

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IPK – Projekt 1
Klient–server pro získání informace
o uživatelích

Obsah

1	Úvod	2
2	Implementace	2
2.1	Aplikační protokol	2
3	Demonstrace činnosti	2
3.1	Server	2
3.2	Client	2

1 Úvod

Zadáním bylo vytvořit a implementovat v jazyce C/C++ aplikační protokol ke komunikaci klient – server, jež bude na základě vstupních argumentů klienta zjišťovat různá data o uživateli uložená v souboru `/etc/passwd` unixových/linuxových systémů a tisknout je na standardní výstup.

2 Implementace

V konstruktoru vytvoříme socket typu `SOCK_STREAM`, tedy bude fungovat na protokolu `TCP/IP`. Jako doménu zvolíme `PF_INET[1]` pro `IPv4` protokol a připojíme se funkcí `connect`. Data jsou odesílána funkcí `send` a přijímána pomocí `recv`. Server dokáže obsloužit více požadavků díky dvojení procesů pomocí funkce `fork`. Funkcí `listen[2]` je nastavena délka fronty příchozích spojení k danému socketu na 10. Pokud je tato fronta plná a další klient žádá o spojení se serverem, obdrží chybový kód `ECONNREFUSED`.

2.1 Aplikační protokol

Klient odesílá požadavky na server ve formátu `<Hlavička>EOL<Flagy>EOL<Data>`. Flagy jsou nastaveny v závislosti na vstupních argumentech klienta. Server zkontroluje, zda se jedná o můj protokol, pomocí zkontrolování hlavičky. V případě že se jedná o můj protokol pomocí funkce `getpwent[3]` načítá jednotlivé řádky souboru `/etc/passwd` a jejich data ukládá do položek struktury `passwd`, z níž už je jednoduché získat potřebná data. Data odesílaná zpět klientovi jsou odeslána najednou. Klient pouze vypisuje data na standardní výstup.

Při chybě na serveru (například neplatný login) vrací server chybovou hlášku a klient se tímto ukončí, zatímco server pouze vypíše chybovou hlášku. Data jsou jak na straně klienta tak na straně serveru kódována pomocí algoritmu **jehož zdrojový kód jsem nepsal já**[4] a je uložen v souborech `base64.{cpp|hpp}`. (Pozn.: Ano, je to hloupost, ale na funkčnost to nemá žádný vliv :)) Tento zdrojový kód je volně použitelný viz hlavička kódu – odkaz v literatuře.

3 Demonstrace činnosti

3.1 Server

Spuštění:

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

Například:

```
./ipk-server -p 55555
```

3.2 Client

Spuštění:

```
./ipk-client -h host -p port [-nl-fl-l] login
```

Například:

```
./ipk-server -p 55555 -h localhost -n xkocic01
```

Výstup: Kocica Filip,FIT BIT 2r

```
./ipk-server -p 55555 -h localhost -f xkocic01
```

Výstup: /homes/eva/xk/xkocic01

```
./ipk-server -p 55555 -h localhost -l xkocic01
```

Výstup: xkocic01

```
./ipk-server -p 55555 -h localhost -l xkoci
```

Výstup:

xkocia18

xkocic01

xkocij01

xkocin01

Reference

- [1] *System calls: socket* [online]. [cit. 2018-02-28]. Dostupné z: <http://beej.us/net2/html/syscalls.html>
- [2] *Linux man page: listen* [online]. [cit. 2018-02-28]. Dostupné z: <https://linux.die.net/man/2/listen>
- [3] *Encoding and decoding base 64 with c++: Base64* [online]. [cit. 2018-02-28]. Dostupné z: <https://renenyffenegger.ch/notes/development/Base64/Encoding-and-decoding-base-64-with-cpp>
- [4] *Linux man page: getpwent* [online]. [cit. 2018-02-28]. Dostupné z: <http://man7.org/linux/man-pages/man3/getpwent.3.html>