

# **Klient-server pro získání informace o uživateli**

Počítačové komunikace a sítě – Projekt 1

# Obsah

Úvod.....	2
Popis.....	2
Implementace.....	2
Možností spuštění.....	2
Možnosti spuštění serveru.....	2
Možnosti spuštění klienta.....	3
.....	4
Protokol.....	4
Zdroj.....	5

# Úvod

Tento projekt zprostředkovává komunikaci mezi serverem a klientem pomocí vlastně navrženého aplikačního protokolu určeného k získání informací o uživateli serveru.

## Popis

K získání požadovaných informací se využívá soubor `/etc/passwd`, kde se uchovávají informace o všech existujících uživatelských účtech. Pomocí parametru spuštění `-f` můžeme zjistit podrobnější informace o daném uživateli, pomocí parametru `-n` lze zase získat cestu k jeho domovské složce a pomocí parametru `-l` je možné vypsat seznam všech uživatelů nebo jen jejich části, obsahující daný prefix.

## Implementace

Při úspěšném vykonání běhu je program ukončen s návratovým kódem 0 a výstupem na `STDOUT`. Pokud při běhu nastane libovolná chyba, program končí s návratovým kódem 1 a chybovou hláškou v anglickém jazyce ve formě „*ERROR: Popis chyby*“ vypsanou na `STDERR`.

Pokud je výsledkem dotazu více než-li jedna položka, je každá z důvodu přehlednosti vypsaná na nový řádek.

Server po spuštění běží nepřetržitě dokud není jeho činnost ukončena signálem. Po připojení klienta se proces rozdělí na dva, původní stále vyčkává na připojení dalšího klienta a druhý proces zprostředkuje komunikaci s připojeným klientem a po odeslání výsledku se ukončí. Klientský program taktéž po vypsaní výsledku sám svou činnost ukončí.

## Možností spuštění

Klientský program nabízí více možností vstupních parametrů specifikujících požadovanou informaci, kterou má server poslat. Kdežto server poskytuje jedinou možnost.

Parametr má vždy tvar přepínač a povinný či nepovinný argument, oddělený mezerou. Přepínačem se rozumí pomlčka a určitý znak (konkrétně třeba `-l`) a argument je řetězec následující hned za mezerou za přepínačem. Konkrétní přepínač lze využít pouze jednou.

## Možnosti spuštění serveru

Jediný vstupní parametr serverové aplikace je specifikace portu, který bude využívat. Tento parametr je určen povinným přepínačem `-p` a argument je povinný a očekává validní neobsazené číslo portu, jinak při běhu skončí chybově. Obecný tvar spuštění aplikace je tedy:

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

Konkrétní případ spuštění může vypadat například takto:

```
./ipk-server -p 56565
```

## Možnosti spuštění klienta

Vstupní parametry klientského programu jsou mírně pokročilejší. Je jich hned několik, ale na jejich pořadí nezáleží.

Za přepínačem *-h* je třeba povinně specifikovat cílovou IP adresu (např. *192.168.0.1*) nebo jméno hostitele (např. *Localhost*), který bude na informace dotazován.

Přepínač *-p* stejně jako při spuštění klienta očekává jako povinný argument validní číslo portu, na kterém bude komunikace probíhat.

Následně je nutné vybrat jeden z přepínačů *-n*, *-f* a *-l*. První dva mají argument povinný a specifikuje, kterého uživatelského účtu se dotaz týká. U třetího přepínače lze argument využít k definování prefixu uživatelů nebo lze tento argument vynechat a bude vypsán celý seznam.

Obecný tvar spuštění klienta je následující:

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

Konkrétní případy mohou vypadat takto:

Dotaz:

```
./ipk-client -h merlin.fit.vutbr.cz -p 56565 -f xfurda00
```

Výstup:

```
/homes/eva/xf/xfurda00
```

Dotaz:

```
./ipk-client -h merlin.fit.vutbr.cz -p 56565 -n xfurda00
```

Výstup:

```
Furda Jiri,FIT BIT 2r
```

Dotaz:

```
./ipk-client -h merlin.fit.vutbr.cz -p 56565 -l xfuc
```

Výstup:

```
xfucik03
```

```
xfucik07
```

```
xfucik08
```

Může také nastat situace, kdy nebude možné vypsat žádnou položku. Takový stav je považován za chybu a výstup vypadá následovně:

Dotaz:

```
./ipk-client -h merlin.fit.vutbr.cz -p 56565 -n nesmysl
```

Výstup:

```
ERROR: No result to be shown
```

## Protokol

Komunikace mezi serverem a klientem probíhá použitím vlastního aplikačního protokolu. Každá položka je oddělena novým řádkem. Řídící zprávy, krom prvním dotazu na server, jsou vždy čtyřmístné řetězce obsahující dvě velká písmena označující anglickou zkratku popisující akci a je oddělena vykřičníkem před i po této zkratce (např. !OK!). Po vyfiltrování těchto řídících zpráv je možné bez dalších úprav přijaté zprávy rovnou vypisovat.

První zpráva přichází od klienta a server jej očekává ve tvaru *login:flag* (např. *xfurda00:n*), kde login nese informaci, o který účet se klient zajímá (u možností *-n* a *-f*). U možnosti *-l* zase nese informaci o prefixu pravděpodobně více uživatelských účtů, které se mají dohledat. Tato možnost nabízí validní použití i bez specifikace loginu (tedy pouze *:l*).

Odpověď serveru na tento dotaz je na prvním řádku buď *!OK!* značící že je vše v pořádku a na dalším řádku už následují informace, na které se klient dotazoval. Druhý případ je odpověď formou *!NF!* symbolizující nenalezení požadované informace (not found). Oba tyto případy končí řetězcem *!EN!* reprezentující konec komunikace (end).

Ukázka úspěšného dotazu

```
KLIENT: xfurda00:n  
SERVER: !OK!  
SERVER: /homes/eva/xf/xfurda00  
SERVER: !EN!
```

Ukázka neúspěšného dotazu

```
KLIENT: nesmysl:n  
SERVER: !NF!  
SERVER: !EN!
```

## Zdroj

Pro referenci komunikace mezi serverem a klientem jsem místo poskytnutých demo příkladů zvolil návod pana Roberta Ingallse dostupný na internetové adrese

<http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>