FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

IPK – Počítačové komunikace a sítě
Projekt č.1 - Zjištění informací o uživateli

12.3.2018 Tomáš Nereča(xnerec00)

Obsah

1 Zadanie **2**

2 Základné informácie **2**

3 Popis riešenia **2**
3.1 Klient 2
3.2 Server 2

4 Zaujímavosti **3**

5 Príklady použitia **3**

6 Zdroje **4**

1 Zadanie

Zoznámiť sa s kostrami kódov pre programovanie sieťových aplikácií (klient, server) za použitia *BSD socketov*. Navrhnúť vlastný aplikačný protokol realizujúci prenos informácií o používateľoch a naprogramovať klientskú a serverovú aplikáciu.

2 Základné informácie

Komunikáciu medzi serverom a klientom zabezpečuje protokol *TCP*, nakoľko bolo potrebné zabezpečiť bezpečný prenos informácií bez poškodenia dát. TCP komunikácia zahŕňa nadviazanie spojenia. Pri nadväzovaní spojenia prebehne tzv. *three-way handshake*. Ak je všetko v poriadku, začnú sa odosielať dáta medzi klientom a serverom. Na základe zadaných vstupných argumentov pri spúšťaní klienta sa vytvorí správa - *request*, ktorá je odoslaná na server. Server následne v súbore **etc/passwd** vyhľadá požadované údaje a odošle správu obsahujúcu tieto údaje klientovi, prípadne prázdnu správu ak sa údaje nepodarilo nájsť.

3 Popis riešenia

3.1 Klient

Kontrolu vstupných argumentov v oboch aplikáciách zabezpečuje funkcia GETOPT(). V prípade nesprávnej kombinácie, prípadne chýbajúcich argumentov aplikácia vypíše nápovedu a skončí.

Po kontrole argumentov sa vytvorí request, ktorý obsahuje hlavičku v tvare `<--xnerec00_protocol-->`. Ďalej obsahuje číselný kód zvoleného argumentu(1, 2 alebo 3) a prípadne **login**. Jednotlivé časti sú oddelené znakom **&**.

Následne sa klient pokúsi nadviazať TCP komunikáciu so serverom a odoslať request. K tomu je potrebné vykonať nasledujúce kroky:

- vytvoriť *client socket* pomocou funkcie SOCKET()
- vyhladať *DNS* serveru a preložiť, prípadne skontrolovať prítomnosť serveru na zadanej *IPv4* adrese pomocou funkcie GETHOSTBYNAME()
- pripojiť sa na server pomocou funkcie CONNECT()
- odoslať request pomocou funkcie SEND()

Po úspešnom vykonaní týchto krokov je klient pripravený príjimať dáta zo serveru. Dáta su ukladané do *bufferu*. Prijatá správa je ihneď vytlačená na štandardný výstup. Prijímanie dát zabezpečuje funkcia RECV() vo WHILE cykle. Ak prijaté dáta nenaplnia buffer, signalizuje to, že server už odoslal všetky dáta. Po vyskočení z cyklu sa uzavrie client socket pomocou funkcie CLOSE() a aplikácia sa ukončí.

3.2 Server

Po validácii PORTU bude server čakať na request. K tomu je potrebné vykonať nasledujúce kroky:

- vytvoriť *welcome socket* pomocou funkcie SOCKET()
- priradiť adresu k welcome socketu pomocou funkcie BIND()
- pomocou funkcie LISTEN() označiť welcome socket ako pasívny (bude použitý na akceptovanie requestu)

Server teraz v nekonečnom cykle čaká na request. Funkcia ACCEPT() pri nadviazaní spojenia vytvorí *komunikačný socket* a prijme request pomocou funkcie RECV().

Po rozparsovaní requestu sa vytvorí správa, ktorá bude odoslaná klientovi. Request musí obsahovať správnu hlavičku. Správa bude odosielaná vo WHILE cykle po bufferoch veľkosti 1024B. Po odoslaní celej správy sa komunikačný socket uzavrie a server čaká na ďalší request.

Server vybavuje requesty až do prijatia signálu SIGINT. Po jeho odchytení sa uzavrie welcome socket a aplikácia sa ukončí.

4 Zaujímavosti

Pri argumente -p prebieha ihneď kontrola, či bol zadaný validný port pomocou funkcie FIND_FIRST_NOT_OF().

Ak pri argumente -h je prvý znak číslo, kontroluje sa, či bola zadaná korektná IPv4 adresa pomocou funkcie INET_PTON().

Client socket má nastavený timeout na 10s pomocou funkcie SETSOCKOPT()) s parametrom SO_RCVTIMEO. Ak po uplynutí 10s nepríde odpoveď zo serveru, aplikácia sa ukončí.

Aby nedochádzalo k problému, že ihneď po vypnutí serveru sa nedá spustiť server na rovnakom porte, je po vytvorení welcome socketu zavolaná funkcia SETSOCKOPT()) s parametrom SO_REUSEADDR. Tým pádom sa povolí re-bindovanie na port, ktorý bol použitý predtým.

Vyhľadávanie v súbore **etc/passwd** a následné parsovanie riadku zabezpečujú C++ funkcie pracujúce so STRINGAMI. Jednotlivé riadky zo súboru sa získavajú funkciou GETLINE(). Prvá časť každého riadku sa porovná so zadaným **loginom**, a v prípade zhody sa v cykle odrezáva začiatok riadku po znak **:**. Nakoniec sa odreže požadovaný SUBSTRING.

Nakoniec bolo treba vyriešiť situáciu, ktorá nastávala pri odosielaní veľkého množstva dát. Niekedy nastal problém v prijímaní dát a klient odoslal serveru signál SIGPIPE, čo spôsobilo ukončenie serverovej aplikácie. Klient teda po prijatí bufferu odošle serveru správu **OK** a server odosiela ďalší buffer až po úspešnom prijatí tejto správy.

5 Príklady použitia

Zobrazenie nápovedy pri nesprávnych vstupných argumentoch:

```
xnerec00@merlin: ~/SKOLA/ipk/proj1$ ./ipk-server 30100
Invalid parameter!
 Usage: ./ipk-server -p port
```

```
eva ~/SKOLA/ipk/proj1> ./ipk-client -h merlin -p 30100 -n xnerec00 -a
Invalid parameters!
 Usage: ./ipk-client -h host -p port [-n|-f|-l] login
        ./ipk-client -h host -p port -l
```

Spustenie serveru:

```
xnerec00@merlin: ~/SKOLA/ipk/proj1$ ./ipk-server -p 30100
```

Spustenie klienta s parametrom -n:

```
eva ~/SKOLA/ipk/proj1> ./ipk-client -h merlin -p 30100 -n xnerec00
Nereca Tomas,FIT BIT 2r
```

Spustenie klienta s parametrom -f:

```
eva ~/SKOLA/ipk/proj1> ./ipk-client -h merlin -p 30100 -f xnerec00
/homes/eva/xn/xnerec00
```

Spustenie klienta s parametrom -l a prefixom:

```
eva ~/SKOLA/ipk/proj1> ./ipk-client -h merlin -p 30100 -l xner
xnerec00
xnerud01
```

Spustenie klienta s parametrom -l bez prefixu (na obrázku len niekoľko riadkov):

```
eva ~/SKOLA/ipk/proj1> ./ipk-client -h merlin -p 30100 -l
adm
annot01
annot02
annot03
```

6 Zdroje

Pri riešení projektu som sa snažil čo najviac držať materiálov k predmetu. Pri návrhu TCP komunikácie som využil úryvky kódu z prednášky 2.

Niektoré úryvky kódu sú inšpirované odpoveďami na diskusných fórach. Adresa na fórum je v tom prípade v komentári.

Ďalej som čerpal z rôznych manuálových stránok, napr.:

http://man7.org/linux/

https://linux.die.net/man/

http://www.cplusplus.com/reference/