PB152cv/11 Cvičenie 8

(Práca so socketmi z pohľadu server, listen(), accept())

Jozef Sabo

20. 11. 2025

Masarykova Univerzita, Brno

Cvičenie 08

Otváranie socket-u (cvičenie 06)

```
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`

Priradenie adresy socket-u (cvičenie 06)

```
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)`

Nastavenie príznaku prijímania na socket-e

```
int listen(int socket, int backlog);
```

`int socket` – popisovač súboru reprezentujúci socket spájaného protokolu (napr. `SOCK_STREAM`)

`int backlog` – veľkosť fronty čakajúcich požiadavok

volanie nastaví socket do režimu prijímania (počúvania, odposluchu)

socket-u musí byť dopredu nastavená adresa (pomocou volania `bind()`)

pozor! – socket samotný neslúži na komunikáciu priamo (cez `read()`/`write()`)

Prijatie spojenia

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

`int socket` – popisovač súboru reprezentujúci socket spájaného protokolu nastavený na počúvanie (pomocou `listen()`)

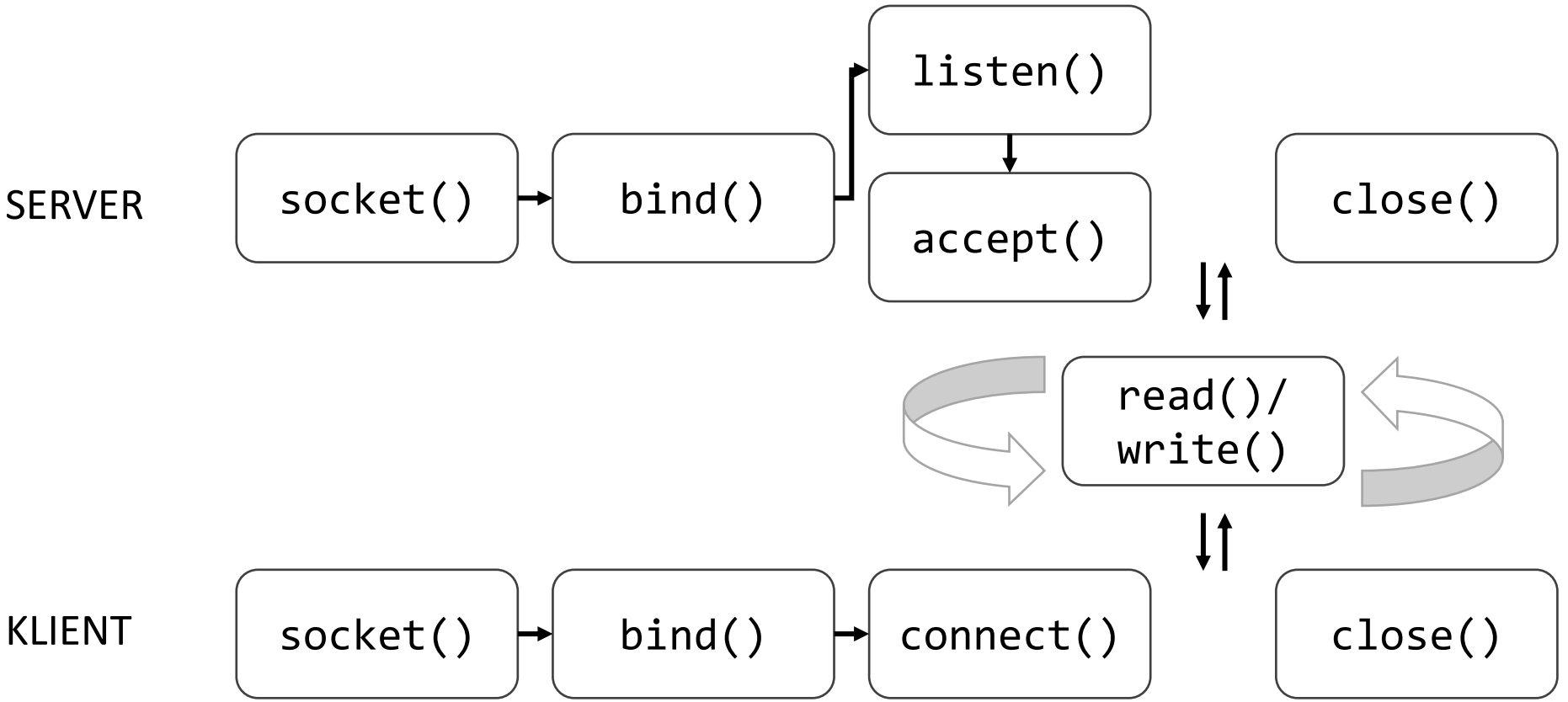
`struct sockaddr *address` – smerník na štruktúru `sockaddr`, do ktorej uložiť adresu pripájaného alebo `NULL` ak adresu zahadzujeme

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

prijme spojenie vyvolané druhou stranou cez `connect()` na adresu počúvajúceho socket-u (nastavené cez `listen()`)

`*address` a `*address_len` rovnako ako pri `recvfrom()` (cvičenie 06)

Príklad komunikácie



Príklad komunikácie

SERVER

socket()

```
int sock_fd = socket(..., SOCK_STREAM, ...);
// e.g. receives sock_fd as 42
```

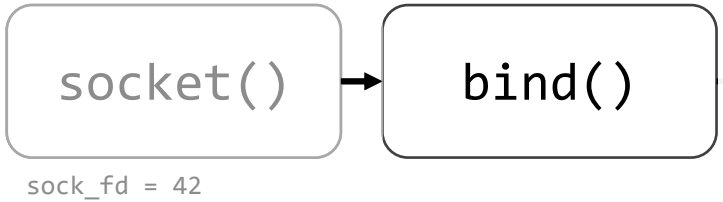
KLIENT

socket()

```
int cl_sock_fd = socket(..., SOCK_STREAM, ...);
// e.g. receives cl_sock_fd also as 42
```

Príklad komunikácie

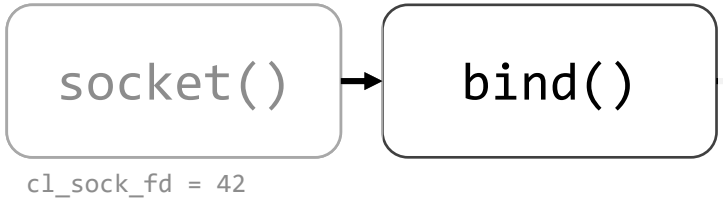
SERVER




```
struct sockaddr_un sa = { 0 };
sa.sun_family = AF_UNIX;
snprintf( sa.sun_path, 10, "%s", "./file1" );

bind( sock_fd, ( struct sockaddr * ) &sa,
      sizeof sa );
```

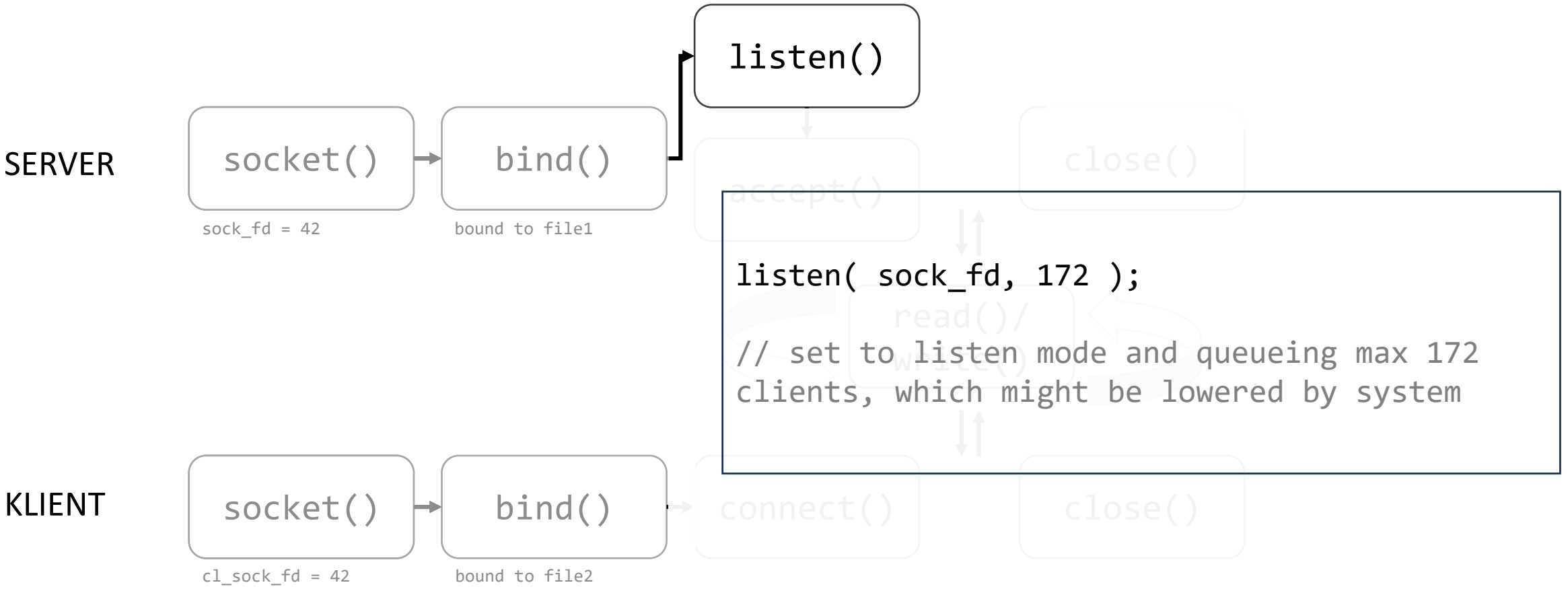
KLIENT




```
struct sockaddr_un sa = { 0 };
sa.sun_family = AF_UNIX;
snprintf( sa.sun_path, 10, "%s", "./file2" );

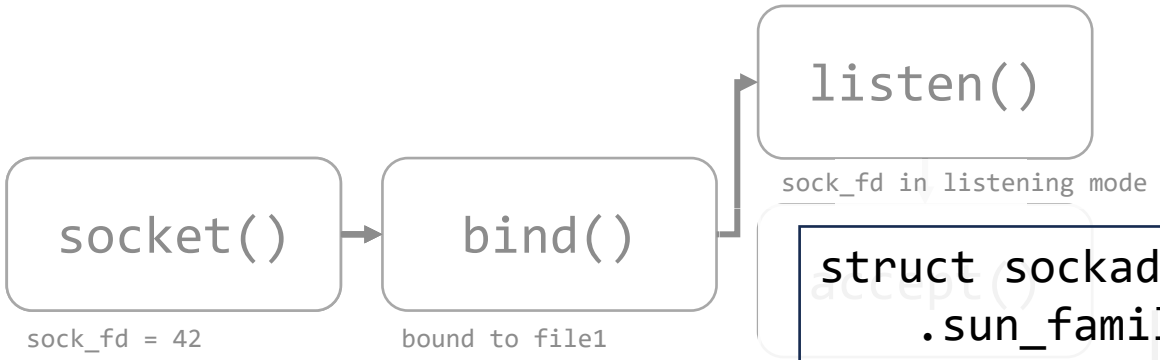
bind( cl_sock_fd, ( struct sockaddr * ) &sa,
       sizeof sa );
```

Príklad komunikácie



Príklad komunikácie

SERVER



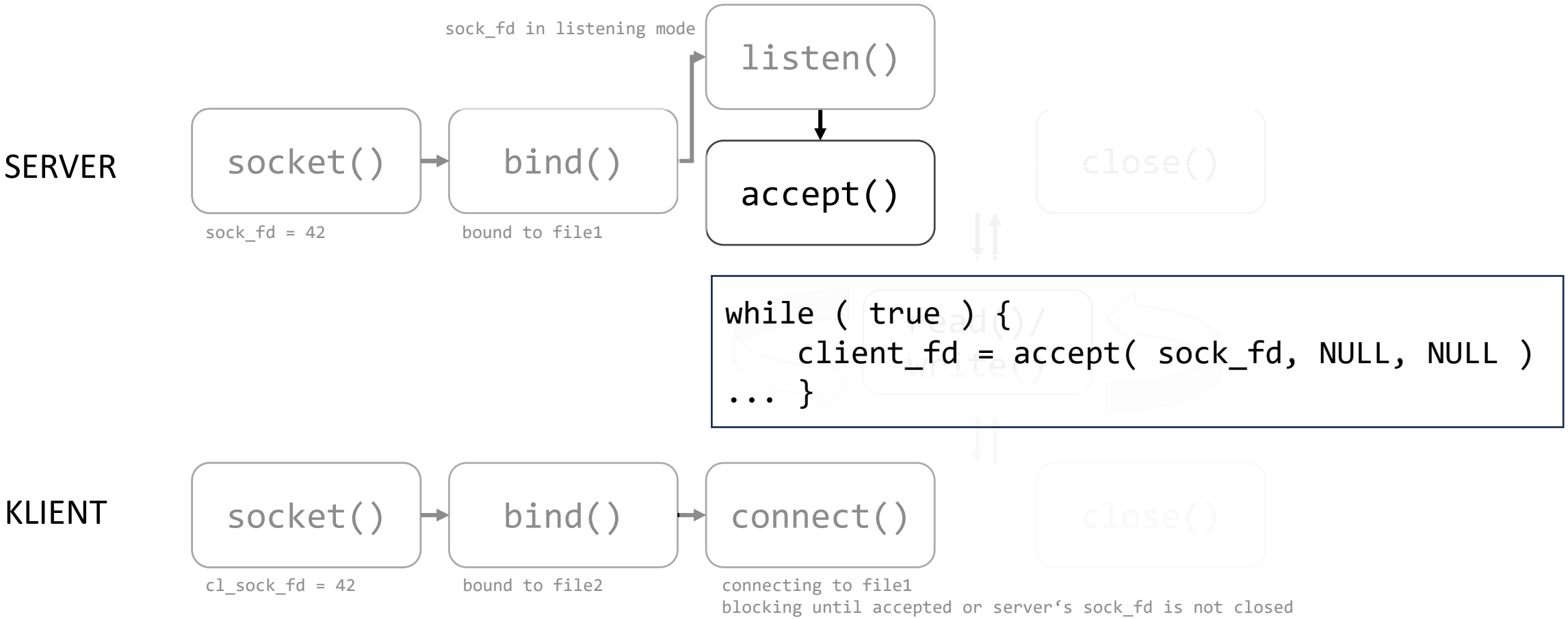

```
struct sockaddr_un addr = {
    .sun_family = AF_UNIX,
    .sun_path = "file1"};

connect( cl_sock_fd,
         (const struct sockaddr *)
         &addr,sizeof addr );
```

KLIENT



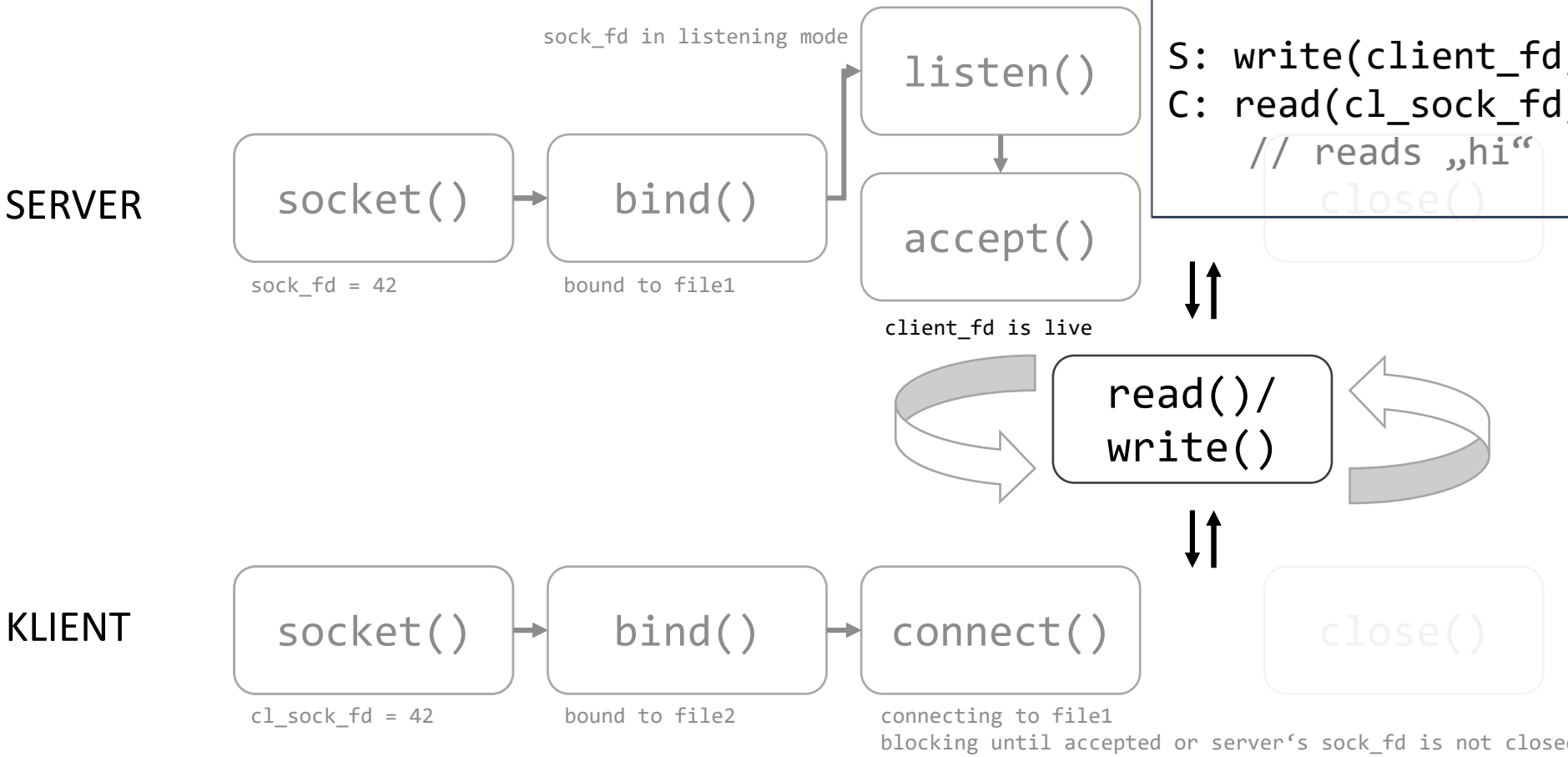
Príklad komunikácie



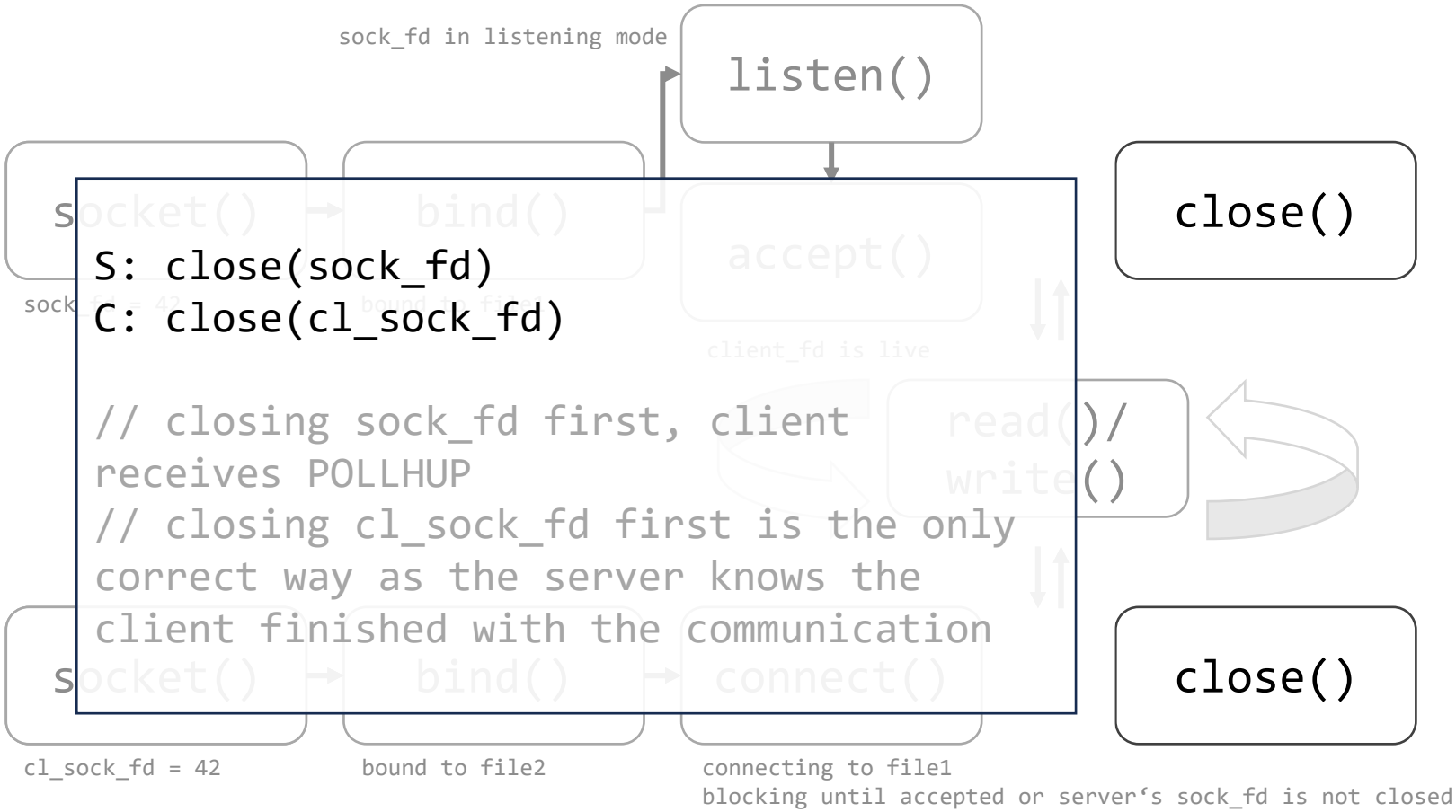
Príklad komunikácie

```
C: write(cl_sock_fd, "hello", ...);
S: read(client_fd, ...);
    // reads „hello“

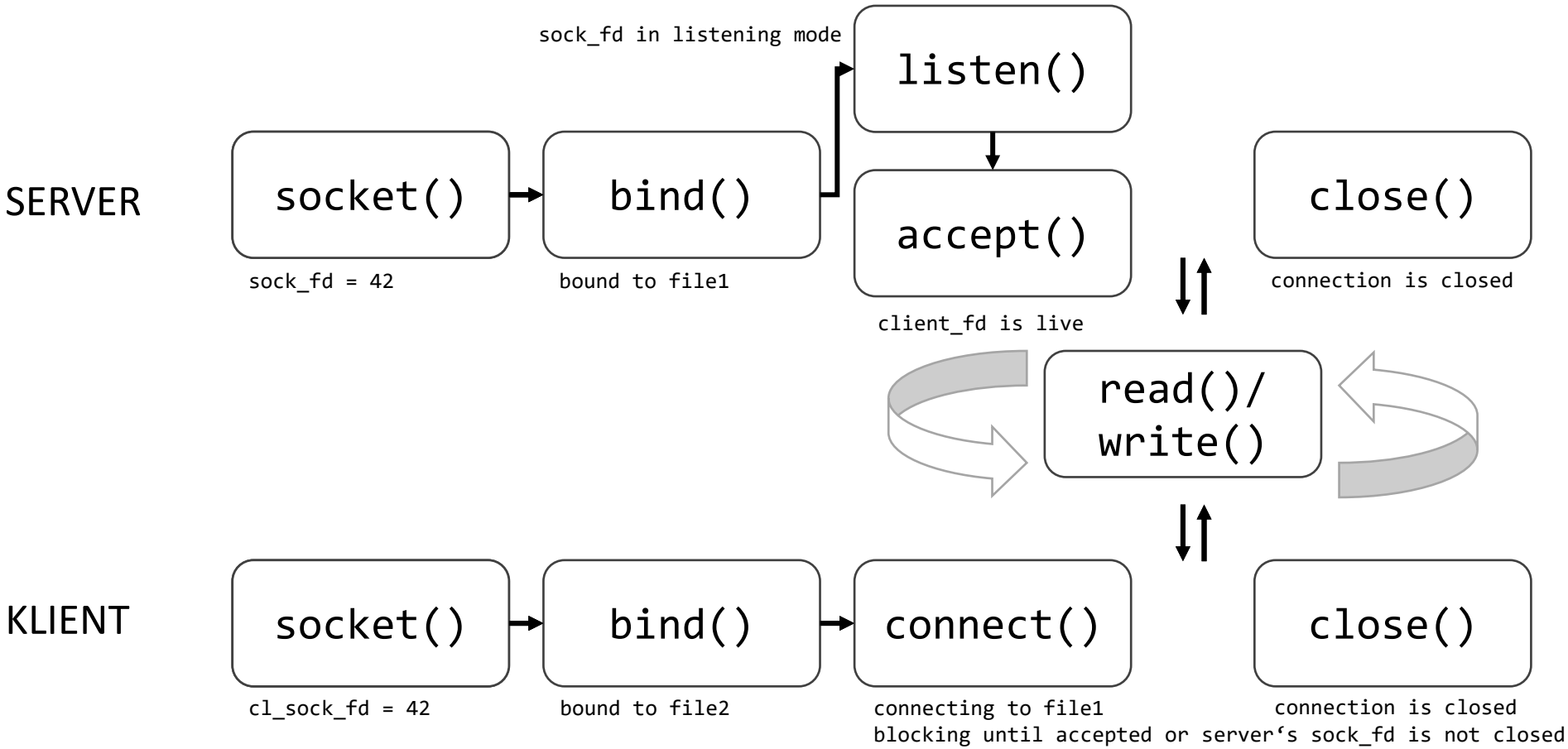
S: write(client_fd, „hi", ...);
C: read(cl_sock_fd, ...);
    // reads „hi“
```



Príklad komunikácie



Príklad komunikácie



Príklad komunikácie

