



Počítačové komunikace a sítě
Dokumentace k 1. projektu

Varianta 1: Klient-server pro získání informace o uživateli

Obsah

1	Popis zadání	2
2	Implementace	2
2.1	Server	2
2.1.1	Návratové kódy	2
2.2	Klient	2
2.2.1	Návratové kódy	2
3	Spuštění	3
3.1	Příklady spuštění	3
4	Další informace	4
	Zdroje	5

1 Popis zadání

Úkolem bylo navrhnout a implementovat klient-server aplikaci v C/C++ realizující zprostředkování informací o uživateli na serveru. Aplikační protokol získává informace o uživateli ze souboru `/etc/passwd`.

2 Implementace

Jsou implementovány dvě aplikace, jedna pro klienta a druhá pro server. Aplikace jsou napsané v jazyce C podle nejnovějšího standardu C11. Síťová komunikace využívá TCP protokol a je realizována pomocí BSD socketů. Jsou podporovány pouze IPv4 adresy.

2.1 Server

Server je spuštěn s jedním argumentem a to s číslem portu na kterém má běžet. Nejprve se ověří korektnost tohoto argumentu, poté se otevře pro čtení soubor `/etc/passwd` a inicializují se zdroje pro síťovou komunikaci s klientem. Server v nekonečném cyklu čeká na navázání spojení klienta, popř. klientů. Server dokáže obsloužit více klientů zároveň. Nejprve dostane přes socket zprávu ve které klient specifikuje, které informace požaduje. Ty server poté začne číst a postupně je klientovy přes socket odesílat. Server je ukončen zasláním signálu `SIGINT`. Na `STDOUT` nic netiskne, pouze v případě chyby tiskne chybové hlášení na `STDERR`.

2.1.1 Návrátové kódy

- 0 – Korektní ukončení aplikace (signál `SIGINT`)
- 10 – Chybné zadání parametrů aplikace
- 20 – Chyba při otevírání souboru `/etc/passwd`
- 40, 41, 43, 44, 45, 46 – Chyba při volání funkcí pro síťovou komunikaci (`socket()`, `bind()`, `listen()`, `accept()`, `send()`, `recv()`)
- 99 – Chyba alokace paměti

2.2 Klient

Klient vyžaduje 3 argumenty, ip adresu nebo host/domain name serveru, číslo portu na kterém server běží a poslední argument pro specifikaci dat, které bude požadovat. Po ověření korektnosti těchto argumentů klient inicializuje komunikaci se serverem. Připojí se k němu a přes socket mu zašle zprávu, ve které upřesní které informace si žádá. Klient poté přijímá přes socket zprávy od serveru s požadovanými informacemi a ty tiskne na `STDOUT`. V případě chyby se ukončí a vytiskne chybové hlášení na `STDERR`.

2.2.1 Návrátové kódy

- 0 – Korektní ukončení aplikace
- 1 – Žádná shoda pro uvedený login
- 10 – Chybné zadání parametrů aplikace
- 30 – Chyba při konvertování host/domain name na ip adresu

- 40, 42, 45, 46 – Chyba při volání funkcí pro síťovou komunikaci (`socket()`, `connect()`, `send()`, `recv()`)
- 99 – Chyba alokace paměti

3 Spuštění

Server

```
$ ./ipk-server -p port
```

port – číslo portu

Klient

```
$ ./ipk-client -h host -p port [-n|-f|-l] login
```

host – host/domain name nebo ip adresa

port – číslo portu

-n – zjištění jména pro zadaný parametr login

-f – zjištění domovského adresáře pro zadaný parametr login

-l – vypísání všech loginů, parametr login zde není povinný, je-li uveden, funguje jako prefix

3.1 Příklady spuštění

```
bash:~$ ./ipk-server -p 6666
bash:~$ ./ipk-client -h localhost -p 6666 -l m
```

```
-----
man
mail
messagebus
```

```
eva~> ./ipk-server -p 1234
bash:~$ ./ipk-client -h eva.fit.vutbr.cz -p 1234 -f xknet100
```

```
-----
/homes/eva/xk/xknet100
```

```
xdusek27@merlin:~$ ./ipk-server -p 4242
bash:~$ ./ipk-client -h merlin.fit.vutbr.cz -p 4242 -n xhosal00
```

```
-----
Hosala Martin,FIT BIT 2r
```

```
eva~> ./ipk-server -p 9876
xdusek27@merlin:~$ ./ipk-client -h eva.fit.vutbr.cz -p 9876 -l xkuk
```

```
-----
xkukam00
xkukan00
xkukli03
xkukos00
xkukuc04
```

4 Další informace

Program se sestavuje pomocí utility `make` z následujících zdrojových souborů:

`client.c`

`client.h`

`server.c`

`server.h`

`error.c`

`error.h`

Celkem se jedná o 825 řádků zdrojového kódu. Velikost výsledných binárních souborů je 18,7 kB pro `ipk-server` a 14,2 kB pro `ipk-client`.

Zdroje

- [1] Petr Břelík, TCP [online], 2014-06-10, <https://www.sallyx.org/sally/c/linux/tcp>, [cit. 2018-03-12]
- [2] K Hong, SOCKETS - SERVER & CLIENT - 2018 [online], http://www.bogotobogo.com/cplusplus/sockets_server_client.php#block_vs_non_blocking, [cit. 2018-03-12]
- [3] Daniel Scocco, Example of Client-Server Program in C (Using Sockets and TCP) [online], 2014-04-22, <https://www.programminglogic.com/example-of-client-server-program-in-c-using-sockets-and-tcp/>, [cit. 2018-03-12]