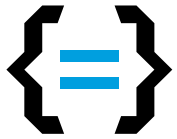


Univerzita Hradec Králové
Fakulta informatiky a managementu

Roury a zprávy

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

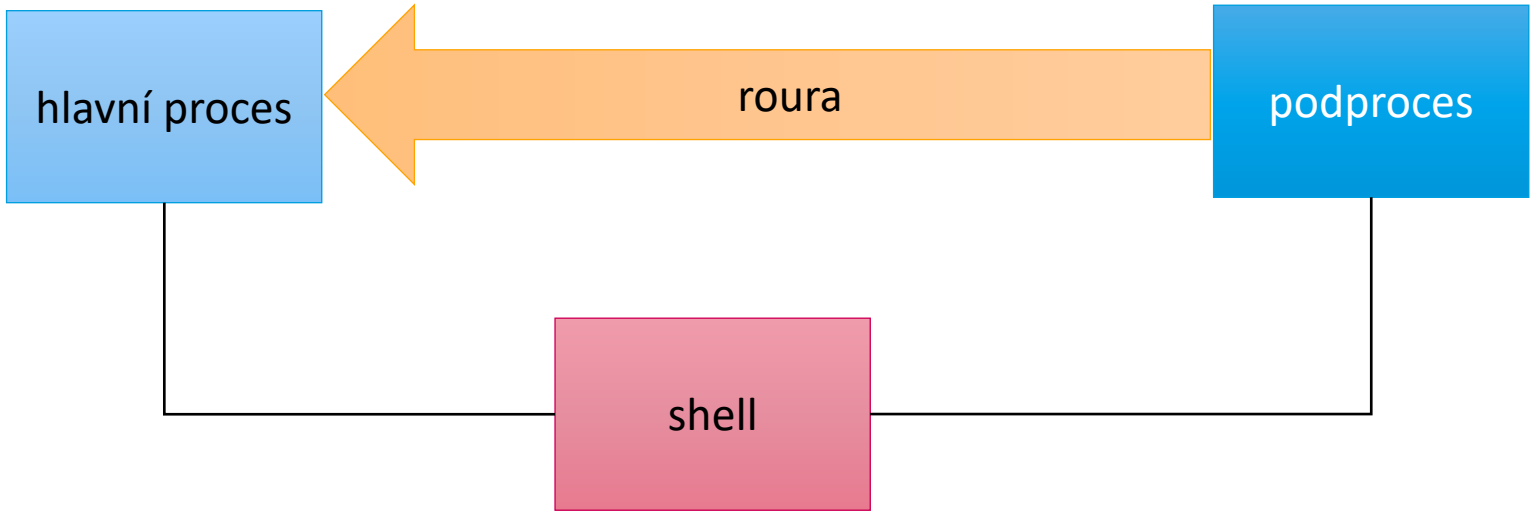
- Jde o metodu místní komunikace mezi procesy.
- Jedná se o:
  - Anonymní roury
  - Pojmenované roury
  - Komunikace pomocí zpráv

Anonymní roury (pipe)

- Nejjednodušší mechanismus pro komunikaci mezi procesy.
- Jedná se o proud bajtů – do jednoho konce se zapisuje, ze druhého se čte.
- Na obou koncích může být ten samý proces nebo různé procesy.
- Konec roury může být sdílen i více procesy.
- Z pohledu implementace jde o rozhraní k paměťovému bufferu.
- Buffer je omezený (starší jádra 4096 bajtů) u novějších 65536 bajtů.
- Zapisuje-li do roury několik procesů zároveň, nelze garantovat pořadí, ve kterém se data v rouře objeví.

Jednoduchá roura do procesu

```
Popen("myprog", "r")
```



Roura s oběma konci

- Jde o flexibilnější řešení.
- Náročnější na implementaci – funkce pipe().
- Vytvoří rouru, do které máme k dispozici oba konce.
- Plnohodnotná komunikace s podprocesy.
- Komunikace mezi vlákny jednoho procesu.
- Řízení procesu událostí.
- Relativně bezpečné ošetření signálu.

Komunikace s podprocesem

- Jak komunikovat s podprocesem?
- Vytvořit jich dostatečný počet a nechat je zdědit.
- Chceme.li komunikovat s procesem, který o nich „nic“ neví musíme jimi nahradit standardní výstup, vstup nebo chybový výstup.
- Náhrada proběhne po vytvoření podprocesu, ale před spuštěním nového programu.

Komunikace mezi vlákny

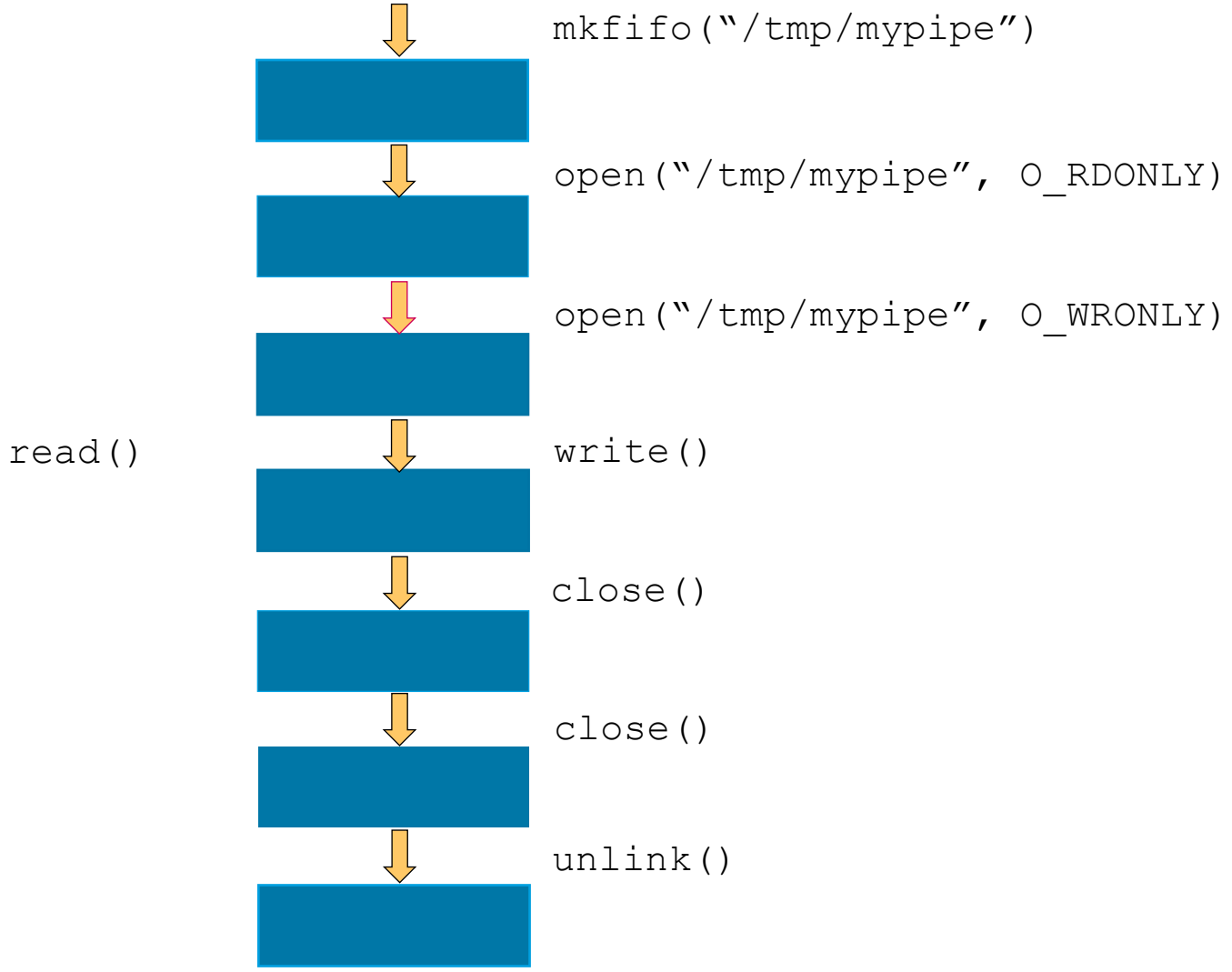
- Vytvoří se roura, jejíž jeden konec se používá v jednom vlákně a druhý v jiném, nebo i ve více vláknech.
- Při rušení vlákna je doporučeno zavřít příslušný konec roury.

Pojmenované roury

- Pojmenovaná roura (named pipe, fronta FIFO) – komunikační objekt používaný podobně jako nepojmenovaná roura.
- Pojmenovaná roura přebývá ve svém domovském adresáři i v době, kdy s ní žádný proces nepracuje.
- Může být tedy uložena do archivu a následně obnovena – jde vlastně o speciální soubor.
- Před využitím pojmenované roury se musí vytvořit k tomu využijeme volání **mkfifo()**.
- Jako argument využívá cestu v souborové systému a masku oprávnění.
- Pokud rouru nepotřebujeme zrušíme ji voláním funkce **unlink()**.

FIM UHK

Práce s rourou



Roury v shellu

- Jednosměrná pipe se značí symboly > < |
- Vylistování seznamu souborů
  - `ls`
- Vylistování souborů s filtrem
  - `ls | grep „*.log“`
- Vylistování souborů s filtrem a uložením do souboru
  - `ls | grep „*.log“ > seznam.txt`

- Aplikace se spustí, otevře kanál v syslogu a jednosměrnou rouru
- `echo „NODE LIST“ > /tmp/node-emulator`

```
//open syslog
setlogmask(LOG_UPTO(LOG_DEBUG));
openlog("node-emulator", LOG_CONS | LOG_PID | LOG_NDELAY,

        LOG_LOCAL1);

//create the FIFO (named pipe)
if (mkfifo(„/tmp/node-emulator", 0666) == -1) {
        if (errno != EEXIST) {
        cout << "mkpipe()!" << strerror(errno) << endl;
                return false;
    }
}

return true;
```

Komunikace pomocí zpráv

- Roury jsou objekty orientované na proud bajtů – nestrukturovaná data.
- Potřebujeme-li přenášet zprávy o pevně daném formátu využijeme technologii určenou pro tuto oblast např:
  - POSIX Message Queues.
  - MQTT
  - REST API
  - WebSockets
  - SOAP
  - D-Bus

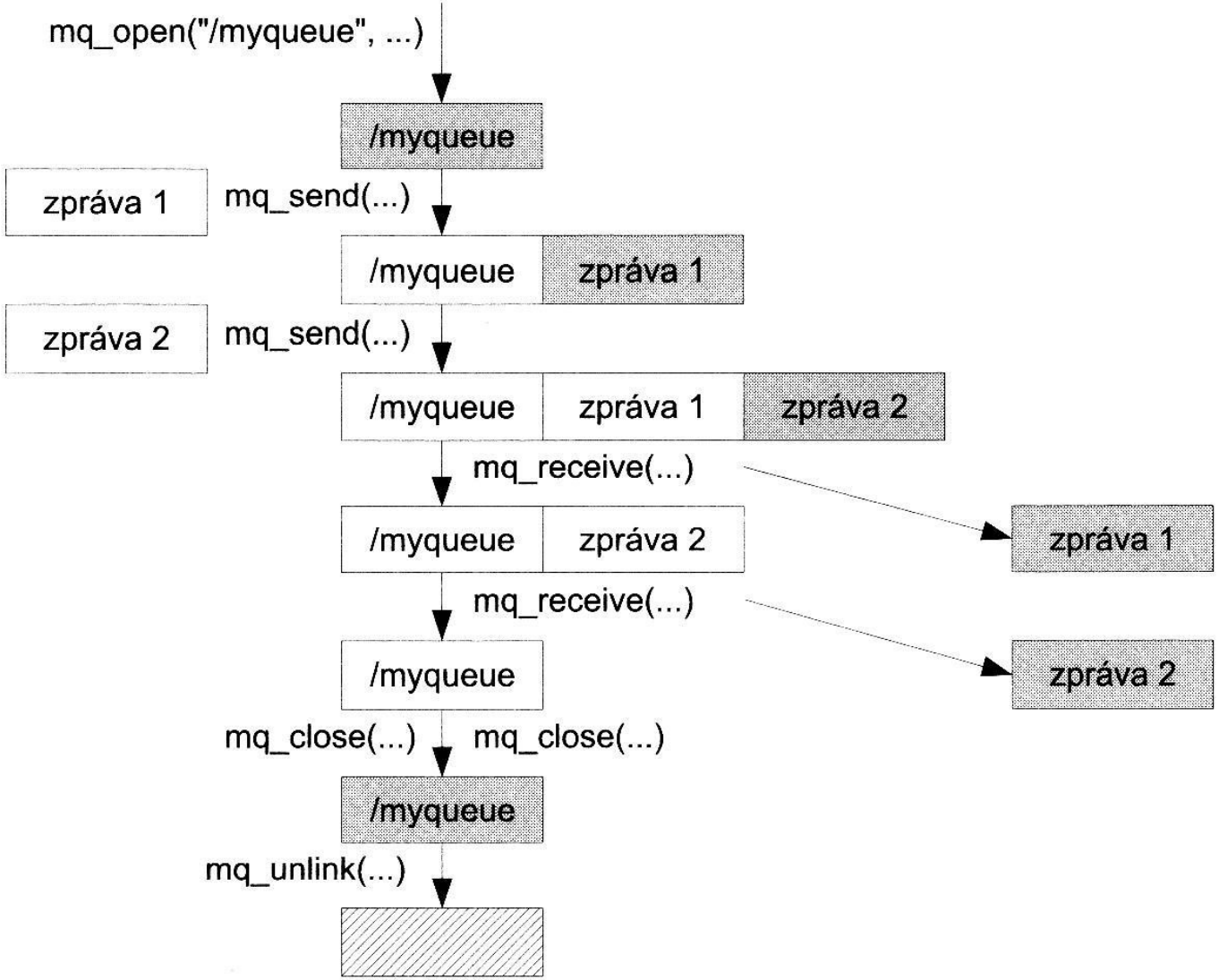
POSIX Message Queues

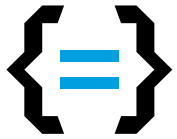
- Technologie specifikovaná přímo na POSIX.
- Vyžaduje podporu jádra, příslušnou knihovnou pro uživatelské programy.
- Technologie POSIX PMQ umožňuje vytvářet pojmenované fronty.
- Do téže fronty může proces zprávy posílat a odebírat.
- Notifikace při příchodu zprávy může být asynchronní nebo synchronní.

POSIX Message Queues

- Každá zpráva má určenu prioritu.
- Linux podporuje 32 768 úrovní priorit.
- Vytvořená fronta funguje do restartu systému nebo jejího zrušení.
- Nepřečtená zprávy zůstávají ve frontě.
- Limity pro fronty:
  - měkké
  - tvrdé

Práce s frontou zpráv PMQ





Univerzita Hradec Králové
Fakulta informatiky a managementu

Děkuji za pozornost

Další téma: Sockety