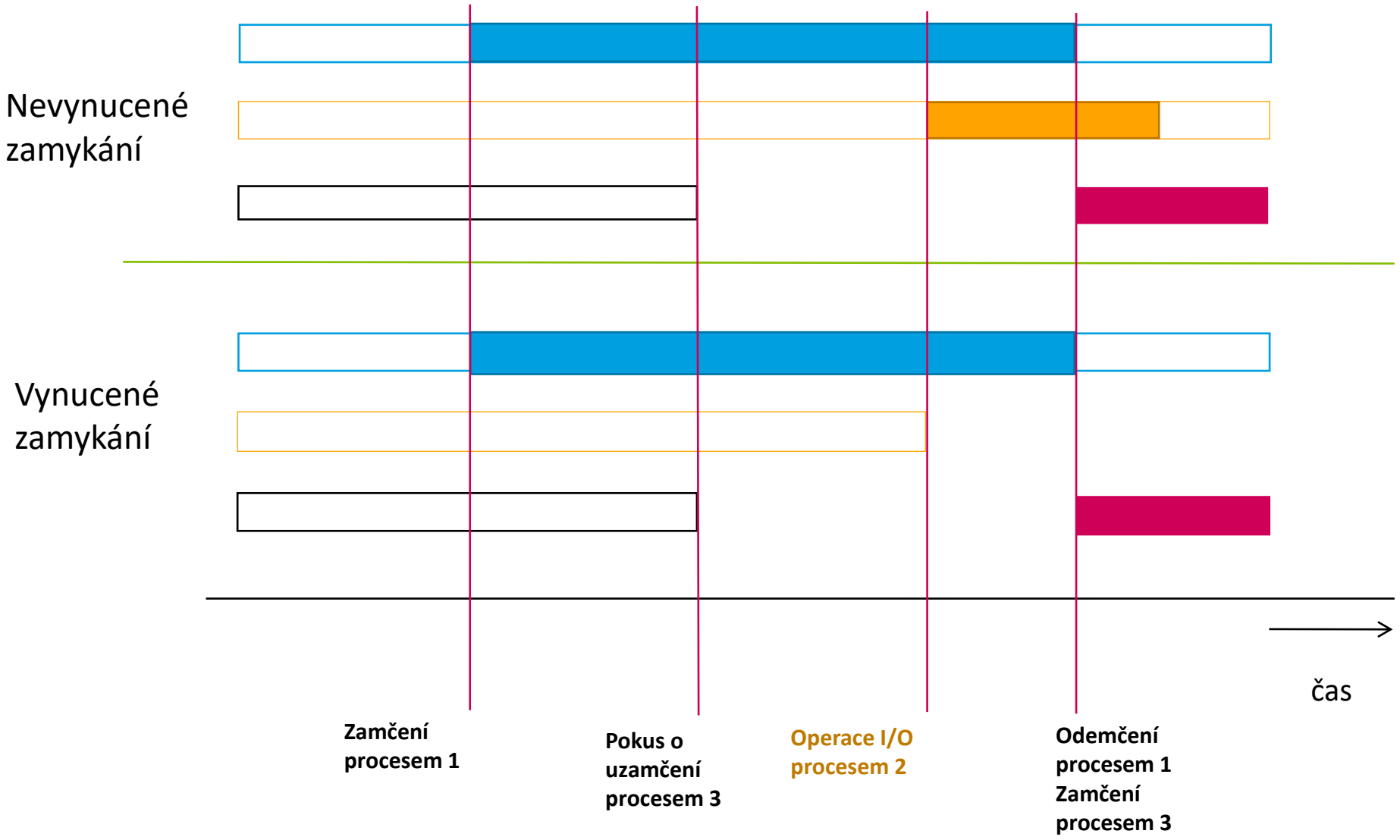
- Zamknutím souboru zabráníme přístupu jiných procesů do otevřeného souboru.
  - nevynucené zamykání (advisory locking).
  - vynucené zamykání (mandatory locking).
- Společné vlastnosti
  - Podporují blokující i neblokující operace
  - Zamknout lze jenom soubor, ne složku
  - Automaticky zanikají, když je proces skončí / je ukončen

Rozdíl v zamykání

- **Nevynucené**
  - Vyžaduje spolupráci všech procesů přistupujících k souboru.
  - Proces se pokusí získat zámek, pokud se mu to povede, drží jej až do uvolnění.
  - Pokud se mu to nepovede, bude čekat na uvolnění nebo skončí chybou.
  - Operace: čtení, zápis atd. nejsou zámkem uvolněny.
- **Vynucené**
  - Nevyžaduje spolupráci procesů, ovlivňuje přímo souborové operace.
  - Ty se buď blokují do odemčení nebo hned selžou.
  - Není stále 100% spolehlivý
    - “deadlock“ při souběhu volání
    - špatně napsaný program nemusí zámek nikdy uvolnit

FIM UHK

Rozdíl v zamykání



Další operace na souborech

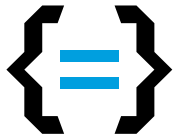
- Vytváření adresářů
- Vytváření speciálních souborů
- Odstranění souborů a adresářů
- Práce s adresářovým stromem
- Přejmenování a přesun souborů
- Atributy souborů

Další operace na souborech

- Vytváření adresářů **mkdir()**
  - pracuje s relativní cestou vzhledem k aktuálnímu adresáři nebo s absolutní cestou.
- Odstranění souborů **unlink()**
  - maže veškeré soubory, ale ne adresáře. Pro mazání musí mít uživatel právo zápisu do adresáře, kde je soubor umístěn.
- **rmdir()** – funkce pro mazání adresářů.
  - adresář musí být prázdný,
- nesmí být aktuálně používaným přípojným bodem pro souborový systém,
  - nesmí se z pohledu procesu jednat o kořenový adresář.

Další operace na souborech

- rename() – funkce pro přejmenování/přesun
  - jde do značné míry o totéž – jedná se pohyb v rámci jednoho souborového systému,
  - přesun souboru do jiného adresáře probíhá tak, že se nejdřív vytvoří nový název a pak se odstraní ten starý,
  - nemůže tedy nastat situace, že by při selhání souboru zcela zmizel.
- Práce se soubory v terminálu
  - Příkazy touch, cp, mv, rm, chmod, chown
  - Midnight Commander



Univerzita Hradec Králové
Fakulta informatiky a managementu

Děkuji za pozornost

Další téma: Souborové systémy