

מטלת סיום תקשורת

מגיש: דולב דובלון ת.ז. 207867342

הערות ותשובות לשאלות

מצורף קובץ makefile שמקמפל את כל הקבצים בנוחות כדי שיהיה נוח להריץ את הקוד

בעיה 1

Send & receive udp

1. לקמפל ולהריץ, להסביר מה קרה ולהראות שאני מבין את התוכנות

בקובץ receive udp אנו פותחים שרת udp באמצעות הפקודה

```
;socket_fd = socket(AF_INET, SOCK_DGRAM, 0)
```

```
32  
33 socket_fd = socket(AF_INET, SOCK_DGRAM, 0);
```

כאשר הדגל SOCK_DGRAM מייצג פתיחת סוקט למטרת תיקשורת ב UDP

```
int socket(int __domain, int __type, int __protocol)  
Create a new socket of type TYPE in domain DOMAIN, using  
protocol PROTOCOL. If PROTOCOL is zero, one is chosen automatically.  
Returns a file descriptor for the new socket, or -1 for errors.
```

כאשר הפרמטר האחרון שהוא 0, מציע לנו בחירה אוטומטית של הפרוטוקול כמו שרשום.

הערך המוחזר הוא מספר ה file descriptor עבור הסוקט החדש, או 1- עבור שגיאה.

INADDR_ANY דגל שמייצג לכל הכתובות כאשר אין דרישה ספציפית ל IP

INADDR_ANY is used when you don't need to bind a socket to a specific IP. When you use this value as the address when calling bind(), the socket accepts connections to all the IPs of the machine

ולפורט הזה

(u_short)0x3333

```
37 s_in.sin_family = (short)AF_INET;  
38 s_in.sin_addr.s_addr = htonl(INADDR_ANY); /* WILDCARD */  
39 s_in.sin_port = htons((u_short)0x3333);  
40
```

```
// printsin( &s_in, "RCV_UDP", "Local socket is:");
fflush(stdout);

bind(socket_fd, (struct sockaddr *)&s_in, sizeof(s_in));

for (;;)
{
    fsize = sizeof(from);
    cc = recvfrom(socket_fd, &msg, sizeof(msg), 0, (struct sockaddr *)&from, &fsize);
    // printsin( &from, "recv_udp: ", "Packet from:");
    printf("Got data ::%c%ld%c\n", msg.head, (long)ntohl(msg.body), msg.tail);
    fflush(stdout);
}
```

לאחר קונפיגורציה של משתני הסוקט , נבצע ניקוי של הטרמינל עם הפונקציה fflush

נבצע bind על הסוקט לסטראקט s_in

וניכנס ללולאה אין סופית שבה נחכה לפאקטות שמתקבלות , באמצעות הפונקציה recvfrom , שבעזרתה נדפיס את הקלט שמתקבל לתוך משתנה msg

// סוף הסבר על receive udp

הסבר על send_udp

```
socket_fd = socket(AF_INET, SOCK_DGRAM, 0);|
bzero((char *)&dest, sizeof(dest)); /* They say you must do this */
hostptr = gethostbyname(argv[1]);
dest.sin_family = (short)AF_INET;
bcopy(hostptr->h_addr, (char *)&dest.sin_addr, hostptr->h_length);
dest.sin_port = htons((u_short)0x3333);

msgbuf.head = '<';
msgbuf.body = htonl(getpid()); /* IMPORTANT! */
msgbuf.tail = '>';

sendto(socket_fd, &msgbuf, sizeof(msgbuf), 0, (struct sockaddr *)&dest,
| | | sizeof(dest));
```

מאתחלים קונפיגורציות של סוקט של UDP , מבצעים השפה ל פורט המתאים ולכתובת הIP מתאימה

ולבסוף שולחים את ההודעה לפי הקונפיגורציה המתאימה באמצעות פקודת sendto

מעבירים לתוכנה פרמטר שהוא שם המחשב (אצלי זה היה ubuntu) בשורה השלישית הוא מתקבל בתור פרמטר ל gethostbyname שלוקחת את השם ומחליטה מי ה host , getpid() פונקציה מחזירה את מספר ה תהליך שרץ , המשמעות היא בשבילנו שנריץ כמה פעמים , שנדע איזה מספר התהליך שלנו.

ואותו אנו שולחים , כך נדע אם ה תהליך עדיין עובד כשנבדוק איזה תהליכים רצים במחשב.

2. לממש פונקציה חדשה printsin()

```
13 void printsin(struct sockaddr_in *s, char *str1, char *str2) {  
14     printf("%s\n", str1);  
15     printf("%s ", str2);  
16     printf("ip= %s ", inet_ntoa(s->sin_addr));  
17     printf("port= %d", ntohs(s->sin_port));  
18     printf("\n");  
19 }  
20
```

דפי man שנעזרתי בהם

<https://man7.org/linux/man-pages/man2/socket.2.html>

<https://man7.org/linux/man-pages/man7/socket.7.html>

<https://linux.die.net/man/2/recvfrom>

הרצה

Make clean

Make

./recv_udp

And in a new terminal

./send_udp localhost

או

./send_udp <hostname>

יצרתי את שלושת הקבצים, לפי ההוראות

קובץ ראשון מקור

Source.c

מקבל 2 פרמטרים , הראשון hostname השני מספר port למשל 1000

ושולח הודעות ל gateway

Gateway.c

מקבל את ההודעות ומטיל הסתברות של חצי לשליחת ההודעה הלאה ל sink.c ושולח אם ההסתברות יצאה 1

מקבל בטרמינל 3 פרמטרים , הראשון hostname השני מספר הפורט של source.c למשל 1000 והפרמטר השלישי מספר ה Port של sink.c למשל 1001

Sink.c

מדפיס את הודעות שהצליחו להגיע אליו.

מקבל 2 פרמטרים בטרמינל

הראשון hostname השני מספר port למשל 1001

הרצה

Make clean

Make

```
./sink localhost 1001
```

And in a new terminal

```
./gateway localhost 1000 1001
```

And in anew terminal

```
./source localhost 1000
```

You can switch the word “localhost” with the name of your hostname

```
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish# ./sink localhost 1001
Recv udp:
Local socket is: ip= 0.0.0.0 port= 59651
^rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<0>
rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<1>
rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<2>
rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<3>
rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<4>
rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<5>
rcv udp:
Packet from: ip= 127.0.0.1 port= 59395
Got data ::<11>
0.
```

ניתן לראות שלא כל ההודעות הגיעו ל sink בטרמינל בצד שמאל לפי הקפיצה בין 5 ל 11

- Use the files net_server.c and net_client.c and compile them review the programs to be sure you understand them

הקובץ של השרת TCP מייצר שרת ב port 9999

<https://man7.org/linux/man-pages/man2/listen.2.html>

מאזין ל חיבור אחד

```
28
29 listen(sock, 1);
30
```

לאחר מכן מתבצע שלב הקבלה של החיבור באמצעות accept()

<https://man7.org/linux/man-pages/man2/accept4.2.html>

SYNOPSIS [top](#)

```
#include <sys/socket.h>

int accept(int sockfd, struct sockaddr *restrict addr,
           socklen_t *restrict addrLen);

#define _GNU_SOURCE /* See feature_test_macros(7) */
#include <sys/socket.h>

int accept4(int sockfd, struct sockaddr *restrict addr,
            socklen_t *restrict addrLen, int flags);
```

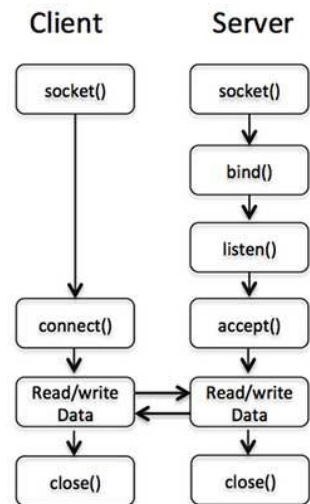
ואז מתבצעת שליחה

של קאונטר מסויים לקליינט באמצעות הפונקציה write על ה connect socket שלנו ,
הלולאה מתבצעת SIM_LENGTH פעמים אך המשתנה לא מוגדר.

```
32
33 connect_sock = accept(sock, (struct sockaddr *)&serv_name, &len);
34
35 for (count = 1; count <= SIM_LENGTH; count++)
36 { write(connect_sock, &count, 4);
37   printf("Server has written %d to socket.\n", count);
38 }
39
40 close(connect_sock);
41 close(sock);
```

בסיום סוגרים את הסוקטים.

שלבי הפעולה של שרת tcp



בקובץ net_client

```

11 #define SIM_LENGTH 10
12 #define IP_ADDRESS "???"
13 #define PORT 1337
14

```

אנו מבצעים חיבור של סוקט TCP ל פורט 1337 אך ללא IP מוגדר, המשתנה LENGTH_SIM מוגדר פה וגם הוא מוגדר לערכו ל 10

```

32
33 bzero(&cli_name, sizeof(cli_name));
34 cli_name.sin_family = AF_INET;
35 cli_name.sin_addr.s_addr = inet_addr(IP_ADDRESS);
36 cli_name.sin_port = htons(PORT);
37

```

הפרמטרים הנל מועברים לפונקציות להשמה של הפורט ל סטראקט sockaddr_in

```

37 if (connect(sock, (struct sockaddr *)&cli_name, sizeof(cli_name)) < 0)
38 {
39     perror("Error establishing communications");
40     close(sock);
41     exit(1);
42 }
43
44 for (count = 1; count <= SIM_LENGTH; count++)
45 {
46     read(sock, &value, 4);
47     printf("Client has received %d from socket.\n", value);
48 }
49

```

מבצעים connect לפי שלבי tcp, ואז מבצעים קריאה read עשר פעמים לסוקט, לקרוא ערך בגודל 4 בייטים.

```

root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish# make task3
gcc -o net_client net_client.c -Wall
net_client.c: In function 'main':
net_client.c:34:30: warning: implicit declaration of function 'inet_addr' [-Wimplicit-function-declaration]
   34 |     cli_name.sin_addr.s_addr = inet_addr(IP_ADDRESS);
      |                               ~~~~~
gcc -o net_server net_server.c -Wall
net_server.c: In function 'main':
net_server.c:33:62: warning: passing argument 3 of 'accept' from incompatible pointer type [-Wincompatible-pointer-types]
   33 |     connect_sock = accept(sock, (struct sockaddr *)&serv_name, &len);
      |                                                              ~~~~
      |                                                              |
      |                                                              size_t * {aka long unsigned int *}
In file included from net_server.c:7:
/usr/include/x86_64-linux-gnu/sys/socket.h:233:28: note: expected 'socklen_t * restrict' {aka 'unsigned int * restrict'} but argument is of type 'size_t *' {aka 'long unsigned int *'}
   233 |     socklen_t * _restrict __addr_len);
      |     ~~~~~
net_server.c:35:28: error: 'SIM_LENGTH' undeclared (first use in this function)
   35 |     for (count = 1; count <= SIM_LENGTH; count++)
      |                             ~~~~~
net_server.c:35:28: note: each undeclared identifier is reported only once for each function it appears in
make: *** [makefile:13: net_server] Error 1

```

שגיאות קומפילציה, SIM_LENGTH לא מוגדר בקובץ net_server, בנוסף יש שגיאות של טיפוסים לא מתאימים. שגיאות לוגיות / בזמן ריצה :

מספרי PORT שונים בין הקובץ net_client לבין הקובץ net_server ולכן השליחה והקבלה לא תתבצע כהלכה
אם מספר ה LENGTH_SIM יהיה שונה בין קובץ לקובץ כלומר השליחה והקבלה יהיה שונה, לא כל המידע יתקבל
או ישלח ואחד הקצוות ימתין להודעה או ישלח הודעה בעוד שהסוקט האחר יחכה להודעה הוא ישלח הודעה שאף אחד לא מחכה לקבל אותה

מוגדר ADDRESS_IP בקובץ CLIENT בעזרת #define מה שיוצר את האזהרה הבאה

```

net_client.c: In function 'main':
net_client.c:34:30: warning: implicit declaration of function 'inet_addr' [-Wimplicit-function-declaration]
   34 |     cli_name.sin_addr.s_addr = inet_addr(IP_ADDRESS);
      |                               ~~~~~

```

תיקון התקלות:

בחירת PORT משותף וזהה, נבחר את 9999

השמה של הכתובת IP ל 127.0.0.1

LENGTH_SIM

לאחר התיקונים ניתן לקמפל ולהריץ וזה יראה כך

```
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish# make
gcc -o recv_udp recv_udp.c -Wall
gcc -o send_udp send_udp.c -Wall
Server has written 1 to socket.
Server has written 2 to socket.
Server has written 3 to socket.
Server has written 4 to socket.
Server has written 5 to socket.
Server has written 6 to socket.
Server has written 7 to socket.
Server has written 8 to socket.
Server has written 9 to socket.
Server has written 10 to socket.
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish#

root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish# ./net_cli
Client has received 2 from socket.
Client has received 3 from socket.
Client has received 4 from socket.
Client has received 5 from socket.
Client has received 6 from socket.
Client has received 7 from socket.
Client has received 8 from socket.
Client has received 9 from socket.
Client has received 10 from socket.
Exiting now.
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish#
```

הרצה

Make clean

Make

./net_server

./net_client localhost

בצד ימין טרמינל של לקוח , בצד שמאל טרמינל של שרת

הרצת nslookup/

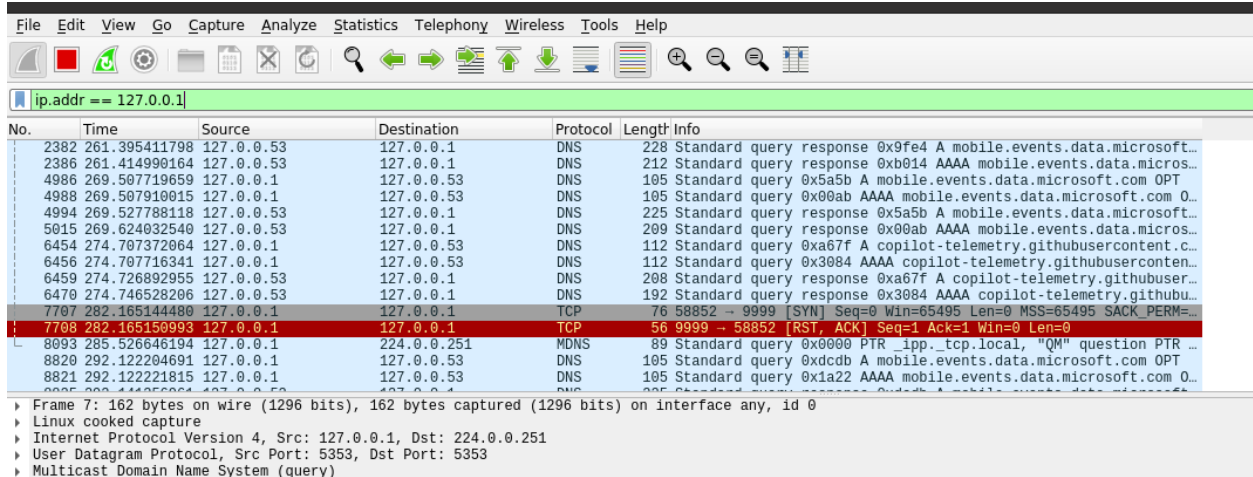
```
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish# nslookup localhost
Server:          172.27.192.1
Address:         172.27.192.1#53

Non-authoritative answer:
Name:   localhost
Address: 127.0.0.1
Name:   localhost
Address: ::1
```


כאשר מריצים את הקליינט כאשר השרת לא רץ נקבל שגיאת התחברות

```
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish# ./net_client
Client is alive and establishing socket connection.
Error establishing communications: Connection refused
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/final_finish#
```

שאלתי את חיים בעקבות קושי לעבוד עם tcpdump כלי שלא למדנו עליו , אם יש אפשרות לעבוד עם wireshark ונמסר שכן



The screenshot shows the Wireshark interface with a packet capture on the interface 'ip.addr == 127.0.0.1'. The packet list shows several DNS queries and responses, and a TCP RST, ACK packet. The packet details pane shows the selected packet (No. 7708) as a Multicast Domain Name System (query) packet from 127.0.0.1 to 224.0.0.251.

No.	Time	Source	Destination	Protocol	Length	Info
2382	261.395411798	127.0.0.53	127.0.0.1	DNS	228	Standard query response 0x9fe4 A mobile.events.data.microsoft...
2386	261.414990164	127.0.0.53	127.0.0.1	DNS	212	Standard query response 0xb014 AAAA mobile.events.data.micros...
4986	269.507719659	127.0.0.1	127.0.0.53	DNS	105	Standard query 0x5a5b A mobile.events.data.microsoft.com OPT
4988	269.507910015	127.0.0.1	127.0.0.53	DNS	105	Standard query 0x00ab AAAA mobile.events.data.microsoft.com 0...
4994	269.527788118	127.0.0.53	127.0.0.1	DNS	225	Standard query response 0x5a5b A mobile.events.data.microsoft...
5015	269.624032540	127.0.0.53	127.0.0.1	DNS	209	Standard query response 0x00ab AAAA mobile.events.data.micros...
6454	274.707372064	127.0.0.1	127.0.0.53	DNS	112	Standard query 0xa67f A copilot-telemetry.githubusercontent.c...
6456	274.707716341	127.0.0.1	127.0.0.53	DNS	112	Standard query 0x3084 AAAA copilot-telemetry.githubusercontent...
6459	274.726892955	127.0.0.53	127.0.0.1	DNS	208	Standard query response 0xa67f A copilot-telemetry.githubuser...
6470	274.746528206	127.0.0.53	127.0.0.1	DNS	192	Standard query response 0x3084 AAAA copilot-telemetry.githubu...
7707	282.165144480	127.0.0.1	127.0.0.1	TCP	76	58852 -> 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=...
7708	282.165150993	127.0.0.1	127.0.0.1	TCP	56	9999 -> 58852 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8093	285.526646194	127.0.0.1	224.0.0.251	MDNS	89	Standard query 0x0000 PTR _ipp_tcp.local, "QM" question PTR ...
8820	292.122204691	127.0.0.1	127.0.0.53	DNS	105	Standard query 0xdcdb A mobile.events.data.microsoft.com OPT
8821	292.122221815	127.0.0.1	127.0.0.53	DNS	105	Standard query 0x1a22 AAAA mobile.events.data.microsoft.com 0...

Frame 7: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 224.0.0.251
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (query)

ניתן לראות בתמונה הנ"ל מה ההודעה שמקבלים בסוקט של הקליינט כאשר מריצים רק את קליינט כאשר הסרבר לא מאזין.

בדיקת hostname בעזרת nslookup.c

הרצה כמה פעמים

```
gcc -o gateway gateway.c -Wall
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish# ./nslookup google.com
Address for google.com is 216.58.205.206
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish#
```

```
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish# ./nslookup yahoo.com
Address for yahoo.com is 98.137.11.164
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish#
```

```
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish# ./nslookup ynet.com
Address for ynet.com is 123.60.211.213
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish#
```

השינויים שביצעתי בקוד של net_client הם שהוספתי את הקוד מ nslookup כך שיתנו לי את הכתובת ip שצריך כאשר היא משתמשת ב getaddrinfo ומעבירה לו את שם ה hostname שמקבלים בתור פרמטר מהטרמינל והכנסתי את הכתובת שקיבלתי במקום המתאים במקום המשתנה IP_ADDRESS שהיה שם בתחלה. בתמונות למטה ניתן לראות את השינויים.

```
42 hostname = argv[1]; // hostname is the first argument
43 printf("Hostname is %s\n", hostname);
44
45 if (0 != getaddrinfo(hostname, NULL, NULL, &res)) // getaddrinfo is a function that returns a struct addrinfo* that
46 // contains a linked list of struct addrinfo (from nslookup.c)
47 {
48     fprintf(stderr, "Error in resolving hostname %s\n", hostname);
49     exit(1);
50 }
51
52
53
54
55
56
57
58
59
60 saddr = (struct sockaddr_in *)res->ai_addr; // get the address of the server socket address structure (from nslookup.c)
61 hostaddr = inet_ntoa(saddr->sin_addr); // get the IP address of the server (from nslookup.c)
62
63
64
65 cli_name.sin_addr.s_addr = inet_addr(hostaddr); // set the IP address of the client (from nslookup.c)
```

בעיה 3 חלק שני

Wget mimic

בניית לקוח שמבקש אתרי אינטרנט

ניעזר בקבצים הבאים

wget.c

Parseurl.c

Parseurl.h

Testparseurl.c

<https://weinman.cs.grinnell.edu/courses/CSC213/2008F/examples/web/parseurl.h>

<https://weinman.cs.grinnell.edu/courses/CSC213/2008F/examples/web/parseurl.c>

<https://weinman.cs.grinnell.edu/courses/CSC213/2008F/examples/web/testparseurl.c>

בקבצים

Parseurl.c

Parseurl.h

Testparseurl.c

אני פשוט כתבתי קוד שמפרק לי את כל החלקים של האתר כך שאני יוכל לבקש hostname בנפרד כמו שביקשו ואז ה Path בנפרד

לפי הפורמט של הבקשות GET

ב testparseurl אני מעביר לו בפרמטר שהוא באמת מפרד את הכתובת של האתר

```
30
31     if (0 != parse_url(url, argv[1]))
32     {
33         perror("Error: Could not parse URL\n");
34         exit(1);
35     }
36
37     printf("URL: %s\n", url->url);
38     printf("Protocol: %s\n", url->protocol);
39     printf("Hostname: %s\n", url->hostname);
40     printf("Port: %d\n", url->port);
41     printf("Path: %s\n", url->path);
42
```

```

root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish# ./testparse http://info.cern.ch/
URL: http://info.cern.ch/
Protocol: http
Hostname: info.cern.ch
Port: 80
Path:
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish#

```

ואז פשוט ב wget.c אני משלב את כל הקוד ביחד

גם מ parse.h

וגם מ nslookup.c

ומתחברים בעזרת הסוקטים המתאימים והגדרות המתאימות והקונפיגורציה המתאימה ל ip המתאים של השרת ופותחים חיבור tcp אליו

מבצעים בקשת get לפי הפורמט

```

107 // make a string to send to the server
108 char *send_string = malloc(sizeof(char) * (strlen(hostname) + SIM_LENGTH)); // allocate memory for the string
109 if (strlen(url->path) > 0) // send the request , check if we have a path or not
110 {
111     sprintf(send_string, "GET /%s HTTP/1.1\r\nHost: %s\r\nConnection: close\r\n\r\n", url->path, hostname); // if we have a path
112 }
113 else
114 {
115     sprintf(send_string, "GET / HTTP/1.1\r\nHost: %s\r\nConnection: close\r\n\r\n", hostname); // if we don't have a path, send t
116 }
117

```

אם קיים path נוסף מגישים גם אותו , אחרת מגישים בקשת get רגילה , כמובן כל הבקשות יהיו ל port 80 כי אנחנו עובדים עם פרוטוקול http שמוגדר להיות על הפורט הזה.

ואז שולחים את הבקשה לשרת

```

110 }
117
118 // const char *request = "GET / HTTP/1.0\r\nHost: {hostaddr}\r\nConnection: close\r\n\r\n";
119 int n;
120 printf("Sending request to server.\n%s\n", send_string); // print the request
121 n = write(sock, send_string, strlen(send_string)); // send the request to the server
122 if (n < 0) // if there is an error , print error
123 {
124     perror("Error writing to socket");
125     close(sock);
126     exit(1);
127 }

```

לאחר מכן עוברים למצב קבלה, שבו נחכה לתוכן האתר שיחזור אלינו, נבצע קריאה לתוך buffer ופשוט נטען אליו את התוכן ונדפיס שורה שורה

```
128 char buffer[4096]; // buffer to store the response from the server
129
130 while (1) //
131 {
132     bzero(buffer, 4096);
133
134     n = recv(sock, buffer, 4095, 0);
135     if (n < 0)
136     {
137         printf("ERROR reading from socket\n");
138         break;
139     }
140     if (n == 0)
141     {
142         printf("Server closed the connection\n");
143         break;
144     }
145     printf("%d\n", n);
146     printf("%s\n", buffer);
147 }
148 printf("Exiting now.\n");
149 close(sock);
150 free_url_data(url);
151 free(url);
152 exit(0);
153
154 // end net_client
155
```

לבסוף כאשר recv יהיה 0 אז משמעות הדבר היא שהחיבור נסגר וסיימנו לקבל את תוכן האתר. נסגור את כל החיבורים ונשחרר את כל הסיקטים ומהשתנים ונסגור את התוכנית.

דוגמת ריצה על האתר <http://info.cern.ch>

```
aaa@ubuntu:~/Desktop/why_so_hard$ ./wget http://info.cern.ch/
URL: http://info.cern.ch/
Protocol: http
Hostname: info.cern.ch
Port: 80
Path:
Address for info.cern.ch is 188.184.21.108
Client is alive and establishing socket connection.
Sending request to server.
GET / HTTP/1.1
Host: info.cern.ch
Connection: close

878
HTTP/1.1 200 OK
Date: Thu, 04 Aug 2022 14:36:21 GMT
Server: Apache
Last-Modified: Wed, 05 Feb 2014 16:00:31 GMT
ETag: "286-4f1aadb3105c0"
Accept-Ranges: bytes
Content-Length: 646
Connection: close
Content-Type: text/html

<html><head></head><body><header>
<title>http://info.cern.ch</title>
</header>

<h1>http://info.cern.ch - home of the first website</h1>
<p>From here you can:</p>
<ul>
<li><a href="http://info.cern.ch/hypertext/WWW/TheProject.html">Browse the first website</a></li>
<li><a href="http://line-mode.cern.ch/www/hypertext/WWW/TheProject.html">Browse the first website using the line-mode browser simulator</a></li>
<li><a href="http://home.web.cern.ch/topics/birth-web">Learn about the birth of the web</a></li>
<li><a href="http://home.web.cern.ch/about">Learn about CERN, the physics laboratory where the web was born</a></li>
</ul>
</body></html>

Server closed the connection
Exiting now.
aaa@ubuntu:~/Desktop/why_so_hard$
```

וכאשר נגיש בקשה לכתובת אבל עם path שלא קיים <http://info.cern.ch/does-not-exist>


נקבל שגיאה 400 או 404

```
Path:
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish# ./web_client https://info.cern.ch/does-not-exist
URL: https://info.cern.ch/does-not-exist
Protocol: https
Hostname: info.cern.ch
Port: 80
Path: does-not-exist
Address for info.cern.ch is 188.184.21.108
Client is alive and establishing socket connection.
Sending request to server.
GET /does-not-exist HTTP/1.1
Host: info.cern.ch
Connection: close

392
HTTP/1.1 400 Bad Request
Date: Wed, 03 Aug 2022 21:59:22 GMT
Server: Apache
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>

Server closed the connection
Exiting now.
root@DESKTOP-V5TJKTH:/mnt/c/Users/Dolev_Lenovo/Desktop/fuckufinish#
```

 404 Not Found

כך נראה האתר בדפדפן

← → ↻ ⚠ Not secure | info.cern.ch

http://info.cern.ch - home of the first website

From here you can:

- [Browse the first website](#)
- [Browse the first website using the line-mode browser simulator](#)
- [Learn about the birth of the web](#)
- [Learn about CERN, the physics laboratory where the web was born](#)

← → ↻ ⚠ Not secure | info.cern.ch/doesnotexist

Not Found

The requested URL /doesnotexist was not found on this server.