```c
/*********************************************************************

 CLIENT-SERVER su socket di tipo STREAM


 S   E   R   V  E   R

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


#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <sys/timeb.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <errno.h>


#include "ftpdefines-230702.h"


char    buf[BUFSIZE];

/*********
 Utilizzo
 server
 **********/



main()
{
  int   sock,length;
  struct        sockaddr_in     server,client;
  char buff[512];
  int   s,msgsock,rval,rval2,i;
  struct        hostent *hp,*gethostbyname();


  /* Crea la  socket STREAM */
  sock= socket(AF_INET,SOCK_STREAM,0);
  if(sock<0)
    {   perror("opening stream socket");
    exit(1);
    }

  /* Utilizzo della wildcard per accettare le connessioni so ogni interfaccia*/
  server.sin_family =   AF_INET;
  server.sin_addr.s_addr= INADDR_ANY;
  server.sin_port = htons(SERVERPORT);
  if (bind(sock,(struct sockaddr *)&server,sizeof server)<0)
    {
      perror("binding stream socket");
      exit(1);
    }


  length= sizeof server;
  if(getsockname(sock,(struct sockaddr *)&server,&length)<0)
    {
      perror("getting socket name");
      exit(1);
    }

  printf("Socket port #%d\n",ntohs(server.sin_port));

  /* Pronto ad accettare connessioni */

  listen(sock,2);

```

```c
   do
     {
       /* Attesa di una connessione */

       msgsock= accept(sock,(struct sockaddr *)&client,(int *)&length);


       if(msgsock ==-1)
         perror("accept");
       else
           {
             if(fork()==0)
               {
                 printf("Serving connection from %s, port %d\n",
                        inet_ntoa(client.sin_addr),
                        ntohs(client.sin_port));



                 close(sock);
                 ftpserv(msgsock);
                 close(msgsock);
                 exit(0);
               }
             else
               close(msgsock);
           }

     }
 while(1);
 }


 ftpserv(int sock)
 {
 int s,rval,nread;
 int fd;
 struct stat fbuf;
 RICHIESTA_MSG rich_mesg;
 RISPOSTA_MSG  risp_mesg;
  int file_completed;

  s = 0;

  /* Ricezione del comando GET  */

  if((rval=read(sock,&rich_mesg,sizeof(RICHIESTA_MSG)))<0)
    {
      perror("reading client request");
      exit(-1);
    }

  if((fd= open(rich_mesg.filename,O_RDONLY))<0)
    {
      fprintf(stderr,"Non riesco ad aprire il file %s (%s)...uscita !\n",rich_mesg.filename,strerror(errno));
      exit(-1);
    }

    fstat(fd,&fbuf);

 /* Quale meta' del file vuole il cliente ? */
    if( rich_mesg.whichhalf == LASTHALF)
      lseek(fd,fbuf.st_size/2,SEEK_SET);

    file_completed=0;

  do {
    if(fbuf.st_size/2<sizeof(buf))
      nread= fbuf.st_size/2;
    else
      nread= sizeof(buf);

    if((nread = read(fd,buf,nread))<0)
          perror("reading from file");
```

Functions


```c

    s+= nread;

    if (nread >0)
         if((rval = write(sock,buf,nread))<0)
                   perror("writing on stream socket");

    if(rich_mesg.whichhalf == FIRSTHALF)
      file_completed= s==(fbuf.st_size/2);
    else
       file_completed= (nread==0);       /* LASTHALF */


 }
 while(!file_completed);




 }




```