

```
/* Svolgimento Compito 1 - 21/12/05 */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define DIM_BUFF 100
#define LENGTH_NAME 20
```

```
int main(int argc, char *argv[])
{
    int sd, nread, nwrite;
    char buff[DIM_BUFF], ok[LENGTH_NAME];
    struct hostent *host;
    struct sockaddr_in servaddr;
```

```
    /* CONTROLLO ARGOMENTI ----- */
    if(argc!=3)
    {
        printf("Error:%s serverAddress serverPort\n", argv[0]);
        exit(1);
    }
```

```
    printf("Client avviato\n");
```

```
    /* PREPARAZIONE INDIRIZZO SERVER ----- */
    memset((char *)&servaddr, 0, sizeof(struct sockaddr_in));
    servaddr.sin_family = AF_INET;
```

```
    host = gethostbyname(argv[1]);
    if (host == NULL)
    {
        printf("%s not found in /etc/hosts\n", argv[1]);
        exit(2);
    }
```

```
    servaddr.sin_addr.s_addr=((struct in_addr*) (host->h_addr))->s_addr;
    servaddr.sin_port = htons(atoi(argv[2])); // Controllare anche numero porta!!
```

```
    /* CORPO DEL CLIENT: */
    /* ciclo di accettazione di richieste di file ----- */
    printf("Richieste di visualizzazione fino alla fine del file di input\n");
    printf("Un tasto per visualizzare, EOF per terminare\n");
```

```
    while (gets(ok))
    {
```

```
        /* CREAZIONE E CONNESSIONE SOCKET (BIND IMPLICITA) ----- */
        /* in questo schema è necessario ripetere creazione, settaggio opzioni e connect */
        /* ad ogni ciclo, perchè il client fa una nuova connect ad ogni ciclo */
        sd=socket(AF_INET, SOCK_STREAM, 0);
        if (sd <0)
        {perror("apertura socket "); exit(3);}
        printf("Creata la socket sd=%d\n", sd);

        if (connect(sd,(struct sockaddr *) &servaddr, sizeof(struct sockaddr))<0)
        {perror("Errore in connect"); exit(4);}
        printf("Connect ok\n");
    }
```

```

/* chiudo l'output */
shutdown(sd, SHUT_WR);

while((nread=read(sd, buff, sizeof(buff)))>0)
{
    if ((nwrite=write(l, buff, nread))<0)
    {
        perror("write");
        break;
    }
}

printf("Chiudo connessione\n");
/* chiudo l'input */
shutdown(sd, SHUT_WR);

printf("Un tasto per visualizzare, EOF per terminare\n");

} //while

printf("\nClient: termino...\n");
exit(0);
}

```