PB152cv/07 Cvičenie 6

(Práca so socketmi, preklad adries)

Jozef Sabo

03. 11. 2025

Masarykova Univerzita, Brno

Cvičenie 06

Otváranie socket-u (pätice?)

```
int socket(int domain, int type, int protocol);
```

`int domain` – adresovacia rodina

- `AF_UNIX` – komunikácia v rámci lokálneho systému
- `AF_INET, AF_INET6` – komunikácia v rámci internetu - IPv4, resp. IPv6

`int type` – typ socketu

- `SOCK_STREAM` – spájaný socket – protokol TCP
- `SOCK_DGRAM` – nespájaný socket – protokol UDP (potrebné použiť `send()`/`recv()`)

`int protocol` – protokol, používame hodnotu `0`

Zapisovanie do socket-u s priložením adresy (iba SOCK_DGRAM)

```
ssize_t sendto(int socket, void *message, size_t length, int flags,
               struct sockaddr *dest_addr, socklen_t dest_len);
```

`int socket` – popisovač súboru reprezentujúci socket

`void *message` – buffer obsahujúci dáta na odoslanie

`size_t length` – veľkosť dát na odoslanie

`int flags` – menia správanie, používame hodnotu `0`

`struct sockaddr *dest_addr` – smerník na `sockaddr`, v ktorom je uložená adresa

`socklen_t dest_len` – dĺžka tejto adresy – v závislosti na type

Čítanie zo socket-u s uložením adresy (iba SOCK_DGRAM)

```
ssize_t recvfrom(int socket, void *buffer, size_t length, int flags,
                 struct sockaddr *address, socklen_t *address_len);
```

`int socket` – popisovač súboru reprezentujúci socket

`void *buffer` – buffer pre prijímanie dát

`size_t length` – veľkosť buffera

`int flags` – menia správanie, používame hodnotu `0`

`struct sockaddr *address` – smerník na štruktúru `sockaddr`, do ktorej uložiť adresu odosielateľa alebo `NULL` ak adresu zahadzujeme

`socklen_t *address_len` – smerník na premennú typu `socklen_t`, do ktorej uložíme dĺžku adresy odosielateľa alebo `NULL` ak adresu zahadzujeme

Priradenie adresy socket-u (v PB152cv iba `AF_UNIX`, server)

```
int bind(int socket, struct sockaddr *address, socklen_t address_len);
```

`int socket` – popisovač súboru reprezentujúci socket

`struct sockaddr *address` – smerník na `sockaddr`, v ktorom je uložená adresa, ktorú chceme priradiť socketu. V našom prípade to bude pretypovaná adresa na typ `struct sockaddr_un`

`socklen_t address_len` – dĺžka adresy, v našom prípade `sizeof(struct sockaddr_un)`

Nadviazanie spojenia socket-u (klient)

```
int connect(int socket, struct sockaddr *address, socklen_t address_len);
```

`int socket` – popisovač súboru reprezentujúci socket

`struct sockaddr *address` – smerník na `sockaddr`, v ktorom je uložená adresa, ku ktorej sa chceme pripojiť

`socklen_t address_len` – dĺžka adresy reprezentovanej v `struct sockaddr *address`

pre `SOCK_STREAM` a `SOCK_UNIX` sa pokúsi o nadviazanie spojenia na danej adrese

pre `SOCK_DGRAM` resetuje (a nastaví) implicitnú adresu – potrebná pre komunikáciu so serverom na známej adrese

Preklad adresy

```
int getaddrinfo(char *nodename, char *servname, struct addrinfo *hints,
                struct addrinfo **res);
```

`char *nodename` – reťazec obsahujúci meno hostiteľa

`char *servname` – reťazec obsahujúci meno alebo číslo portu

`struct addrinfo *hints` – požiadavky na preklad adresy

- `.ai_family` – adresovacia rodina socketu ktorý žiadame (`AF_INET`, `AF_INET6`)
- `.ai_socktype` – typ socketu ktorý žiadame (`SOCK_STREAM`, `SOCK_DGRAM`)

`struct addrinfo **res` – smerník na smerník do ktorého sa uloží výsledok

výsledok je vo forme zreťazeného zoznamu ukončeného (ako zvyčajne) smerníkom `NULL`

pozor, návratová hodnota značí aj typ chyby, nielen že chyba nastala (neupravuje `errno`)

Preklad adresy - uvoľnenie

```
int freeaddrinfo(struct addrinfo *ai);
```

`struct addrinfo *ai` – hlava zreťazeného zoznamu obsahujúca preložené adresy