UCS1511 - Networks Lab — Name: *Krithika Swaminathan*
AY: *2022-23* — Roll No.: *205001057*

Assignment 7 – HTTP Web Client Program to Download a Web Page using TCP

Date: **07/10/2022**

Aim:

To implement a HTTP web client program to download the webpage using C socket programming.

Algorithm:

Client:

1. Start
2. Read the name of the server as a command line argument.
3. Get the address of the server using the gethostbyname() function that returns the pointer to the network data structure for a given host.
4. Create a TCP socket using the socket() system call.
5. Connect to the remote server.
6. Send a request using a GET /path/filename HTTP/1.1\r\n request using the send() system call.
7. Receive the response using the recv() system call.
8. Parse the response to find out if the request succeeded and what format the file data is being sent in.
9. Receive the file data, if present, using the recv() system call and write the downloaded page into a file under a different name in a local folder.
10. Close the file.
11. Close the socket.
12. Stop

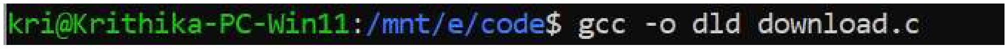
Code:

```
//HTTP Web Client Program for Downloading a Webpage using TCP socket

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
```

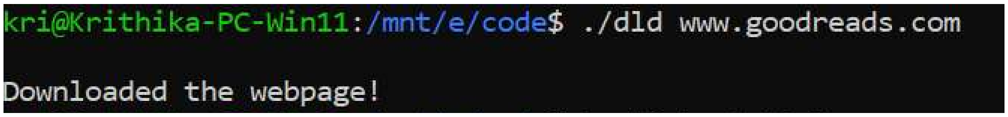
```
#include <netdb.h>

#define MAX 256

int main(int argc, char* argv[])
{
    int sock;
    char host[MAX];
    char message[MAX];
    char port[] = "80";
    struct addrinfo hints, *res;

    strcpy(host, argv[1]);
    snprintf(message, MAX, "GET / HTTP/1.1\nHost: %s\n\n", host);

    unsigned int i;
    char buf[1024];
    int bytes_read;
    int status;

    memset(&hints, 0, sizeof hints);
    hints.ai_family = AF_INET;
    hints.ai_socktype = SOCK_STREAM;
    status = getaddrinfo(host, port, &hints, &res);
    if (status != 0) {
        perror("getaddrinfo");
        return 1;
    }
    sock = socket(res->ai_family, res->ai_socktype, res->ai_protocol);
    if (sock == -1) {
        perror("socket");
        return 1;
    }
    status = connect(sock, res->ai_addr, res->ai_addrlen);
    if (status == -1) {
        perror("connect");
        return 1;
    }
    freeaddrinfo(res);
    send(sock, message, strlen(message), 0);

    FILE *out_file = fopen("pageOutput.html", "w");
    do {
        bytes_read = recv(sock, buf, 1024, 0);
        if (bytes_read == -1) {
            perror("recv");
        }
        else {
            fputs(buf, out_file);
        }
    } while (bytes_read > 0);

    fclose(out_file);
    printf("\nDownloaded the webpage!\n");
    close(sock);

    return 0;
}
```

Output:

Downloading the web page:

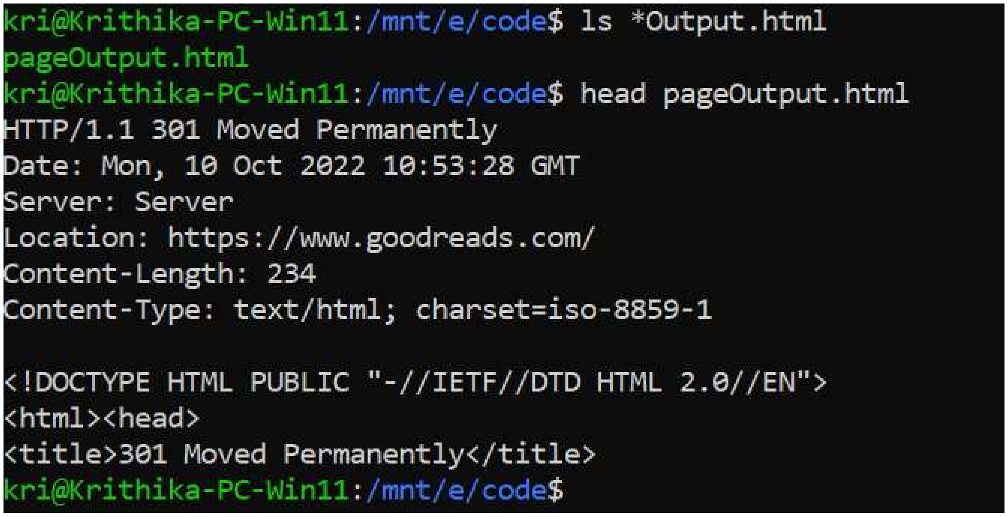


```
kri@Krithika-PC-Win11:/mnt/e/code$ gcc -o dld download.c
```



```
kri@Krithika-PC-Win11:/mnt/e/code$ ./dld www.goodreads.com

Downloaded the webpage!
```

Checking if the web page was downloaded and saved as a file:



```
kri@Krithika-PC-Win11:/mnt/e/code$ ls *Output.html
pageOutput.html
kri@Krithika-PC-Win11:/mnt/e/code$ head pageOutput.html
HTTP/1.1 301 Moved Permanently
Date: Mon, 10 Oct 2022 10:53:28 GMT
Server: Server
Location: https://www.goodreads.com/
Content-Length: 234
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
kri@Krithika-PC-Win11:/mnt/e/code$
```

On opening the file, the user is taken to *www.goodreads.com*.

Learning outcomes:

- The concept of using HTTP GET to send a request to a server to get the required data was understood and implemented.
- The concept of receiving data from a server or a web page using TCP sockets was understood and implemented.
- A web client program to download a web page as a file was implemented using TCP socket programming.