Secure Internet Application Development

CPS 499-63/592-65, Summer 2017

LAB 2: A Simple TCP Client application in C

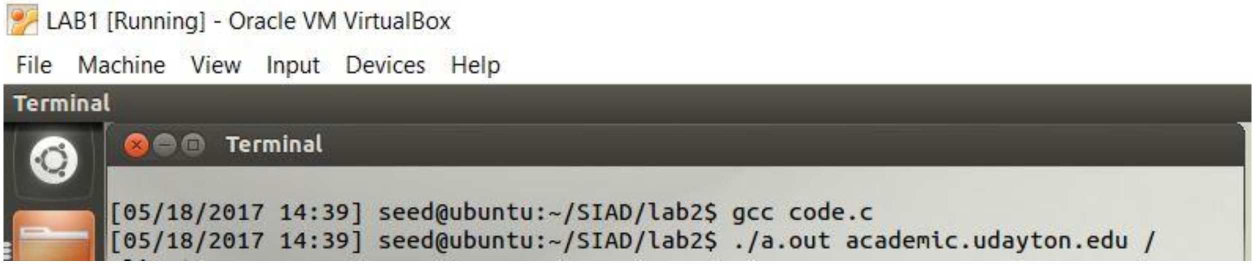
Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/socket.h>

#include<netdb.h>

#include<string.h>


int main(int argc, char *argv[])

{

        char *servername;

        char *path;

        int port;

        printf("Client program \n");

        if(argc!=3)

        {

                printf("Usage : %s <server> <port> \n", argv[0]);

                exit(0);

        }


        servername = argv[1];

        path=argv[2];
        port = atoi(argv[2]);

        printf("Servername= %s , port=%d \n",servername,port);


        int sockfd;

        sockfd = socket(AF_INET, SOCK_STREAM, 0);

        if(sockfd<0)

        {

                perror("ERROR opening socket");

        }


        struct hostent *server_he; //an host address entry

        if ((server_he = gethostbyname(servername)) == NULL)

        {

            perror( " error in gethostbyname");

            return 2;

        }

        struct sockaddr_in serveraddr; // store the server's address


        //prepare to copy: clear sockaddr_in structure

        /* (In network programming, we often need to initialize a field, copy the contents of one field to
        another */

        bzero((char *) &serveraddr, sizeof(serveraddr));


        //set the family to AF_INET (IPv4)

        serveraddr.sin_family = AF_INET;


        /* copy the server’ address from gethostbyname(..), //return in struct hostent (stored in
*server_he) */
```

```
        bcopy((char *)server_he->h_addr,
        (char *)&serveraddr.sin_addr.s_addr,
        server_he->h_length);
        serveraddr.sin_port = htons(port);
        if(connect(sockfd, (struct sockaddr *) &serveraddr, sizeof(serveraddr)) < 0)
        {
                perror("Cannot connect to the server");
                exit(0);
        }
        else
                printf("Connected to the server");
        char request[255];
        sprintf(request,"GET /%s HTTP/1.0\r\nHost:%s\r\n\r\n",path,servername);
        char *msg = "GET / HTTP/1.0\r\n\r\n";
        int bytes_sent;
        printf("message:%s",request);
        bytes_sent = send(sockfd, msg, strlen(msg) , 0);
        char buffer[1024];
        bzero(buffer,1024);
        int byte_received;
        bzero(buffer , 1024);
        byte_received = recv(sockfd, buffer , 1024, 0);
        if(byte_received < 0)
          error ("ERROR reading from socket");
        printf("Message received : %s", buffer);
        close(sockfd);
        return 0;
}
```

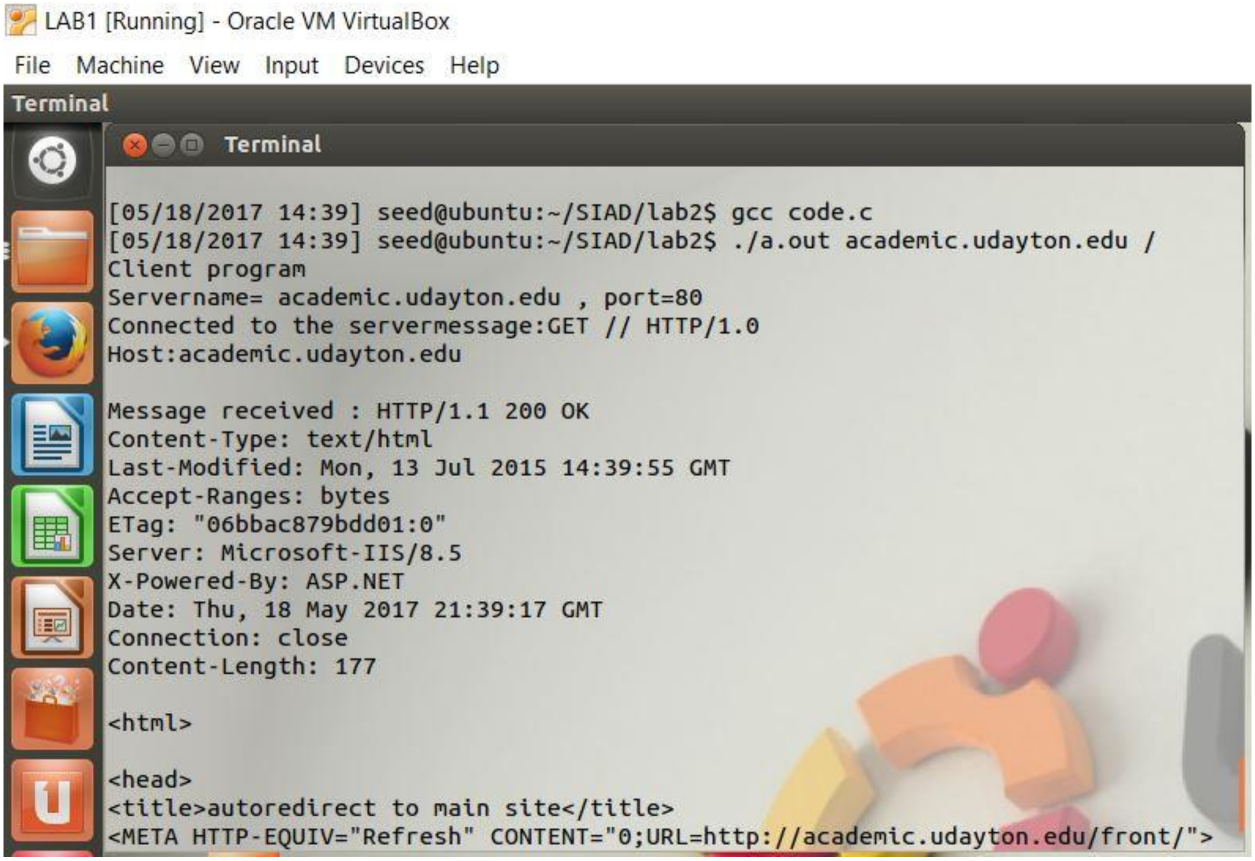
Screenshots:

Input URL 1:


LAB1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

```
[05/18/2017 14:39] seed@ubuntu:~/SIAD/lab2$ gcc code.c
[05/18/2017 14:39] seed@ubuntu:~/SIAD/lab2$ ./a.out academic.udayton.edu /
```

Output 1:


LAB1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

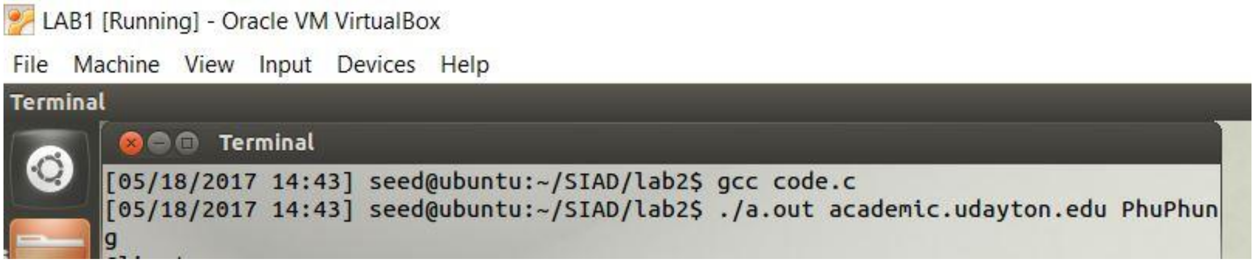
```
[05/18/2017 14:39] seed@ubuntu:~/SIAD/lab2$ gcc code.c
[05/18/2017 14:39] seed@ubuntu:~/SIAD/lab2$ ./a.out academic.udayton.edu /
Client program
Servername= academic.udayton.edu , port=80
Connected to the servermessage:GET // HTTP/1.0
Host:academic.udayton.edu

Message received : HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Mon, 13 Jul 2015 14:39:55 GMT
Accept-Ranges: bytes
ETag: "06bbac879bdd01:0"
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
Date: Thu, 18 May 2017 21:39:17 GMT
Connection: close
Content-Length: 177

<html>

<head>
<title>autoredirect to main site</title>
<META HTTP-EQUIV="Refresh" CONTENT="0;URL=http://academic.udayton.edu/front/">
```

Explanation:

In this example, an URL is the input : the call will be sent to the server and a HTTP Response will be displayed.

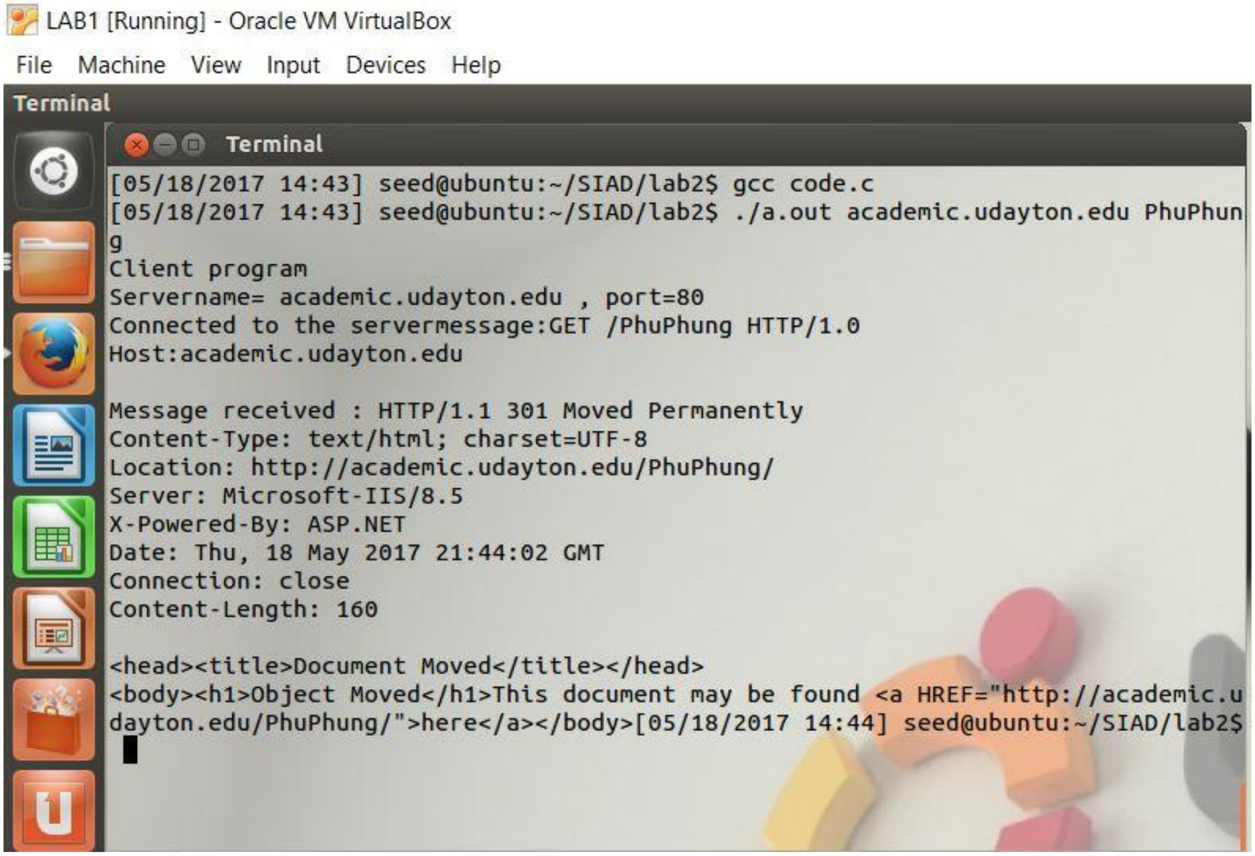
HTTP/1.1 200 OK – Successful HTTP Response

Input URL 2:


LAB1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

```
[05/18/2017 14:43] seed@ubuntu:~/SIAD/lab2$ gcc code.c
[05/18/2017 14:43] seed@ubuntu:~/SIAD/lab2$ ./a.out academic.udayton.edu PhuPhun
g
```

Output 2:


LAB1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

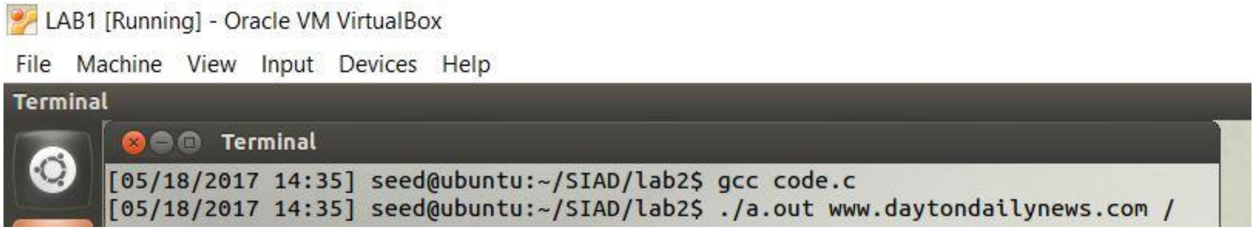
```
[05/18/2017 14:43] seed@ubuntu:~/SIAD/lab2$ gcc code.c
[05/18/2017 14:43] seed@ubuntu:~/SIAD/lab2$ ./a.out academic.udayton.edu PhuPhun
g
Client program
Servername= academic.udayton.edu , port=80
Connected to the servermessage:GET /PhuPhung HTTP/1.0
Host:academic.udayton.edu

Message received : HTTP/1.1 301 Moved Permanently
Content-Type: text/html; charset=UTF-8
Location: http://academic.udayton.edu/PhuPhung/
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
Date: Thu, 18 May 2017 21:44:02 GMT
Connection: close
Content-Length: 160

<head><title>Document Moved</title></head>
<body><h1>Object Moved</h1>This document may be found <a HREF="http://academic.u
dayton.edu/PhuPhung/">here</a></body>[05/18/2017 14:44] seed@ubuntu:~/SIAD/lab2$
```

Explanation:

In this example, an URL is the input : the call will be sent to the server and a HTTP Response will be displayed.

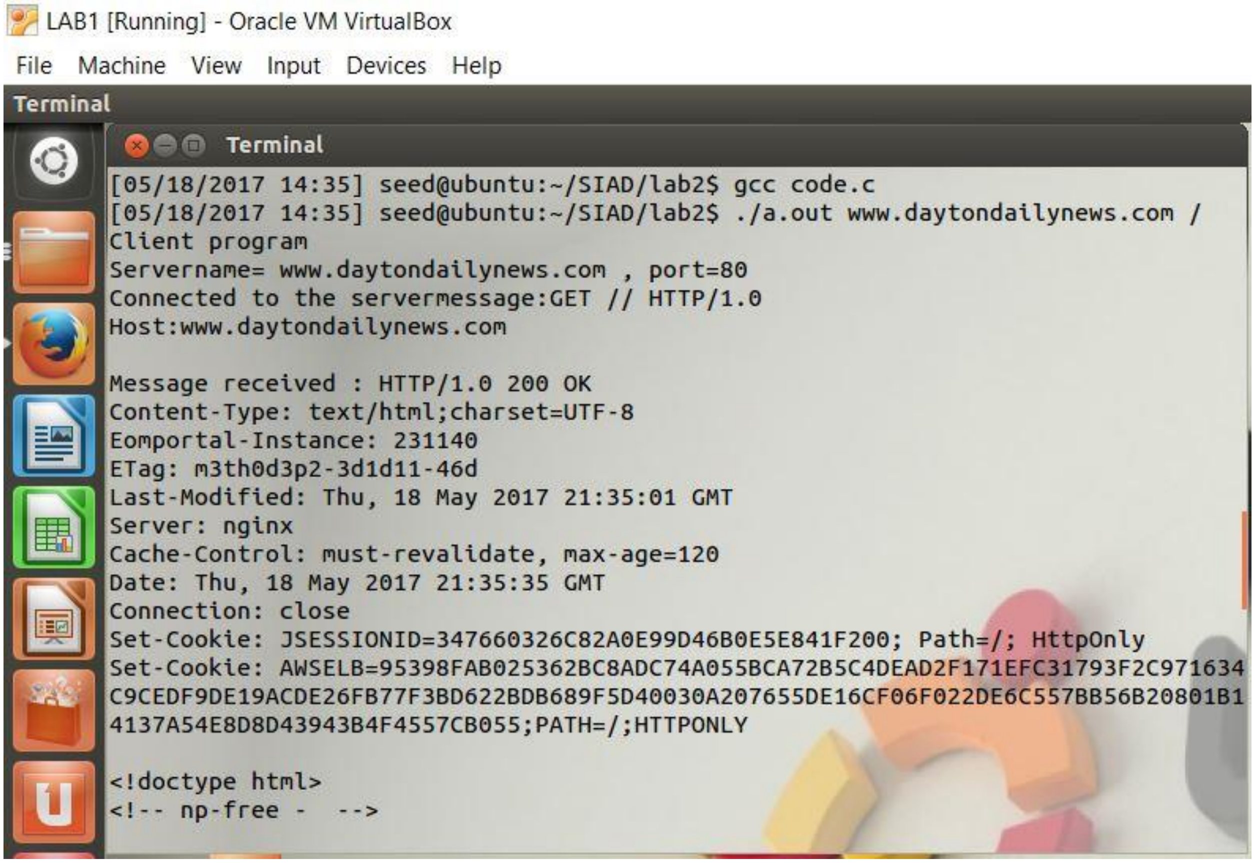
HTTP/1.1 301 Moved Permanently – HTTP Response which indicates that site is moved to another location permanently

Input URL 3:


LAB1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

```
[05/18/2017 14:35] seed@ubuntu:~/SIAD/lab2$ gcc code.c
[05/18/2017 14:35] seed@ubuntu:~/SIAD/lab2$ ./a.out www.daytondailynews.com /
```

Output 3:


LAB1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

```
[05/18/2017 14:35] seed@ubuntu:~/SIAD/lab2$ gcc code.c
[05/18/2017 14:35] seed@ubuntu:~/SIAD/lab2$ ./a.out www.daytondailynews.com /
Client program
Servername= www.daytondailynews.com , port=80
Connected to the servermessage:GET // HTTP/1.0
Host:www.daytondailynews.com

Message received : HTTP/1.0 200 OK
Content-Type: text/html;charset=UTF-8
Eomportal-Instance: 231140
ETag: m3th0d3p2-3d1d11-46d
Last-Modified: Thu, 18 May 2017 21:35:01 GMT
Server: nginx
Cache-Control: must-revalidate, max-age=120
Date: Thu, 18 May 2017 21:35:35 GMT
Connection: close
Set-Cookie: JSESSIONID=347660326C82A0E99D46B0E5E841F200; Path=/; HttpOnly
Set-Cookie: AWSELB=95398FAB025362BC8ADC74A055BCA72B5C4DEAD2F171EFC31793F2C971634
C9CEDF9DE19ACDE26FB77F3BD622BDB689F5D40030A207655DE16CF06F022DE6C557BB56B20801B1
4137A54E8D8D43943B4F4557CB055;PATH=/;HTTPONLY

<!doctype html>
<!-- np-free -  -->
```

Explanation:

In this example, an URL is the input : the call will be sent to the server and a HTTP Response will be displayed.

HTTP/1.1 200 OK – Successful HTTP Response