

```
1  /*
2  * File:   MainClient.cpp
3  * Author: user
4  *
5  * Created on 18 dicembre 2008, 8.38
6  */
7
8  #include <stdlib.h>
9  #include "../Socket/ConnAll.h"
10
11  /*
12  *
13  */
14  char* removeTAGs(char*);
15
16
17  int main(int argc, char** argv) {
18      int server_port; //dichiarazione intero per la porta
19      char* completa,*sistemata,*rix,*msg,*supporto;//dichiarazione stringhe per
20          la pagina ricevuta completa, la pagina con i tag rimossi,pacchetto
21          ricevuto,nome pagina da richiedere
22      Address* ilMioServer; //indirizzo a cui faccio la richiesta
23      Connection* conny; //connessione generica
24      ClientTCP* myself; //client TCP attualmente in uso
25      if(argc!=4)//controllo sui parametri
26          errore("n° parametri errato:",argc);
27      server_port=str2int(*(argv+2)); //ottiene la porta dalla riga di comando e
28          la converte in intero
29      ilMioServer=new Address(server_port,*(argv+1)); //creazione indirizzo
30          destinatario
31      msg=joinStr("GET /", *(argv+3));
32      msg=joinStr(msg," "); //crea la stringa di richiesta al server
33      myself=new ClientTCP(0);
34      conny=myself->Connetti(ilMioServer); //stabilisci una connessione con il
35          server
36      myself->invia(msg);//invia richiesta
37      completa=(char*)malloc(sizeof(char));
38      supporto=(char*)malloc(sizeof(char));
39      do
40      {
41          rix=myself->ricevi(); //ricevi pacchetti gia' spezzati
42          if(rix!=NULL) //se ho ricevuto qualcosa
43          {
44              //supporto=cpyStr(completa);
45              //printf("\nlungh fuori: %d %d %d\n",lenStr(completa),lenStr
46                  (supporto),lenStr(rix));
47              //fflush(stdout);
48              completa=joinStr(completa,rix); //unisci i pacchetti
49          }
50      }
51      while(rix!=NULL);
52      printf("\nCompleta: %s\n",completa);
53      fflush(stdout);
54      sistemata=removeTAGs(completa);//funzione per pulire la stringa dai tag
55          html
56      printf("\nSistemata: %s\n",sistemata);
```

```
50     fflush(stdout);
51     return (0);
52 }
53
54 char* removeTAGs(char* s)
55 {
56     char* ret; //variabile che ritornera' la stringa pulita
57     int i,j,flag,lenStringa; //dichiarazione indici, flag di copia/non copia e  ↗
58     //lunghezza stringa di tutta la pagina
59     flag=0;
60     j=0;
61     lenStringa=lenStr(s);
62     ret=(char*)malloc(sizeof(char)*lenStringa);
63     for(i=0;i<lenStringa;i++)//per tutta la lunghezza della stringa completa
64     {
65         if((*s+i)=='<' && flag==0)//se trovi un < disattiva la copia
66         {
67             flag=1;
68         }
69         if(flag==0)//se il flag e' a 0 copia il carattere di completa nella  ↗
70             //sistemata
71         {
72             *(ret+j)=*(s+i);
73             j++;
74         }
75         if((*s+i)=='>' && flag==1) //se trovi un > riattiva la copia
76         {
77             flag=0;
78         }
79     }
80     *(ret+j)='\0';//aggiungi carattere di fine stringa
81     return(cpyStr(ret));
82 }
```