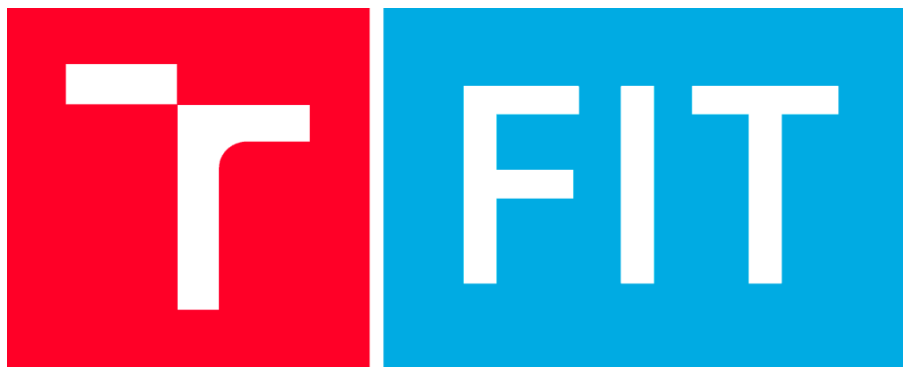


Vysoké učení technické v Brně

Fakulta informačních technologií



Počítačové komunikace a sítě – 1.projekt
Klient-server pro získání informace o uživateli

V Brně dne 12. 3. 2018

Peter Kapičák (xkaptic02)

Obsah

Zadanie	3
Teória	3
Klient-server model	3
BSD socket	3
Komunikačný protokol	3
Klient	3
Server	3
Implementácia	4
Príklad činnosti aplikácie	4
Referencie	5

Zadanie

Vypracovať jednoduchú klient-server aplikáciu. Navrhnuť vlastný komunikačný protokol realizujúci prenos informácií o užívateľoch na strane serveru zo zložky */etc/passwd*.

Naprogramovať klientskú aj serverovú časť v jazyku C/C++.

Teória

Klient-server model

Klient v tomto modeli posiela požiadavky na server o dáta alebo službu. Na druhú stranu server vybavuje požiadavky na dáta alebo službu jednému alebo viacerým klientom.

BSD socket

Komunikácia medzi klientom a serverom prebieha prostredníctvom portu. Ak odosielateľ pošle správu na port, na ktorom nikto nečaká na správu, tak správa bude zahodená. Aplikácie vytvárajú sockety, ktorých prostredníctvom je umožnený prenos správy na konkrétny port.

Komunikačný protokol

Klient

Po nadviazaní komunikácie medzi klientom a serverom odošle klient požiadavku, ktorá bude obsahovať:

- login užívateľa na základe, ktorého majú byť vyhľadávané informácie.
- Informáciu (príznak), čo má byť na strane servera vyhľadávané

Po odoslaní požiadavky je klient pripravený na prijímanie informácií. Ak server odoslal, všetky vyžiadané informácie, respektíve sa žiadne nenašli, tak dostane klient správu, že posielanie informácií zo serveru sa skončilo. Túto správu majú obe strany napevno určenú.

Server

Server po prijatí požiadavky, čo má hľadať otvorí súbor */etc/passwd*. Server nájdené informácie ukladá postupne do bufferu, ktorý má veľkosť 1500B ako segment v normálnej komunikácii. Ak sa buffer zaplní, tak server odošle správu (buffer) klientovi. Potom buffer vyprázdni a pokračuje vo vyhľadávaní. Ak server prišiel na koniec súboru, respektíve nenašiel požadovanú informáciu odošle server klientovi správu, že posielanie skončilo.

Implementácia

Zaujímavou časťou implementácie bola práca so samotným súborom, ktorá spočívala v načítaní jedného riadku súboru. Riadok obsahuje viacero informácií o užívateľovi a bolo potrebné separovať konkrétnu informáciu, ktorá bola požadovaná od klienta.

Informácií sa do bufferu zmestí len obmedzené množstvo. Vždy sa nájde informácia len o jednom užívateľovi a pred pridaním do bufferu sa skontroluje dĺžka bufferu spolu s novou informáciou a keď dĺžka presahuje stanovenú dĺžku bufferu musí sa informácia na strane servera rozdeliť do dvoch a viac bufferov.

Aby server dokázal obslúžiť viac požiadaviek súčasne som vyriešil prostredníctvom vlákien.

Príklad činnosti aplikácie

Ak klient odošle požiadavku na server napríklad:

```
./ipk-client -h merlin.fit.vutbr.cz -p 5555 -n xkaptic02
```

Pre server to znamená, že bude hľadať riadok s loginom *xkaptic02*. Login je vždy prvá informácia na riadku podľa konvencie súboru */etc/passwd* a ak nájde zhodu, tak sa server na danom riadku posunie na informáciu, ktorú si klient vyžiadal prepínačom *-n*. Tento prepínač znamená, že má byť nájdené celé meno užívateľa.

Príklad riadku:

```
xkaptic02:x:29733:251:Kapicak Peter,FIT BIT 2r:/homes/eva/xk/xkaptic02:/bin/ksh
```

V prípade nájdania zhody server odošle správu, ktorú klient vypíše na STDOUT.

Výsledok na výstupe:

```
Kapicak Peter,FIT BIT 2r
```

Referencie

Ryšavý, O. (dátum neznámy). Prednáška - Sít'ové aplikace (VUT FIT).

www.en.wikipedia.org. (04. 03 2018). Dostupné na Internet: Wikipedia:

https://en.wikipedia.org/wiki/Client-server_model

www.wiki.treck.com. (25. 03 2010). Dostupné na Internet:

http://wiki.treck.com/Introduction_to_BSD_Sockets