Functions


```
/**********************************************************************

 CLIENT-SERVER su socket di tipo STREAM


 C  L  I  E  N  T

 *********************************************************************/

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>


#include "ftpdefines-230702.h"



struct protoent *pp;

char    buf[BUFSIZE];

#define SERVER1 "localhost"
#define SERVER2 "localhost"




/************
 Utilizzo:
 client nomefile
 ************/



main(argc,argv)
int     argc;char *argv[];
{
  int   i,s;
  char c;
  int fd,fd1,fd2,sock1,sock2;
  int rval,rval2;
  int reception_complete,server1_completed,server2_completed;
  struct sigaction  act;
  sigset_t sigmask,oldmask,currentsignals;
  fd_set read_fds;
  RICHIESTA_MSG rich_mesg1,rich_mesg2;
  RISPOSTA_MSG  risp_mesg;


/* Controllo argomenti */

if(argc != 2)
  {
    fprintf(stderr,"Uso: %s nomefile\n\n",argv[0]);
    exit(-1);
  }

/* Gestione segnali */

 sigemptyset( &sigmask);
 sigaddset(&sigmask, SIGINT);

 sigprocmask(SIG_BLOCK, &sigmask, &oldmask);


/* Socket e creazione connessioni */
```

```

sock1=  create_socket_sstream();

connect_to_server(sock1, SERVER1,SERVERPORT);

printf("Connected to the first server.\n");

sock2=  create_socket_sstream();

connect_to_server(sock2, SERVER2,SERVERPORT);

printf("Connected to the second server.\n");



 strncpy(rich_mesg1.filename,argv[1],RICHMSG_MAXPATHNAME);

 rich_mesg1.whichhalf= FIRSTHALF;

/* Invio comando RICHIESTA (GET) */

 if(write(sock1,&rich_mesg1,sizeof(RICHIESTA_MSG))<0)
   {
     perror("sending file request to the first server: ");
     exit(-3);
   }


strncpy(rich_mesg2.filename,argv[1],RICHMSG_MAXPATHNAME);
 rich_mesg2.whichhalf= LASTHALF;

 if(write(sock2,&rich_mesg2,sizeof(RICHIESTA_MSG))<0)
   {
     perror("sending file request to the second server: ");
     exit(-3);
   }

/* Apertura due file temporanei  */

if  ((fd1=open("/tmp/filetemp1", O_RDWR | O_CREAT |O_TRUNC, 0644))<0)
  {
   perror("creating tmp file1: ");
     exit(-4);
  }

if  ((fd2=open("/tmp/filetemp2", O_RDWR | O_CREAT |O_TRUNC, 0644))<0)
  {
   perror("creating tmp file2: ");
     exit(-4);
  }



/* Ricezione dai server */
 reception_complete=FALSE;
 server1_completed=FALSE;
 server2_completed=FALSE;

do
  {

 FD_ZERO(&read_fds);

 if(!server1_completed)
   FD_SET(sock1,&read_fds);

 if(!server2_completed)
   FD_SET(sock2,&read_fds);

 if  ((rval=select(sock2+1,&read_fds,NULL,NULL,NULL))<0)
   {
     perror("select: ");
     exit(-4);
   }

```

```
  if(FD_ISSET(sock1,&read_fds))
    {
      if((rval = read(sock1,buf,sizeof buf))<0)
        perror("receiving from server1");

      if(rval >0)
       {
         write(fd1,buf,rval);
         putchar('.');
       }
      else
        server1_completed=TRUE;
    }

  if(FD_ISSET(sock2,&read_fds))
    {
      if((rval = read(sock2,buf,sizeof buf))<0)
        perror("receiving from server1");

      if(rval >0)
       {
         write(fd2,buf,rval);
         putchar(',');
       }
      else
        server2_completed=TRUE;
    }

  reception_complete = server1_completed && server2_completed;

   }
  while(!reception_complete);


  sigpending(&currentsignals);

  if(sigismember(&currentsignals,SIGINT))
    {
      printf("Do you really want to remove the received file ?\n");
      c= getchar();

      if(c=='s' || c=='S'|| c=='y'|| c=='Y')
        {
          unlink("/tmp/file1");
          unlink("/tmp/file2");
          exit(0);
        }
    }

/* Chiediamo allo shell di concatenare le due parti  */
    system("cat /tmp/filetemp2 >>/tmp/filetemp1 ; mv /tmp/filetemp1 copiafile");



close(sock1);
 close(sock2);
close(fd1);
close(fd2);

exit(0);


}

int create_socket_sstream()
{
  int sock;


/* Crea una  socket di tipo STREAM per il dominio TCP/IP */
sock=   socket(AF_INET,SOCK_STREAM,0);

if(sock<0)
        {
        perror("opening stream socket");
```

```c
        exit(1);
        }
else
  return sock;
}


int connect_to_server(int sock,char *servername,int serverport)
{
struct  sockaddr_in     server;
struct  hostent *hp,*gethostbyname();


/* Ottiene l'indirizzo del server */

server.sin_family=      AF_INET;

hp=     gethostbyname(servername);

if(hp==0)
        {
        fprintf(stderr,"%s: unknown host",servername);
        exit(2);
        }


memcpy( (char *)&server.sin_addr, (char *)hp->h_addr ,hp->h_length);



server.sin_port= htons(serverport);

/* Tenta di realizzare la connessione */
printf("Connecting to server %s...\n",servername);
if(connect(sock,(struct sockaddr *)&server,sizeof server) <0)
        {
        perror("connecting stream socket");
        exit(1);
        }





}

```