



Univerzita Hradec Králové  
Fakulta informatiky a managementu

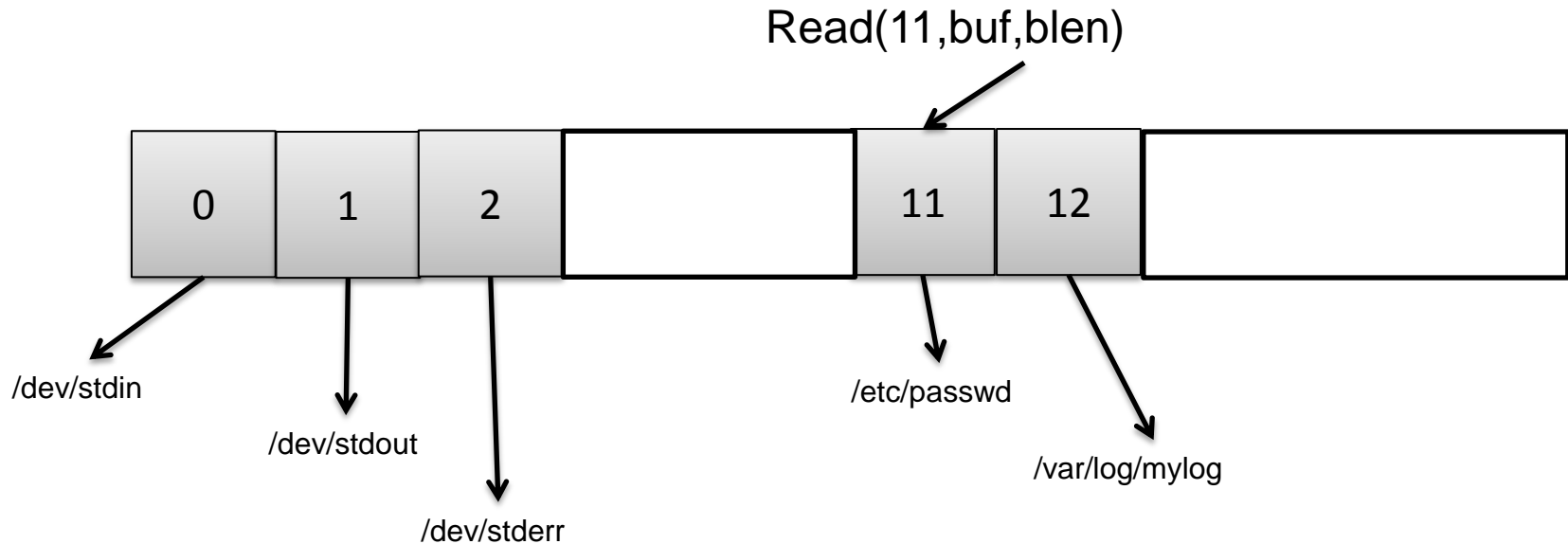
# Souborové operace

Mgr. Josef Horálek



- = 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.)

- = 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 st. vstup; 1 pro st. výstup; 2 pro chybový výstup)



= Vazba deskriptorů a otevřenými soubory

## = 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

- = 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 fd=open(“”soubor.txt“, O_RDONLY);
```

- = 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
  - = O\_SYNC – synchronní režim
  - = O\_NOATIME – režim bez aktualizace času přístupu

- = Režim chování funkce:
  - = O\_CREAT – vytvoření souboru
  - = O\_EXCL – ochrana existujícího souboru
  - = O\_NOLINK – otevření symbolického odkazu místo cílového souboru
  - = O\_TRUNC – zkrácení souboru na nulovou délku



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

- = 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.

Open("/etc/hosts", O\_RDONLY)

fork ()

...

Close()

...

Close()

Otevření souboru  
(1 reference)

Duplikace deskriptoru  
(2 reference)

Práce se souborem

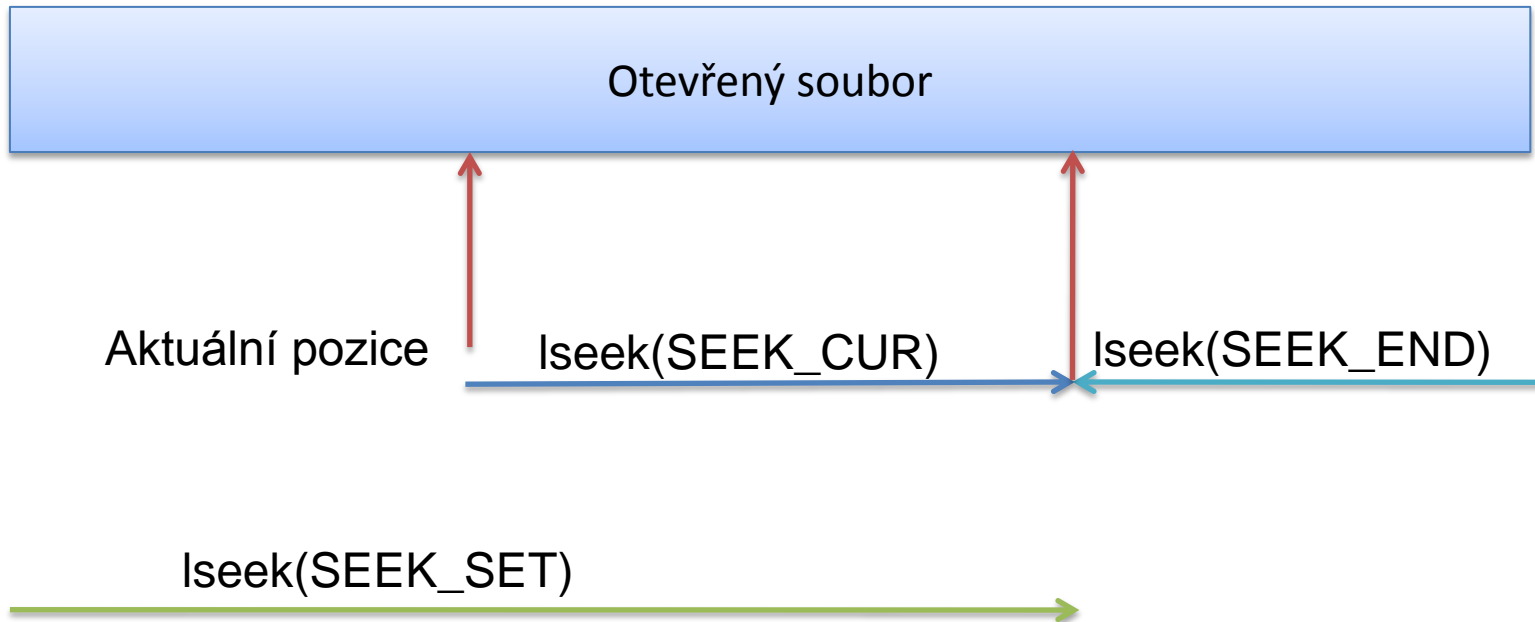
Zrušení deskriptoru  
synchronizace  
(1 reference)

Zrušení deskriptoru  
synchronizace  
zavření souboru

- = 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.

- = 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 vrátí novou absolutní pozici.



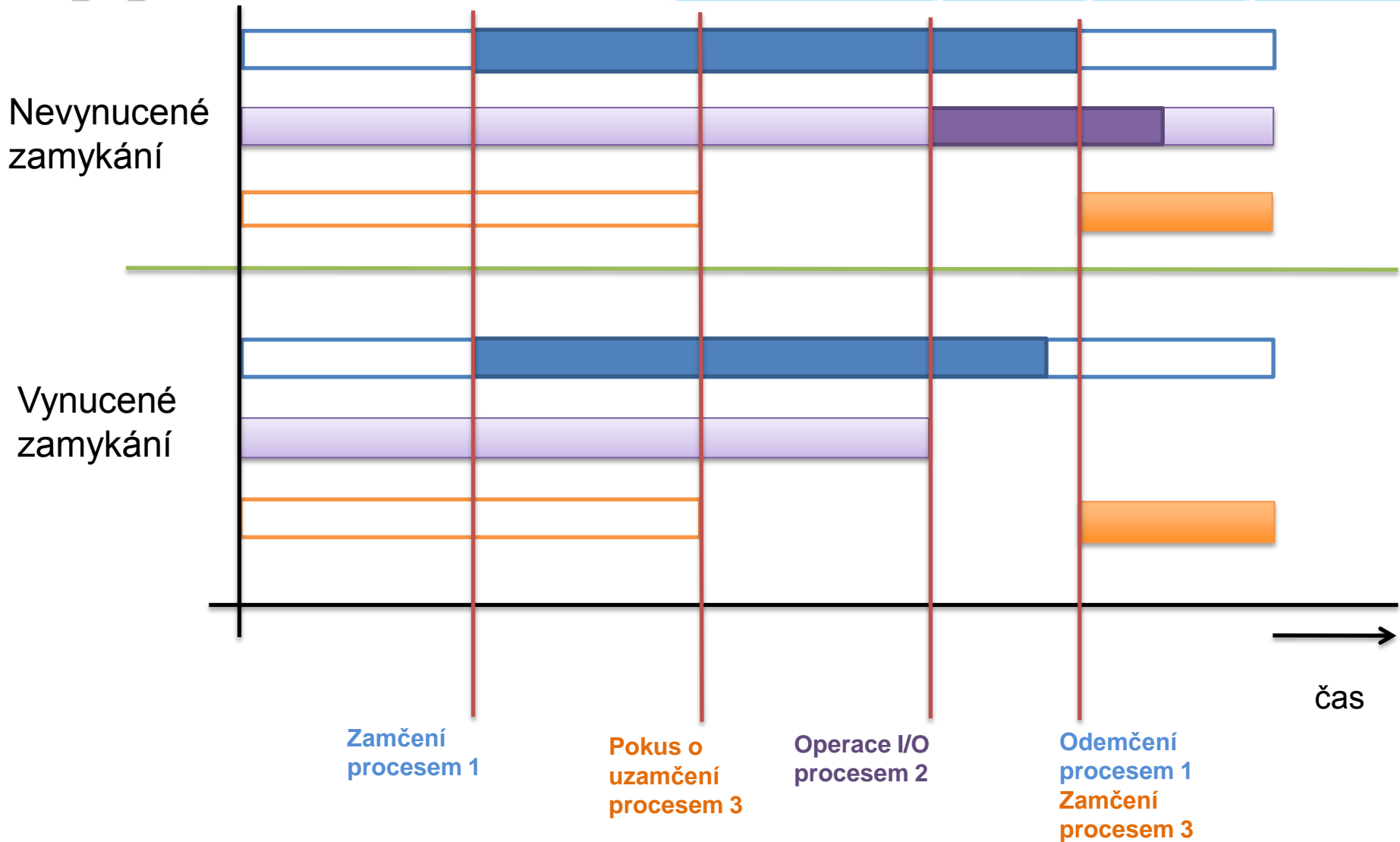


- = Jádro většinou nezapiše souborová data hned není opuštění funkce **write()** dostatečným potvrzením o ukončení zápisu.
- = Pro spolehlivý zápis stačí zavolat funkci **sync()**, která zajistí zápis všech nezapsaných dat v celém systému.
- = Není-li nutné zapisovat vše stačí použít funkci **fsync()**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

- = Zamknutím souboru zabráníme přístupu jiných procesů do otevřeného souboru.
- = Nevynucené zamykání (advisory locking)
- = Vynucené zamykání (mandatory locking)

- = Vyžaduje spolupráci všech procesů přistupujících k souboru.
- = Proces se pokusí získat zámek, pokud se mu to provede, drží jej až do uvolnění.
- = Pokud se mu to nepovede, bude čekat na uvolnění nebo skončí chybou.
- = Operace: čtení, zápis atd. nejsou zámkem ovlivněny.

- = Nevyžaduje spolupráci procesů, ovlivňuje přímo souborové operace.
- = Ty se buď blokují do odemčení nebo hned selžou.



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

- = 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.



- = **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.



Univerzita Hradec Králové  
Fakulta informatiky a managementu

Děkuji za pozornost...

