Sveučilište u Zagrebu
PMF – Matematički odjel

Mreže računala

Vježbe 03

Matko Botinčan
Zvonimir Bujanović
Igor Jelaska
Maja Karaga

Prije početka...

- programiramo pod UNIX (Linux) operacijskim sustavom
- programi komuniciraju na udaljenim računalima – potrebno ih je nekako prenijeti do udaljenog računala, tamo kompajlirati i pokrenuti

- ftp – prijenos datoteka između lokalnog i udaljenog računala
- telnet – izvršavanje naredbi (npr. kompajliranje & pokretanje programa) na udaljenom računalu

telnet

- protokol definiran na aplikacijskom sloju
- udaljeno računalo redovito koristi port 23

- korištenje:
  - `telnet imeUdaljenogStroja`
  - unesemo login i password; unosimo naredbe koje se izvršavaju na udaljenom stroju; po završetku rada napišemo `exit`

- telnet ne koristi enkripciju podataka prilikom prenošenja → iz sigurnosnih razloga često nije dozvoljen takav pristup udaljenom računalu

ssh (*Secure SHell*)

- sigurna zamjena za telnet, koristi šifriranje podataka
- udaljeno računalo redovito koristi port 22

- korištenje:
  - `ssh login@imeUdaljenogStroja`
  - unesemo password; unosimo naredbe na udaljenom stroju; po završetku rada napišemo `exit`

- Windows-i ne dolaze sa *ssh-klijentom* – možemo koristiti npr. putty (http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe)

Zadatak 1

- pomoću naredbe `finger` ispišite popis svih korisnika koji su ulogirani na računalo za kojim sjedite u praktikumu
- iz praktikuma je dozvoljen ssh samo na ostala računala u lokalnoj mreži i na računalo `student`. Odabrerite proizvoljno računalo iz lokalne mreže i pogledajte tko je sve logiran na njemu. Provjerite i tko je sve logiran na računalu `student`.

ftp (*File Transfer Protocol*)

- još jedan protokol iz aplikacijskog sloja
- udaljeno računalo redovito koristi port 21
- također nesiguran protokol koji ne koristi enkripciju
- zamjena je sftp (*SSH File Transfer Protocol*)
- sftp ne dolazi sa Windows-ima...postoje brojni programi (*ftp-klijenti*) koji koriste grafičko sučelje
  - FileZilla (http://filezilla-project.org/)
  - psftp (http://the.earth.li/~sgtatham/putty/latest/x86/psftp.exe)

ftp / sftp

- pokretanje (s)ftp klijenta
  - `ftp imeUdaljenogRacunala`
  - `sftp login@imeUdaljenogRacunala`
- interne naredbe unutar (s)ftp klijenta
  - `?` ili `help` – popis svih internih naredbi
  - `ls` – ispis svih datoteka u trenutnom direktoriju na udaljenom računalu
  - `lls` – ispis svih datoteka u trenutnom direktoriju na lokalnom računalu
  - `pwd` – ime trenutnog direktorija na udaljenom računalu
  - `lpwd` – ime trenutnog direktorija na lokalnom računalu
  - `cd imeDirektorija` – promjena direktorija na udaljenom računalu
  - `lcd imeDirektorija` – promjena direktorija na lokalnom računalu

ftp / sftp

- interne naredbe unutar (s)ftp klijenta
  - `get imeDatoteke` – prenosi datoteku imeDatoteke iz trenutnog direktorija na *udaljenom* računalu u trenutni direktorij na *lokalnom* računalu
  - `put imeDatoteke` – prenosi datoteku imeDatoteke iz trenutnog direktorija na *lokalnom* računalu u trenutni direktorij na *udaljenom* računalu
  - `exit` ili `bye` – izlazak iz (s)ftp klijenta

ftp / sftp

- tekstualne datoteke na Windows-ima i UNIX-ima imaju različite oznake za kraj retka – takve datoteke (*.txt, *.c, *.h, *.html i slične) treba prenositi na poseban način
- ftp (ali ne i sftp!) ima 2 posebne interne naredbe
  - `ascii` – iduće datoteke koje sudjeluju u prijenosu sa get i put će biti tekstualne i napravit će se korekcija oznaka za krajeve retka (ako je potrebno)
  - `binary` – iduće datoteke koje sudjeluju u prijenosu će biti binarne (npr. *.jpg, *.mp3 i slične) i ne treba raditi korekciju oznake kraja redaka
- sftp uvijek prenosi datoteke binarno

Zadatak 2

- na udaljenom računalu (asistent će vam reći ip-adresu, te korisničko ime i password) dostupna je ftp usluga
- prekopirajte datoteku `slika.jpg` sa udaljenog na svoje računalo. Pogledajte pristiglu datoteku.
- na svom računalu napravite tekst-datoteku koja se zove `login.txt` (login zamijenite svojim korisničkim imenom), te ju prenesite na udaljeno računalo. Provjerite je li datoteka stigla.

Mrežno programiranje

- programiramo u aplikacijskom sloju, za ostale se brinu operacijski sustav i hardware
- aplikacijski sloj u internetskom protokolarnom stogu obuhvaća 3 sloja referentnog modela:
  - sloj sesije – uspostava konekcije između 2 računala
  - prezentacijski sloj – konverzija podataka tako da budu pogodni za prijenos preko mreže
  - aplikacijski sloj – konkretan način komunikacije između 2 računala, svojstven samo našoj aplikaciji
- koristit ćemo SocketAPI za programiranje u mrežnom sloju – to je kolekcija struktura podataka i funkcija u programskom jeziku C namjenjena mrežnom programiranju

Header datoteke

- svaka funkcija iz SocketAPI treba neku header datoteku
- jednostavnosti radi, možemo uvijek uključiti sve koje bi nam mogle zatrebati:

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netdb.h>
#include <netinet/in.h>
#include <unistd.h>
#include <arpa/inet.h>
```

Rad sa mrežnim adresama

- 3 načina reprezentacije adresa:
  - *host-name* – npr. www.google.com
  - IP-adresa u dekaskom zapisu – npr. 192.84.105.1
  - IP-adresa u 32-bitnom binarnom zapisu
- interno se sav rad sa adresama mora odvijati u 32-bitnom binarnom zapisu
- postoje funkcije za konverziju iz jednog zapisa u drugi

Rad sa mrežnim adresama

- binarni IP-zapis čuva se u specijalnoj strukturi:

```
struct in_addr
{
   unsigned long s_addr;
};
```

- host-name i dekadski IP-zapis čuvaju se kao stringovi

dekadski IP → binarni IP

```
int inet_aton(
       const char *dekadskiIP,
       struct in_addr *binarniIP );
```

Povratna vrijednost: 0 ako dekadska adresa nije valjana, ne-nula inače.

Mnemotehnika: inet = internet, a = ASCII, n = network

Primjer:

```
char  dekadskiIP[] = "161.53.8.14";
struct in_addr binarniIP;

if( inet_aton( dekadskiIP, &binarniIP ) == 0 )
   printf( "%s nije dobro zadana adresa\n", dekadskiIP );
```


binarni IP → dekadski IP

```
char *inet_ntoa( struct in_addr binarniIP );
```

Povratna vrijednost: pointer na memoriju unaprijed alociranu unutar funkcije. Na toj adresi je dekadska IP-adresa.

Mnemotehnika: inet = internet, a = ASCII, n = network

Primjer:

```
char  dekadskiIP[] = "161.53.8.14";
struct in_addr binarniIP;

if( inet_aton( dekadskiIP, &binarniIP ) == 0 )
   printf( "%s nije dobro zadana adresa\n", dekadskiIP );

char natragUDekadskiIP[20];
strcpy( natragUDekadskiIP, inet_ntoa( binarniIP ) );
printf( "%s\n", natragUDekadskiIP );
```

host-name ↔ binarni IP

- koristi se specijalna struktura:

```
struct hostent {
   char *h_name;
   char **h_aliases;
   int h_addrtype;
   int h_length;
   char **h_addr_list;
};
#define h_addr h_addr_list[0]
```

- h_name – službeni host-name računala
- h_aliases – polje alternativnih host-name-ova računala (zadnji je NULL)
- h_addrtype – tip adrese, kod nas uvijek AF_INET
- h_length – duljina adrese u byteovima, kod nas uvijek 4
- h_addr_list – polje binarnih zapisa IP-adresa računala (zadnja je NULL)
- h_addr – prvi binarni zapis u gornjem polju

host-name → binarni IP

```
struct hostent *gethostbyname( const char *hostName );
```

Prima host-name kao string, vraća popunjenu hostent strukturu ili NULL ako je došlo do greške. U tom slučaju, poziv funkcije herror će na ekran ispisati poruku o greški. Struktura koja se vraća je unaprijed alocirana unutar funkcije gethostbyname!

Primjer:

```
struct hostent *hostInfo;
hostInfo = gethostbyname( "www.yahoo.com" );
if( hostInfo == NULL ) herror( "gethostbyname" );

printf( "Sluzbeni host-name: %s\n", hostInfo->h_name );
struct in_addr binarniIP=*((struct in_addr *)hostInfo->h_addr);
char *dekadskiIP = inet_ntoa( binarniIP );
printf( "Dekadska IP-adresa: %s\n", dekadskiIP );
```


binarni IP → host-name

```
struct hostent *gethostbyaddr(
   const char *binarniIP, int duljina, int tipAdrese );
```

- duljina = sizeof(binarniIP)
- tipAdrese = AF_INET (za nas, inače može biti i nešto drugo...)
- povratna vrijednost – popunjena hostent struktura

Primjer:

```
struct hostent *hostInfo;
struct in_addr binarniIP;
inet_aton( "161.53.8.14", &binarniIP ); // error-check...
hostInfo = gethostbyaddr(
  (const char *)&binarniIP, sizeof( binarniIP ), AF_INET );
if( hostInfo == NULL ) herror( "gethostbyaddr" );

printf( "Sluzbeni host-name: %s\n", hostInfo->h_name );
char *dekadskiIP = inet_ntoa(
   *((struct in_addr *)hostInfo->h_addr) );
printf( "Dekadska IP-adresa: %s\n", dekadskiIP );
```


Zadatak 3

- Napišite program koji se ponaša slično mrežnom alatu nslookup.
- Program sa komandne linije treba dobiti host-name nekog računala.
- Program treba ispisati sve host-name-ove i sve IP-adrese tog računala.
- Ako host-name nije bio dobar, program treba ispisati poruku o greški.