



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Počítačové komunikace a sítě Dokumentace k projektu 1

Varianta 1: Klient-server pro získání informace o uživateli

Marek Kovalčík

xkoval14, 196248

8. březen 2017

Obsah

1	Zadání	2
2	Trocha teorie	2
3	Popis řešení aplikačního protokolu	3
3.1	Implementace klientské aplikace	3
3.2	Implementace serverové aplikace	3
4	Závěr	3
4.1	Zhodnocení	3
5	Přílohy	4
5.1	./ipk-server -p 5000 běží na serveru merlin.fit.vutbr.cz	4
5.2	./ipk-client -h merlin.fit.vutbr.cz -p 5000 -n xkova14	4
5.3	./ipk-client -h merlin.fit.vutbr.cz -p 5000 -f rysavy	4
5.4	./ipk-client -h merlin.fit.vutbr.cz -p 5000 -l a	4
6	Použité zdroje	5

1 Zadání

Zadáním tohoto projektu bylo vytvořit aplikaci v jazyce C/C++ pro klienta a server s použitím BSD socketů pro přenos informací o uživateli/uživateli ze serveru.

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

- host (IP adresa nebo fully-qualified DNS name) identifikace serveru jakožto koncového bodu komunikace klienta; port (číslo) cílové číslo portu;
- -n značí, že bude vráceno plné jméno uživatele včetně případných dalších informací pro uvedený login (User ID Info);
- -f značí, že bude vrácena informace o domácím adresáři uživatele pro uvedený login (Home directory);
- -l značí, že bude vrácen seznam všech uživatelů, tento bude vypsán tak, že každé uživatelské jméno bude na zvláštním řádku; v tomto případě je login nepovinný. Je-li však uveden bude použit jako prefix pro výběr uživatelů.
- login určuje přihlašovací jméno uživatele pro výše uvedené operace.

Spuštění serveru: `./ipk-server -p port`

- port (číslo) číslo portu, na kterém server naslouchá na připojení od klientů.

2 Trocha teorie

Síťový socket (anglicky network socket) je v informatice koncový bod připojení přes počítačovou síť. S rozvojem Internetu většina komunikace mezi počítači používá rodinu protokolů TCP/IP. Vlastní přenos zajišťuje IP protokol, a proto je používáno i označení internetový socket. Vlastní socket je handle (abstraktní odkaz), který může program použít při volání síťového rozhraní pro programování aplikací (API), například ve funkci „odeslat tato data na tento socket“. Sockety jsou vnitřně často jen celá čísla, která odkazují do tabulky aktivních spojení.

Počítačové procesy, poskytující aplikační služby, jsou označovány jako servery a při startu vytvářejí sockety, které jsou ve stavu naslouchání (listen). Tyto sockety čekají na iniciativu od klientských programů.

TCP server může obsluhovat více klientů současně pomocí podřízeného procesu vytvořeného pro každého klienta a navázání TCP spojení mezi podřízeným procesem a klientem. Pro každé spojení jsou vytvořeny unikátní specializované sockety. Když je zaváděno socket-to-socket virtuální spojení nebo virtuální okruh (VC), také známo jako TCP relace, pomocí dálkového socketu, což poskytuje duplexní proud bytů, jsou tyto unikátní specializované sockety ve stavu spojeno (connected).[3]

3 Popis řešení aplikačního protokolu

3.1 Implementace klientské aplikace

Pro spuštění klienta je nutné zadat správně argumenty, v opačném případě se klient nespustí. Před začátkem komunikace musí být server již spuštěný. Komunikace vždy začíná klient odesláním welcome socketu na server.

Jakmile je spojení navázáno, odešle klient zprávu, ve které první písmeno označuje požadovanou operaci a zbytek zprávy označuje hledaný login či prefix. V případě přepínače -l se loginy přijímají po jednom s tím, že klient vždy pošle zprávu zpět serveru o úspěšném přijetí aby server mohl pokračovat v posílání. Jakmile server odešle poslední login a přijde mu potvrzující zpráva, odešle zprávu end-communication , klient ji obdrží a komunikace se uzavře.

Pokouší-li se klient připojit na nějaký neočekávaný server, např. -h merlin.fit.vutbr.cz -p 80 (webový server) a nedostává žádnou odpověď, nebude čekat nekonečně dlouho, ale ukončí se po 5 vteřinách.

3.2 Implementace serverové aplikace

Serverová aplikace musí být spuštěná před tím, než se na ni klient/i začnou připojovat. Pro spuštění je třeba zadat přepínač -p a číslo portu. Budou-li argumenty zadány nesprávně, server se nespustí.

Po spuštění server čeká v nekonečné smyčce na požadavky klientů a v jeden moment je schopen obsluhovat více klientů, tzn., že je konkurentní. Konkurence schopnost je zabezpečená pomocí procesů.

Server obdrží od klienta takovou zprávu, že první písmeno určuje funkci, která se má provést a zbytek zprávy určuje hledaný login nebo prefix. (např. nxkova14 nebo frysavy nebo l (v případě žádného prefixu))

V případě nenalezení shody v souboru /etc/passwd na serveru se vrátí informační zpráva o nenalezení.

V případě, že klient žádá o list uživatelů s určitým prefixem, každý login je posílán samostatně a server zpátky od klienta obdrží zprávu o úspěšném přijetí aby mohl pokračovat.

4 Závěr

4.1 Zhodnocení

Tento aplikační protokol je funkční, splňuje body v zadání a odchyťává spoustu neočekávaných stavů a vypořádá se s nimi. Celkově jsem se svým řešením spokojen.

5 Přílohy

Klienta i server jsem testoval v mnoha varintách. Spuštění serverové i klientské aplikace na domácím počítači, spuštění serveru na vzdáleném serveru a klienta na domácím počítači nebo spuštění obou aplikací na vzdáleném serveru. Výstupy fungovaly tak jak měly, avšak pro přehlednost a stručnost dokumentace zde uvádím jen pár demonstračních příkladů.

5.1 `./ipk-server -p 5000` běží na serveru merlin.fit.vutbr.cz

```
xkoval14@merlin: ~/IPK/projekt1$ make
gcc -std=gnu99 -Wall -Werror -Wextra -pedantic -pthread -o ipk-client ipk-client.c -lm
gcc -std=gnu99 -Wall -Werror -Wextra -pedantic -pthread -o ipk-server ipk-server.c -lm
xkoval14@merlin: ~/IPK/projekt1$ ./ipk-server -p 5000
Connection to closed
INFO: New connection:
INFO: Client address is ::ffff:147.229.208.143
INFO: Client port is 33962
Received from client: nxkoval14
Bude se vypisovat plne jmeno uzivatele: xkoval14
Send to client: Kovalcik Marek,FIT BIT 2r
Connection to ::ffff:147.229.208.143 closed
```

5.2 `./ipk-client -h merlin.fit.vutbr.cz -p 5000 -n xkoval14`

```
marek@marek-Lenovo-G505s:~/Dropbox/VUT/2. letni semestr/IPK - Počítačové komunikace a
sítě/projekty/projekt1$ ./ipk-client -h merlin.fit.vutbr.cz -p 5000 -n xkoval14
Kovalcik Marek,FIT BIT 2r
```

5.3 `./ipk-client -h merlin.fit.vutbr.cz -p 5000 -f rysavy`

```
marek@marek-Lenovo-G505s:~/Dropbox/VUT/2. letni semestr/IPK - Počítačové komunikace a
sítě/projekty/projekt1$ ./ipk-client -h merlin.fit.vutbr.cz -p 5000 -f rysavy
/homes/kazi/rysavy
```

5.4 `./ipk-client -h merlin.fit.vutbr.cz -p 5000 -l a`

```
marek@marek-Lenovo-G505s:~/Dropbox/VUT/2. letni semestr/IPK - Počítačové komunikace a
sítě/projekty/projekt1$ ./ipk-client -h merlin.fit.vutbr.cz -p 5000 -l a
adm
annot01
annot02
annot03
annot04
annot05
annot06
annot07
annot08
annot09
annot10
avahi
```

6 Použité zdroje

[1] Demonstrační soubory, Demo_C.zip [online]. upraveno: 2018-02-22 [cit. 2018-03-07]. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIPK-IT%2Fother&cid=11963>

[2] Počítačové komunikace a sítě, PDF prezentace [online]. [cit. 2018-03-07]. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2017L-02-APLIKACE.pdf&cid=11963>

[3] Síťový socket, WIKIPEDIA, [online], upraveno: 14.1.2017 [cit 2018-03-07]. Dostupné z: <https://cs.wikipedia.org/wiki/S>