STUB РУТИНА

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <signal.h>

#include <stdio.h>

#include <netdb.h>

#include <stdlib.h>

#include <string.h>

#define DAJ\_VERZIJA 1

#define DAJ\_BLOK 2

#define BR\_SERV 10

struct sockaddr\_in servaddr, raddr;

struct hostent \*hp;

struct servent \*sp;

struct poraka

{

 int tip;

 int block;

 char datoteka[10]; // ime na datotekata

 int verzija; // verzijata na datotekata koja jai ma serverot

 char sodrzina [512]; // sodrzina na blokot koj se ispraka

} msg;

int len;

int citaj(char \*dat, int blok, char \*blok1)

{

 struct dserv

 {

 sockaddr\_in addr;

 char ime[10];

 } tabela[BR\_SERV]; // prethodno popolneta tabela vo koja za sekoj server

 // se cuvaat adresite na serverite

int verzija[BR\_SERV]; // koja verzija na datotekata ja ima sekoj server

int counter, s, i, j, maxver;

 int sent\_req; // flag spored koj se znae dali do serverot veke e ili ne

 // e isprateno baranje za verzija

 s=socket(AF\_INET, SOCK\_DGRAM, 0);

 counter=0; // broj na server od koi e dobien odgovor

 msg.block=blok; // koj blok od datotekata se bara

 msg.tip=DAJ\_VERZIJA;

 strcpy(msg.datoteka, dat, 10); // ime na datotekata koja se bara

 len=sizeof(struct sockaddr\_in);

 // do site server se isprakja poraka DAJ\_VERZIJA

 i = 0;

 sent\_req = 0;

 while(i<BR\_SERV)

 {

 verzija i]=0; //minimalna verzija za vo slucaj da ne dobieme odgovor

 if (!sent\_req)

 {

 sendto(s, &msg, sizeof(msg), 0, &tabela[i].addr, len);

 sent\_req = 1;

 }

 alarm(20);

 st=recvfrom(s, &msg, sizeof(msg), 0, &raddr, &len);

 if(st==-1)

 {

 i++;

 sent\_req = 0;

 continue; // serverot so reden broj i ne e dostapen vo

 // momentot, pa se preminuva na sleden server

 }

 // sporedba dali dobieniot odgovor e od serverot na koj posledno mu

 // se ispratilo baranje (sporedba na adresata na serverot vo tabelata

 // I na adresata od koja pristignala porakata (raddr)

 if (memcmp(&raddr, &tabela[i].addr, sizeof(raddr))==0)

 {

 counter++;

 verzija[i] = msg.verzija;

 i++;

 sent\_req = 0;

 }

 else

 continue; // povtorno cekanje na odgovor od istiot server

 }//for

 if (counter>BR\_SERV/2) // prosto mnozinstvo od server se dostapni

 {

 maxver=0; // se bara maksimalnata (najnovata verzija)

 for (i=0;i< BR\_SERV; i++)

if (verzija[i]>maxver) maxver=verzija[i];

 // poraka od tip DAJ\_BLOK mu prakame na onoj server koj ja ima

 // najnovata verzija

 for (j=0;j< BR\_SERV;j++)

 {

 if (verzija[j]==maxver)

 {

 msg.tip=DAJ\_BLOK;

 strcpy(msg.dat, datoteka,10);

while(1)

{

 sendto(s, &msg, sizeof(msg), 0, &tabela[j].addr, len);

alarm(20);

 int r=recvfrom(s, &msg, sizeof(msg), 0, &raddr, &len);

 if(r>0 && (memcmp(&raddr, &tabela[j].addr,

sizeof(raddr))==0)

 break; //dobiena e poraka od serverot so max verzija

}

break; // za izlez od for ciklusot

 }//if

 }//for j

 memcpy(blok1, msg.sodrzina, 512);

 return(1);

 } //if counter

 else // nema prosto mnozinstvo na dostapni serveri

 return (-1);

}// citaj