

# **Počítačové komunikace a sítě**

2017/2018

Projekt 1 – Klient-server pro zjišťování informací o uživateli

Zdeněk Brhel  
xbrhel04

**12. března 2018**

# Obsah

1 Zadání.....	3
2 Protokol.....	4
3 Formát paketu.....	4
4 Struktura programu.....	5
5 Reference.....	5

# 1 Zadání

1) Seznámit se s kostrami kódů pro programování klientských a serverových síťových aplikací za použití BSD socketů a navrhnout vlastní aplikační protokol realizující přenos informací o uživateli na straně serveru a relevantní informace k projektu uveďte v dokumentaci.

2) Vytvořit klientskou a serverovou aplikaci realizující zprostředkování informace o uživateli na serveru.

## Konvence zapisování parametrů u klienta

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

- `./ipk-client` – Jméno aplikace
- `-h {host}` – Povinný parametr, host reprezentuje IP adresu serveru, nebo jeho DNS jméno
- `-p {port}` – Povinný parametr, port reprezentuje číslo cílového portu
- `[-l|-n|-f] {login}` – Je třeba vybrat si z trojice parametrů. V případě `-l` vypíše všechny jména začínající řetězcem `login`, pokud není řetězec zadán, vypíše všechny. V případě `-n` vypíše informace o uživateli, řetězec `login` je povinný parametr a nenalezne-li server takového uživatele, vypíše do konzole `user not found`. V případě `-f` vypíše cestu k uživatelskému domovskému adresáři.

př. - `./ipk-client -h merlin.fit.vutbr.cz -p 25565 -l xbrh`

## Konvence zapisování parametrů u klienta

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

- `./ipk-client` – Jméno aplikace
- `-h {host}` – Povinný parametr, host reprezentuje IP adresu serveru, nebo jeho DNS jméno
- `-p {port}` – Povinný parametr, port reprezentuje číslo portu, na kterém server naslouchá

př. - `./ipk-server -p 25565`

## 2 Protokol

Protokol slouží k přenosu informací o uživateli mezi klientem a serverem. Využívá protokol TCP.

Komunikaci zahajuje klient. Server mezitím čeká na vstup od klienta. Při neúspěšném připojení se na serveru i na klientovi vypíše chyba.

Po úspěšném připojení k serveru čeká server na odeslání paketu od klienta.

Po přijetí paketu server ho rozdělí na 2 části – příkaz a login (více v kapitole o paketu). Dle obsahu paketu server vykoná po něm požadovanou práci.

Jakmile server práci dokončí, odešle zpět výstup z předchozího kroku.

Po přijetí výstupu klientem ho klient vypíše na konzoli, následně ukončí připojení a uzavře se. Server své ukončení neřeší a dá se uzavřít pouze signálem SIGINT.

## 3 Paket

V případě klienta i serveru je paket řetězec, který se přenáší do bufferu, odkud se následně zpracovává.

Před vytvořením připojení si server vytvoří welcome socket, díky kterému se naváže připojení. Následně si vytvoří klientský socket, přes který posílá pakety.

Paket vypadá například takto:

Příkaz	Login
l	xb

Po přijetí tohoto paketu ho server zpracuje, následně připraví pro klienta paket obsahující výstup.

Paket vypadá například takto:

Řetězec

xborko02\nxbarno00\nxbrhel04\xbudac01
---------------------------------------

V případě chyby paket vypadá takto:

Řetězec

user not found
----------------

## 4 Struktura

- lpk-client.cpp – hlavní soubor pro překlad klienta
- client\_side.cpp – obsahuje třídu a metody pro klienta
- server\_side.cpp – obsahuje třídu a metody pro server
- parameters\_c.cpp – obsahuje třídu a metody pro klienta
- parameters\_s.cpp – obsahuje třídu a metody pro klienta

## 5 Reference

- Understanding /etc/passwd, Online: <https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>
- J. Postel, J. Reynolds, *FILE TRANSFER PROTOCOL (FTP)*, <https://tools.ietf.org/html/rfc959>