

```
1 #define DEBUG
2 #include "../Socket/ConnAll.h"
3 #define MAX_ETH 1500
4
5 void readFile(char*,Connection*);
6 void writeFile(char*,char*);
7 char* prelevaURL(char*);
8
9 int len_Str(char*s) {
10     int ret;
11     for (ret=0;*(s+ret);ret++);
12     return ret;
13 }
14
15 char* join_Str(char* s1,char*s2) {
16     char *ret,*tmp;
17     printf ("joinstr2:%s %s\n",s1,s2);
18     fflush(stdout);
19     ret= (char*)malloc (sizeof(char)*(len_Str(s1)+len_Str(s2)+1));
20     for ( tmp =ret ; *tmp = *s1; s1++, tmp++);
21     for ( ; *tmp = *s2; s2++, tmp++);
22     return (ret);
23 }
24
25 int main(int argc,char** argv)
26 {
27     ServerTCP *myself;//dichiarazione mio Server
28     int port,connId; //dichiarazione variabili che conterranno porta ed
        identificatore della connessione
29     char *rx,*url; //Dichiarazione variabile che conterra' il messaggio
        ricevuto e variabile che conterra' l'url richiesta dal client.
30     char* nomeFile; //dichiarazione variabile contenente la parte iniziale del
        path in cui la pagina richiesta si trova
31     char nomeFileB[]="./miofile.txt";//nome del file in cui vengono scritte le
        richieste ricevute.
32     Iterator* itera; //Iteratore che si occupa del movimento tra i vari nodi
        di connessione.
33     Connection* conny; //Singola connessione
34     Address* myAddr; //Address necessaria alla creazione del server e alla
        bind.
35     if(argc!=2)//caso dei parametri errati
36         errore("Errore:",argc);
37     port=str2int(*(argv+1));
38     myAddr=new Address(port,"0.0.0.0");
39     myself=new ServerTCP(myAddr,1);
40
41     while(1) //ciclo infinito
42     {
43         connId=myself->accetta();//ritorna NULL se nessuna connessione e' stata
        accettata
44         itera=myself->createIterator();
45         itera->moveLast(); //spostati sull'ultimo nodo
46         conny=(Connection*)itera->getCurrent(); //estrai lo
47         if(connId>0)//controlla che il connID ricevuto sia valido
48         {
49             conny=(Connection*)itera->getCurrent(); //passaggio inutile di cui mi
```

```

        sto chiedendo la ragione
50     rx=conny->ricevi(); //metti in rx cio' che stanno inviando
51     printf("\nServer ha Ricevuto: %s\n",rx); //stampalo
52     fflush(stdout);
53     url=prelevaURL(rx); //funzione che ottiene il nome della pagina
        richiesta
54     nomeFile="./myURL";
55     nomeFile=joinStr(nomeFile,url); //lega path di base con la pagina
        richiesta
56     printf("----->%s %s\n<-----",nomeFile,*(argv+1));
57     fflush(stdout);
58     readFile(nomeFile,conny); //leggi il file in archivio e invialo a
        pacchetti
59     writeFile(nomeFileB,rx); //scrivi su di un file che pagina e' stata
        richiesta
60     myself->close(conny); //chiusura connessione
61     }
62 }
63 return (0);
64 }
65
66 void readFile(char* nomeFile,Connection* conny)
67 {
68     char *ret,*completa;
69     char buffer[MAX_ETH]; //buffer che contiene il testo del pacchetto con misura
        1500 max
70     char compare[]="./myURL/"; //stringa contenente il path delle pagine del sito
        senza una pagina richiesta
71     FILE* fp; //puntatore ad un file
72     printf("---- %s ----\n",nomeFile);
73     fflush(stdout);
74     if((cmpStr(nomeFile,compare))==0) //controllo pagina richiesta, se e'
        nulla invio la home page.
75     {
76         nomeFile="./myURL/home.html";
77     }
78     if(fp=fopen(nomeFile,"r")) //apertura file in lettura
79     {
80         while(fgets(buffer,MAX_ETH,fp)!=NULL) //leggi file da inviare 1500
            caratteri alla volta
81         {
82             conny->invia(buffer); //invia cio' che hai letto
83         }
84     }
85     else
86     {
87         fp=fopen("./myURL/E404.html","r"); //apri la pagina di errore 404
88         while(fgets(buffer,MAX_ETH,fp)!=NULL) //leggi file da inviare
89         {
90             conny->invia(buffer); //e inviala
91         }
92     }
93     fclose(fp);
94 }
95
96 void writeFile(char* nomefile,char* rx)

```

```
97 {
98     FILE* fp;
99     if(fp=fopen(nomefile,"w")) //apertura file in write (sovrascrive)
100     {
101         fprintf(fp,"%s \n",rx); //stampa su file la richiesta
102         fclose(fp);
103     }
104     else
105     {
106         printf("Errore sul file");
107         fflush(stdout);
108     }
109 }
110
111 char* prelevaURL(char* rx)
112 {
113     char* url1,*url;
114     int i,j;
115     url1=(char*)malloc(sizeof(char)*1500); //alloca una stringa per contenere il
116     //nome della pagina richiesta
117     *url1='\0';
118     i=4;
119     j=0;
120     while(*(rx+i)!=' ' && j<1499) //cicla fino a che non trovi uno spazio
121     {
122         *(url1+j)=*(rx+i); //copia dal 4° carattere fino allo spazio GET' '<...>
123         j++;
124         i++;
125     }
126     *(url1+j]='\0'; //aggiungi carattere di fine stringa
127     url=cpyStr(url1);
128     return(url);
129 }
```