PB173 – Binární programování Linux

XI. Komunikace mezi procesy, část 1

Jiri Slaby

ITI, Fakulta informatiky

3. 12. 2013

Komunikace mezi procesy

- Alespoň 2 procesy, které chtějí komunikovat
- Dnes
  - Roura (`pipe`)
  - Sdílená paměť (`mmap`)
  - Plná meziprocesová komunikace (IPC)
- Příště
  - Netlink
  - RPC

Část I

Roury

Roury

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

Úkol

Vytvoření „echa” přes roury

1. Vytvořte si 2 roury
   - Pro obousměrnou komunikaci
2. Proveďte `fork`
3. Zavřete nepotřebné deskriptory
4. V jednom procesu v cyklu o několika iteracích:
   - Do jednoho `fd` zapisujte měnící se text
     - Např. `char buf[10] = "Hello";` a `(*buf)++;`
   - Z druhého `fd` čtěte a vypisujte
5. V druhém procesu (echo):
   - Čtěte z prvního `fd`
   - A zapisujte to do druhého `fd`
6. Spusťte

Pojmenovaná roura

- Soubor na disku
- `mknod`
- Demo

Část II

Sdílená paměť

Sdílená paměť

- Mapování a sdílení paměti
  - Jádro při `fork` kopíruje mapování
- Vytvoření: `mmap` s `MAP_ANONYMOUS | MAP_SHARED`
  - Provede se `fork`
  - Paměť se může libovolně měnit
  - Ale pozor na optimalizace překladače
    - `volatile`

Úkol

Vytvoření sdílené paměti

1. Vytvořte si 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

Část III

IPC

IPC

- Plná meziprocesová komunikace
  - Sdílená paměť
  - Zprávy
  - Semafory
- Výpis: `ipcs`
- Dokumentace: `man 5 ipc`

IPC a klíče

- IPC pracuje s klíči (`key_t`)
- Vytvoření: `ftok`
  - Cesta k existujícímu uzlu
  - Nějaký znak
- Např. `ftok(".", 'x')`

Sekce 1

Sdílená paměť

Sdílená paměť (IPC)

- Vytvoření: `shmget`
  - Parametr `IPC_CREAT`
  - Logický OR s právy
- Získaní adresy: `shmat`
- Navrácení adresy: `shmdt`
- Další operace: `shmctl`
  - Informace: `SHM_STAT`
  - Zrušení: `IPC_RMID`
  - ...

Úkol

Vytvoření sdílené paměti (IPC)

1. Vytvořte si klíč (`ftok`)
   - A vypište ho
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:
   - Čtěte a vypište řetězec
   - Čekejte 100 ms
7. V obou proveďte `shmdt`
8. Spusťte

Sekce 2

Zprávy

Zprávy (IPC)

Formát zprávy

```
struct msgbuf {
  long mtype;
  ...
};
```

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

Úkol

Posílání zpráv (IPC)

1. Vytvořte si klíč (`ftok`)
   - A vypište ho
2. Vytvořte si frontu zpráv (`msgget`)
3. Nyní si zkopírujte tento `main` do dvou souborů
4. V jednom souboru v cyklu o několika iteracích:
   - Zapisujte do fronty (`msgsnd`) řetězec
   - Čekejte 100 ms
5. V druhém souboru:
   - Čtěte frontu (`msgrcv`) a vypište řetězec
   - Čekejte 100 ms
6. Spusťte

Sekce 3

Semafory

Semafory (IPC)

- 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;
};
```

Úkol

Práce se semaforem (IPC)

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