PB173 – Binární programování Linux

XII. Komunikace mezi procesy, část 2

Jiri Slaby

ITI, Fakulta informatiky

10. 12. 2013

Komunikace mezi procesy

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

Část I

Netlink

Netlink

- Protokol pro komunikaci s jádrem
  - Standardní socket (`sys/socket.h`)
  - Ale umožňuje i komunikaci mezi procesy
- Adresování: `struct sockaddr_nl` (`linux/netlink.h`)
  - `nl_family = AF_NETLINK`
  - `nl_pid` je většinou číslo procesu
- Vytvoření: `socket(AF_NETLINK, SOCK_RAW, NETLINK_USERSOCK)`
- Server
  - Provede `bind` s nastaveným `sockaddr_nl.nl_pid`
  - Potom poslouchá pomocí `recv`
- Klient
  - Posílá pomocí `sendto`

Úkol

Vytvoření netlink komunikace

1. Vytvořte si 2 programy
   - Server a klient
2. V serveru proveďte:
   - `bind` s nastaveným `sockaddr_nl.nl_pid`
   - Vypište si `nl_pid`
   - A v cyklu `recv` a `write` na standardní výstup
3. V klientovi proveďte:
   - `sendto` nějakých dat
   - Adresu (`nl_pid`) si vezměte s příkazové řádky
4. Spusťte

Část II

RPC

RPC

- Vzdálené volání procedur
  - Volání funkcí přes síť (UDP nebo TCP)
  - Využití např. v NFS
- Portmapper
  - Lokální služba
  - Mapuje číslo programu, verzi a protokol na funkce
    - Registrovaná čísla v `/etc/rpc`

RPC rozhraní pro server

- `rpc/rpc.h`, `man 3 rpc`
- Vytvoření spojení: `svctcp_create`, `svcudp_create`
  - `RPC_ANYSOCK`
- Registrace funkce: `svc_register`
  - Registruje u portmapperu pomocí `pmap_set`
  - Viditelné v `rpcinfo -p`
  - Je zvykem před tím volat `pmap_unset`
  - Volá háček `dispatch`
    - Parametr `svc_req->rq_proc` určuje funkcionalitu
    - Lze odpovědět na zprávu: `svc_sendreply`
- Obslužná smyčka: `svc_run`

Úkol

Vytvoření RPC serveru

1. Vytvořte si TCP spojení (`svctcp_create` a `RPC_ANYSOCK`)
2. Zvolte si čísla funkce a verze
   - Vyhněte se číslům funkcí z `/etc/rpc`
3. Zrušte stará mapování (`pmap_unset`)
4. Registrujte službu (`svc_register`)
   - V `dispatch` funkci proveďte výpis `svc_req->rq_proc`
5. Spusťte smyčku (`svc_run`)
6. Spusťte program
   - Za předpokladu, že běží portmapper

RPC rozhraní pro klienta

- Vytvoření spojení: `clnt_create`
  - Stroj, protokol a číslo funkce+verze
- Zrušení spojení: `clnt_destroy`
- Zavolání funkce: `clnt_call`
  - 2. parametr do `dispatch` → `svc_req->rq_proc`
  - 3. a 5. parametr: kódovací funkce (zatím `xdr_void`)
  - 4. a 6. parametr: parametry vzdálené funkce (zatím `NULL`)
  - Timeout: čas na odpověď

Úkol

Vytvoření RPC klienta

1. Vytvořte si TCP spojení na localhost (`clnt_create`)
   - Použijte čísla zvolená v serveru
2. Zavolejte několikrát `clnt_call`
   - S různým 2. parametrem
3. Spusťte program
   - Za předpokladu, že server běží

RPC a parametry

XDR

- `rpc/xdr.h`, `man 3 xdr`
- Funkce k přenositelnému zakódování dat
  - `xdr_void`
  - `xdr_char`, `xdr_short`, `xdr_int`, `xdr_long`
  - `xdr_float`, `xdr_double`
  - `xdr_string`, `xdr_array`
  - `...`
- Klient
  - 3. a 5. parametr pro `clnt_call`
- Server
  - V `dispatch`: `svc_getargs`

Úkol

Předávání parametrů

1. V serveru vytvořte funkci
   - Vezme `int`, vynásobí `10` a vrátí `long`
2. Z klienta zavolejte
3. Spusťte program