

PB173 – Binární programování Linux

XI. Komunikace mezi procesy, část 1

Jiri Slaby

Fakulta informatiky
Masarykova univerzita

2. 12. 2014

Komunikace mezi procesy

Alespoň 2 procesy, které chtějí komunikovat

Dnes

- 1 Roura
 - Pojmenovaná roura
- 2 Sdílená paměť
- 3 Plná meziprocesová komunikace
 - Sdílená paměť
 - Zprávy
 - Semafory

Příště

- Netlink
- RPC

Sekce 1

Roura

- Jednosměrná komunikace
 - Jádro při `fork` kopíruje deskriptory
- Vytvoření: `pipe`
 - Vrací pole 2 deskriptorů (+ návratovou hodnotu)
 - Jeden pro potomka, druhý pro rodiče
 - Proveďte se `fork`
 - Zavřou se nepotřebné deskriptory
 - Čte se z prvního deskriptoru (`read`)
 - Zapisuje se do druhého deskriptoru (`write`)
- `popen=pipe+fork+exec`

Vytvoření „echa“ přes roury

- 1 Vytvořte si 2 roury
 - Pro obousměrnou komunikaci
 - Budete mít `fd1[2]` a `fd2[2]`
- 2 Proveďte `fork`
- 3 Zavřete nepotřebné deskriptory
- 4 V jednom procesu v cyklu o několika iteracích:
 - Do `fd1[1]` zapisujte měnící se text
 - Např. `char buf[10] = "Hello"; a (*buf)++;`
 - Z `fd2[0]` čtete a vypisujete na standardní výstup
- 5 V druhém procesu (loopback):
 - Čtete z `fd1[0]`
 - A zapisujete to do `fd2[1]`
- 6 Spust'ete

- Soubor na disku
- mknod
- **Demo**

Sekce 2

Sdílená paměť

- Mapování a sdílení paměti
 - Jádru při `fork` kopíruje mapování
- Vytvoření: `mmap` s `MAP_ANONYMOUS` | `MAP_SHARED`
 - Prove se `fork`
 - Paměť je volně k dispozici oběma procesům
 - Jen pozor na optimalizace překladače
 - `volatile`

Vytvoření sdílené paměti

- 1 Vytvořte si sdílené mapování
- 2 Proveďte `fork`
- 3 V jednom procesu v cyklu o několika iteracích:
 - Zapisujte do paměti proměnný řetězec
 - Čekejte 100 ms
- 4 V druhém procesu:
 - Čtěte a vypište řetězec
 - Čekejte 100 ms
- 5 Spusťte

Sekce 3

Plná meziprocesová komunikace

- Plná meziprocesová komunikace
 - Sdílená paměť
 - Zprávy
 - Semaforey
- Výpis: `ipcs` (**Demo**)
- Dokumentace: `man 5 ipc`
- `sys/ipc.h`

- IPC pracuje s klíči (`key_t`)
- Procesy komunikují pomocí klíče
 - Jen pokud mají stejný
- Vytvoření klíče: `ftok` (`sys/ipc.h`)
 - Cesta k existujícímu uzlu
 - Nějaké 8bitové číslo
- Např. `ftok(".", 'x')`

- `sys/shm.h`
- Vytvoření: `shmget`
 - Poslední parametr `IPC_CREAT`
 - Logický OR s právy (0644 apod.)
- Získání adresy (attach): `shmat`
- Navrácení adresy (detach): `shmdt`
- Další operace: `shmctl`
 - Informace: `SHM_STAT`
 - Zrušení: `IPC_RMID`
 - ...

Vytvoření sdílené paměti (IPC)

- 1 Vytvořte si klíč (`ftok`)
 - A vypište ho (`%x`)
- 2 Vytvořte si sdílený segment (`shmget`)
- 3 Získejte adresu (`shmat`)
- 4 Nyní si zkopírujte tento `main` do dvou souborů
- 5 V jednom souboru v cyklu o několika iteracích:
 - Zapisujte do paměti proměnný řetězec
 - Čekejte 100 ms
- 6 V druhém souboru:
 - Čtete a vypište řetězec
 - Čekejte 100 ms
- 7 V obou proveďte `shmdt`
- 8 Spusťte

Formát zprávy

```
struct my_message {  
    long mtype; /* set to something > 0 */  
    ...  
};
```

- `sys/msg.h`
- Vytvoření fronty zpráv: `msgget`
- Odeslání zprávy: `msgsnd`
- Přijetí zprávy: `msgrcv`
- Další operace: `msgctl`
 - Informace: `MSG_STAT`
 - Zrušení: `IPC_RMID`
 - ...

Posílání zpráv (IPC)

- 1 Vytvořte si klíč (`ftok`)
 - A vypište ho (`%x`)
- 2 Vytvořte si frontu zpráv (`msgget`)
- 3 Vytvořte si strukturu zprávy
 - Musí obsahovat `long` `mtype`
 - Přidejte `char` `text[10]`
- 4 Nyní si zkopírujte tento `main` do dvou souborů
- 5 V jednom souboru v cyklu o několika iteracích:
 - Zapisujte do fronty (`msgsnd`) řetězec
 - Čekejte 100 ms
- 6 V druhém souboru:
 - Čtěte frontu (`msgrcv`) a vypište řetězec
 - Čekejte 100 ms
- 7 Spusťte

- `sys/sem.h`
- Meziprocesové zamykání/správa prostředků
- Vytvoření semaforu: `semget`
- Změna stavu semaforu: `semop`
- Další operace: `semctl`
 - *Nastavení iniciální hodnoty*: `SETVAL` (spolu s `union semun.val`)
 - Informace: `SEM_STAT`
 - Zrušení: `IPC_RMID`
 - ...

Parametr `semop`

```
struct sembuf {  
    ushort  sem_num;  
    short   sem_op; /* - 0 + */  
    short   sem_flg;  
};
```

Práce se semaforem (IPC)

- 1 Vytvořte si klíč (`ftok`)
 - A vypište ho (`%x`)
- 2 Vytvořte 2 programy
- 3 Jeden program je tiskovým serverem
 - Vytvořte si 1 semafor (`semget`)
 - Obsluhuje 5 tiskáren (`shmctl` a `SETVAL`)
 - Tisk trvá 10 vteřin
- 4 Druhý program je klientem
 - Chce tisknout
- 5 Spusťte jeden server a více než 5 klientů
 - Do 10 vteřin