

Scrivete lo pseudocodice e le chiamate alla socket library necessarie per realizzare un semplice WWW client con i seguenti requisiti:

1. Dalla linea di comando, leggete (1) la URL che si vuol leggere sul server e (2) il server port number. Create il socket e collegatelo alla porta specificata (e.g., HTTP porta 80) [suggerimento: usate le chiamate getservbyname, gethostbyname, socket, connect].
2. Inviare la richiesta al WWW server usando il messaggio GET del protocollo HTTP. Specificate esplicitamente la richiesta.
3. Leggete i dati dalla connessione HTTP e salvateli su un file temporaneo creato nella vostra WWW cache (e.g., /tmp/yourloginname) sul computer locale
4. Lanciate il processo di un viewer per mostrare a video il file (suggerimento: usate [fork/exec]. Spiegate almeno due tecniche che potreste usare per decidere il viewer da lanciare a seconda del contenuto del file (suggerimento: usate il campo ContentType di HTTP)

---

## [PUNTO 1]

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
```

```
int main (int argc, char *argv[])
{
    int sockid; //descrittore del socket
    int porta; //numero porta da contattare sul server
    struct sockaddr_in serv_addr; //indirizzo IP del server
    int status;
    String richiesta, risposta, tipoFile, viewer; //saranno definiti nel codice

    //inizializzo la struct serv_addr coi parametri passati da linea di comando
    //per l'indirizzo IP dovrò eseguire una gethostbyname() sull'URL passato
    serv_addr.sin_addr = gethostbyname (argv[1]);
    //assegno la porta
    serv_addr.sin_port = argv[2];
    //specifico la famiglia
    //per brevità passo la stringa per assegnamento
    serv_addr.sin_family = "AF_INET";

    //creo il socket e controllo che l'operazione vada a buon fine
    if ((sockid = socket(AF_INET, SOCK_STREAM, 0)) != 0)
    {
        printf("Errore nella creazione del socket\n");
        exit(-1);
    }

    //eseguo la connect e verifico che l'operazione vada a buon fine
    if(status = connect (sockid, &serv_addr, sizeof(serv_addr)) != 0)
    {
        print("Errore nell'esecuzione della connect\n");
        exit(-1);
    }
}
```

## [PUNTO 2]

```
//scrivo in una stringa la richiesta corretta da inviare al WWW server
//per brevità gestisco la stringa con degli assegnamenti concatenati
richiesta = "GET" + argv[1] + "HTTP/1.1 \n Connection: close";
```

```
//invio la richiesta al server con una send
send(sockid, &richiesta, strlen(richiesta), 0);
```

## [PUNTO 3]

```
//memorizzo in una stringa la risposta inviata dal WWW server
recv(sockid, &risposta, strlen(risposta), 0);
```

```
//dal momento che la comunicazione con il server è terminata, chiudo il
socket
close(sockid);
```

```
//memorizzo la stringa risposta in un file temporaneo
//per brevità utilizzo una pseudo-funzione "creaFileTemporaneo" così
definita:
// void creaFileTemporaneo(String StringaDaCopiare, String PercorsoFile);
creaFileTemporaneo(&risposta, "/tmp/DoppioClic/rispostaTemp.temp");
```

## [PUNTO 4]

```
//per sapere con che viewer lanciare il file, dovrò capire che tipo di file ho
ricevuto
//dovrò quindi leggere dal campo "Content-type: " della risposta
```

```
//utilizzo una pseudo-funzione "estraiTipoFile" che con un algoritmo di
string- //matching scansiona una stringa fino a che individua la sequenza di
caratteri // "Content-type: ". A questo punto copia tutti i caratteri da qui al
carattere di fine
```

```
//riga "\n" in una stringa che mi ritorna come uscita
tipoFile = estraiTipoFile (&risposta);
```

```
//associo ad ogni tipoFile una stringa che contiene il nome del viewer
associato
```

```
switch (tipoFile) {
    case "text/html":
        viewer = "Mozilla 5.0";
        break;
    case "image/*":
        viewer = "Gimp";
        break;
    default:
        viewer = "Notepad";
        break;
}
```

```
//per eseguire il viewer utilizzo una pseudo-funzione esegui così definita:  
//void esegui(String NomeProgramma, String PercorsoFile);  
esegui(&viewer, "/tmp/DoppioClic/rispostaTemp.temp");
```

```
return 0;
```

```
}
```