Summary exercise

ID: 314915778

UDP programming

Problem 1: Learn to use datagram sockets by example.

Work as specified. comments in the code.

How to run:

a. Type in terminal: make first
b. split the terminal to 3 or run-in background
c. ./recv 127.0.0.1 (end with ctrl c)
d. ./send for each massage to recv

Problem 2: Create a gateway process that simulates datagram loss.

Work as specified. comments in the code.

Source.c : received the name of a host & port number on the command line, creates a datagram socket to that host, then enters an infinite loop in each iteration of which it sends a datagram onto the socket carrying in its body an integer number, increments the integer, then sleeps for one second only to repeat the cycle upon waking up.

Gateway.c: received the name of a host & port number on the command line and creates a datagram socket to that host (using port number P+1), also creates another datagram socket where it can receive datagrams from any host on port number P. next, it enters an infinite loop in each iteration of which it receives a datagram from port P, then samples a random number using **`((float)random())/((float)RAND_MAX)`** - if the number obtained is greater than 0.5, the datagram received is forwarded onto the outgoing socket to port P+1, otherwise the datagram is discarded and the process goes back to waiting for another incoming datagram. Note that this gateway will simulate an unreliable network that loses datagrams with 50% probability.

Sink.c: received port number on the command line, creates a socket to receive datagrams from any host on port P+1, then enters an infinite loop where it receives a datagram and prints to the screen the information of where the datagram came from -IP address in dotted-decimal notation & message it contains.

How to run:

a. make sec
b. ./sink port
c. ./gateway hostname port
d. ./source hostname port+1

TCP programming

Part A: IP addresses, hostnames and .... HTTP

After Compile both programs (net_server.c and net_client.c), we can see that the connection don`t work (seg fault) because we need to pass IP address.

After change the definition in `net_client.c`, of IP_ADDRESS (using nslookup) so that it is the address of the computer you are working on and recompile,
We can see that the connection work and server send message to the client:

```
dvirb@DESKTOP-MHAH2O2:~/comm_final$ gcc net_server.c -o serv
dvirb@DESKTOP-MHAH2O2:~/comm_final$ ./serv
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.
dvirb@DESKTOP-MHAH2O2:~/comm_final$ []
```

```
dvirb@DESKTOP-MHAH2O2:~/comm_final$ gcc net_client.c -o cli
dvirb@DESKTOP-MHAH2O2:~/comm_final$ ./cli
Client is alive and establishing socket connection.
Client has received 1 from socket.
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.
dvirb@DESKTOP-MHAH2O2:~/comm_final$ []
```

If we run client while the server isn't run the connection denied cause the socket that we want to connect to don't exist.

Part B: A simple web client

if the URL is http://www.yahoo.com

then the data begin with: Content-Security-Policy-Report-Only

```
gcc web_client.c -o wc
dvirx@ubuntu:~/Desktop/comm_final$ ./wc http://www.yahoo.com
Client is alive and establishing socket connection.
HTTP/1.0 301 Moved Permanently
Date: Sun, 31 Jul 2022 09:12:29 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com h
ttps://*.yahoo.com https://*.aol.com https://*.huffingtonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com h
ttps://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentati
on; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=6b04dl1hechrt&partner=;
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; report="https://csp.yahoo.com/beacon/csp?src=fp-hpkp-www"
Location: https://www.yahoo.com/www.yahoo.com
Content-Length: 8

redirect
Exiting now.
```

else, if the URL is : http://www.yahoo.com/does-not-exist

then the data begin with: "Content-Security-Policy

```
dvirx@ubuntu:~/Desktop/comm_final$ ./wc http://www.yahoo.com/does-not-exist
Client is alive and establishing socket connection.
HTTP/1.0 301 Moved Permanently
Date: Sun, 31 Jul 2022 09:24:36 GMT
Server: ATS
Cache-Control: no-store, no-cache
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Content-Security-Policy-Report-Only: connect-src 'self' https://s.yimg.com https://rtr.innovid.com https://*.yahoo.com https://*.oath.com https://*.advertising.com https://*.cdn.yimg.com https
://*.yahoo.net https://ad.doubleclick.net; style-src https://*.btrll.com https://cdn.cmp.advertising.com https://*.oath.com https://*.yimg.com https://*.yahoo.com 'self' 'unsafe-inline' data:
blob:; script-src https://*.btrll.com https://cdn.cmp.advertising.com https://*.oath.com https://*.yahoo.com https://*.yimg.com 'self' 'unsafe-inline' https://platform.twitter.com https://*.bt
rll.com data: blob:; img-src 'self' data: blob: about: https://*.yimg.com https://*.yahoo.com https://sb.scorecardresearch.com https://*.doubleclick.net https://*.adsafeprotected.com https://*
.googlesyndication.com; frame-ancestors https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com https://*
.huffingtonpost.com https://www.aol.com https://*.yahoo.com https://*.autoblog.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com ; s
andbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentation; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&
lang=en-US&device=&yrid=6h3qbodheciik&partner=;
Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com h
ttps://*.yahoo.com https://*.aol.com https://*.huffingtonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com h
ttps://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presentati
on; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=6h3qbodheciik&partner=;
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; report="https://csp.yahoo.com/beacon/csp?src=fp-hpkp-www"
Location: https://www.yahoo.com/does-not-exist
Content-Length: 8

redirect
Exiting now.
```

How to run:

a. make url
b. ./url and the URL address that we want to get info on.