

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentácia k projektu do predmetu IPK

## **Implementácia aplikácie klient - server pre získanie informácií o užívateľoch**

Varianta I

12. března 2018

# Obsah

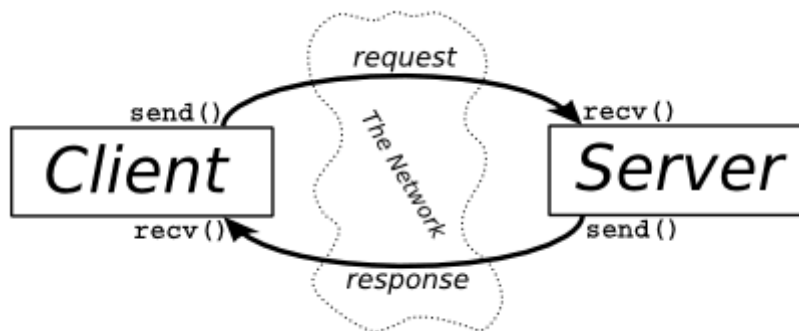
<b>1</b>	<b>Zadanie</b>	<b>2</b>
<b>2</b>	<b>Úvod do problematiky</b>	<b>2</b>
<b>3</b>	<b>Implementácia TCP komunikácie</b>	<b>2</b>
3.1	Vytvorenie spojenia . . . . .	2
3.2	Výmena informácií . . . . .	3
3.3	Ukončenie spojenia . . . . .	3
<b>4</b>	<b>Priblíženie súboru /etc/passwd</b>	<b>3</b>
<b>5</b>	<b>Aplikačný protokol</b>	<b>4</b>
<b>6</b>	<b>Popis implemetácie</b>	<b>5</b>
<b>7</b>	<b>Demonštrácia činnosti implementovaných aplikácií</b>	<b>5</b>
7.1	Konkrétne ukážky spustenia aplikácií s reálnymi výstupmi . . . . .	5

## 1 Zadanie

Cieľom projektu bolo naprogramovať ako klientskú, tak aj serverovú aplikáciu realizujúcu sprostredkovanie informácií o užívateľoch na serveru. Informácie o užívateľoch server získava zo súboru `/etc/passwd`.

## 2 Úvod do problematiky

Štandardná schéma komunikácie medzi dvoma procesami sa nazýva *model klient - server*. Klient i server sú aplikačné procesy, ktoré komunikujú cez sieťové rozhranie. Môže ísť aj o procesy bežiace na rovnakom počítači. Základnými činnosťami klienta je posielanie žiadostí o nejakú sieťovú službu. Server čaka na prichádzajúce požiadavky, prijíma ich, spracováva a posiela späť odpoveď.



Dôležité je, že spracovanie požiadavkov prebieha výhradne na strane servra. Klient iba predáva požiadavku a zobrazuje odpoveď. U modelu klient - server komunikáciu obvykle začína klient. Rovnako tak to je aj v mojej implementácii. Server beží v nekonečnej smyčke. Ukončenie servera je viazané na SIGINT signál. Komunikácia medzi serverom a klientom je popísaná tzv. *protokolom*. Protokol je súbor syntaktických a sémantických pravidiel určujúcich výmenu informácie medzi aspoň dvoma entitami<sup>1</sup>.

## 3 Implementácia TCP komunikácie

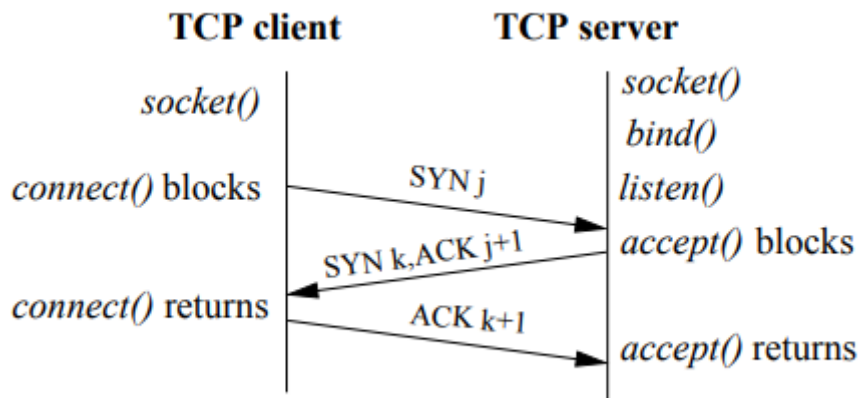
Činnosť TCP komunikácie možno rozdeliť na základné 3 časti:

- Vytvorenie spojenia TCP
- Komunikácia, výmena informácií
- Uzatvorenie spojenia

### 3.1 Vytvorenie spojenia

TCP vytvára spojenie pomocou mechanizmu *three - way handshake*, ktorý zahŕňa výmenu paketov SYN, SYN+ACK, ACK, vid' obrázok nižšie.

<sup>1</sup><http://www.cs.vsb.cz/grygarek/LAN/sockets.html>



Spojenie vyvoláva funkcia `connect()` na strane klienta. Na strane serveru čaká proces na prijatie spojenia funkciou `accept()`. Po výmene synchronizačných paketov je ustanovené spojenie. Na vytvorenie spojenia využívame nasledujúce funkcie:

- `socket()` - vytvorenie dátovej štruktúry schránka
- `bind()` - zviazanie schránky na strane serveru s konkrétnym portom, tzv pasívne otvorenie
- `connect()` - aktívne otvorenie sa strane klienta
- `listen()` - pasívne otvorenie na strane serveru, server čaká na spojenie
- `accept()` - prijatie spojenia

### 3.2 Výmena informácií

Na prenos dát sa používajú funkcie `read()` a `write()`, ktoré sa používajú na čítanie a zápis. Odpoveď z funkcie `read()` môže byť kratšia ako veľkosť bufferu, do ktorej sa dáta ukladajú, preto je treba čítanie vykonávať opakovane, kým funkcia nevráti hodnotu 0 (EOF), alebo -1 (chyba čítania).

### 3.3 Ukončenie spojenia

Ukončenie spojenia väčšinou iniciuje klient. Z pohľadu servera sa jedná skôr o pasívne uzatvorenie komunikácie, čiže, keď sa klient požiada, tak mu odpoviem. Na uzatvorenie sa používajú funkcie `close()` a `shutdown()`.

## 4 Priblíženie súboru `/etc/passwd`

Súbor `/etc/passwd` slúži na ukladanie základných informácií, ktoré sú vyžadované pri prihlasovaní. Obsahuje jeden záznam na riadok pre každého užívateľa (alebo užívateľský účet) systému. Záznam je rozdelený do siedmich polí pričom jednotlivé polia sú od seba oddelené znakom `:`.<sup>2</sup>

```

oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash
  ↓   ↓   ↓   ↓   ↓           ↓           ↓
  1   2   3   4   5           6           7

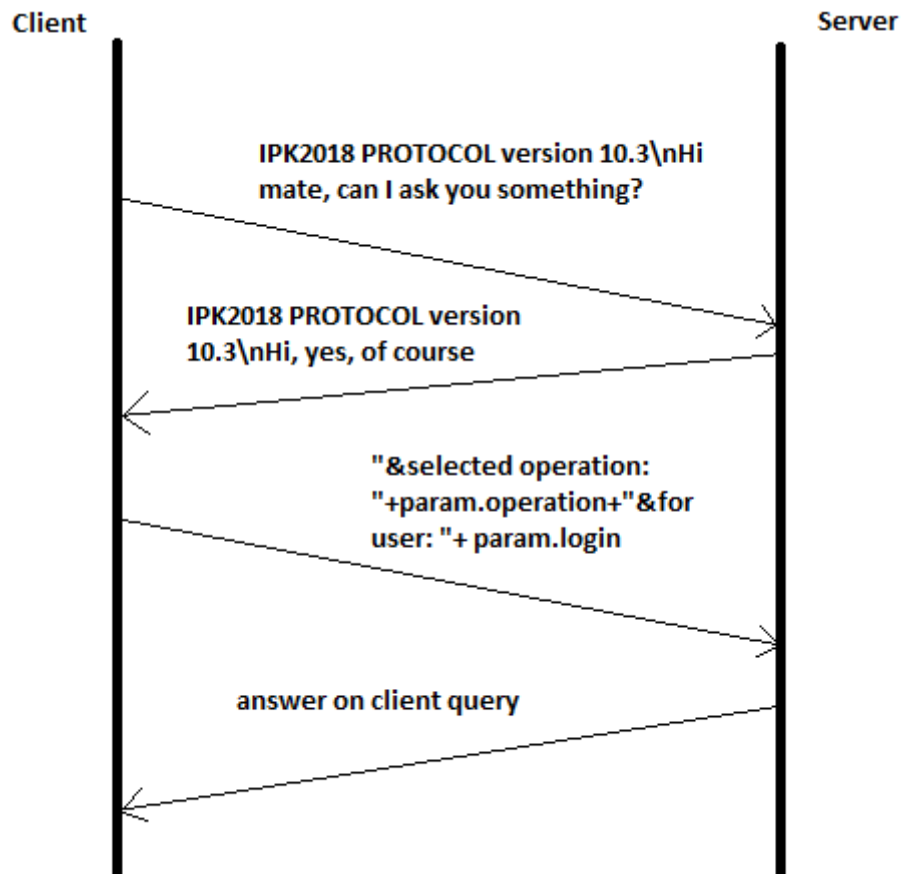
```

<sup>2</sup><https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>

1. Username
2. Password
3. User ID (UID)
4. Group ID (GID)
5. User ID Info
6. Home directory
7. Command/shell

Pri práci na projekte sme využívali polia Username, Home directory a User ID Info.

## 5 Aplikačný protokol



Navrhnutý aplikačný protokol ma nasledovne definovanú štruktúru. Klient odošle na server správu v tvare "IPK2018 PROTOCOL version 10.3\nHi mate, can I ask you something". Server mu odpovie správou v tvare "IPK2018 PROTOCOL version 10.3\nHi, yes, of course". Ak všetko sedí, klient nasledovne pošle serveru dotaz na požiadavku v tvare "&selected operation: "+param.operation+"&for user: "+param.login) , pričom param.operation znamená vybranú operáciu, teda:

- **-n** značí, že bude vrátené plné meno užívateľ a vrátane prípadných ďalších informácií pre uvedený login (User ID Info)
- **-f** značí, že budú vrátené informácie o domácom adresári užívateľ a pre uvedený login (Home directory)
- **-l** značí, že bude vrátený zoznam všetkých užívateľov, v tomto prípade je login nepovinný, ak je ale zadany bude slúžiť ako prefix pre výber užívateľov

a param.login značí vybraný login. V prípade operácie **-l** je v param.login uložený prefix, podľa ktorého sa má vyhľadávať. Následná odpoveď servera je výsledok, ktorý klient zobrazuje na štandardný výstup, viď sekcia Demonštrácia činnosti implementovaných aplikácií.

## 6 Popis implemetácie

Na vytvorenie spojenia som vytvoril u oboch aplikácií funkcie connection(), kde vykonávam všetky nevyhnutné náležitosti, ktoré sú popísané vyššie.

Implementáciu servera som rozšíril o **SO\_REUSEADDR** a **SO\_REUSEPORT**, čo zabezpečí, že v prípade ak server "zomrie" (ukončenie exitom pri chybe) môže nový spustený server využiť rovnakú adresu a port.

```

81
82     int reuse = 1;
83     if (setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (const char*)&reuse, sizeof(reuse)) < 0){
84         cerr<<"Reuse addr error\n";
85         exit(EXIT_FAILURE);
86     }
87     #ifdef SO_REUSEPORT
88     if (setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (const char*)&reuse, sizeof(reuse)) < 0){
89         cerr<<"Reuse port error\n";
90         exit(EXIT_FAILURE);
91     }
92     #endif

```

Komunikáciu začína vždy klient, ktorý sa dotazuje o vypísanie jednotlivých častí zo súboru **/etc/passwd**. Server po tom, ako získa dotaz, skontroluje jeho validitu a analyzuje požadované informácie. V prípade voľby **-f**, alebo **-n** využíva server funkcie knižnice **pwd.h**, ktorá na základe loginu umožňuje nájsť pre daného užívateľ a jeho domovský adresár, prípadne jeho celé meno a nejaké ďalšie informácie. V prípade, že sa daný užívateľ (konkrétne jeho login) v súbore **/etc/passwd** nenachádza je na štandardný chybový výstup vypísané, že sa užívateľ a nepodarilo nájsť. Pri voľbe **-l** sa prechádza celý súbor riadok po riadku a vypisujeme prvé polia každého záznamu. Možno je taktiež vyhľadávanie na základe predpony(prefixu), kedy sa vypíšu len užívatelia, ktorých loginy začínajú danou predponou. V prípade, ak zadaná predpona nie je zhodná so žiadnym loginom, alebo je súbor **/etc/passwd** prázdny, nevypíše sa na štandardny výstup nič a program skončí s návratovým kódom 0.

## 7 Demonštrácia činnosti implementovaných aplikácií

- Server spúšťaťame nasledovne: **./server-ipk -p portnumber**, kde portnumber značí číslo portu
- Klienta spúšťaťame nasledovne: **./client-ipk -l servername -p portnumber [-n|-f|-l] login**, kde servername značí názov servera, portnumber značí číslo portu a login značí login užívateľ a

### 7.1 Konkrétne ukážky spustenia aplikácií s reálnymi výstupmi

- **./client-ipk -h eva.fit.vutbr.cz -p 9999 -n xlisci01**  
Očakávame celé meno užívateľ a s loginom xlisci01

```
kristian@kristian-VirtualBox: ~  
eva ~/IPK> ./ipk-client -h eva.fit.vutbr.cz -p 9999 -n xlisci01  
Liscinsky Kristian,FIT BIT 3r  
eva ~/IPK>
```

- `./client-ipk -h eva.fit.vutbr.cz -p 9999 -f xlisci01`

Očakávame informácie o domovskom adresári užívateľa s loginom xlisci01

```
kristian@kristian-VirtualBox: ~  
eva ~/IPK> ./ipk-client -h eva.fit.vutbr.cz -p 9999 -f xlisci01  
/homes/eva/xl/xlisci01  
eva ~/IPK>
```

- `./client-ipk -h eva.fit.vutbr.cz -p 9999 -l xlis`

Očakávame zoznam všetkých užívateľov, ktorých login ma predponu xlis

```
kristian@kristian-VirtualBox: ~  
eva ~/IPK> ./ipk-client -h eva.fit.vutbr.cz -p 9999 -l xlis  
xlisci01  
xlisci02  
xlisie00  
xliska16  
xlisti00  
eva ~/IPK>
```