TARBE Théophile
QUIDET Victor

ING4 SI
GR 01 (Inter)

- LAB3 -
Sockets and Network Programming

Professor : M. ZITOUNI

22/11/2020

I. Objective

1. Socket network programming
2. Review client/server approach.

- Each group (composed of **3 students** at most) shall submit a report **on campus.**
- The report (**PDF format is accepted**) shall be uploaded on the campus page before the deadline
  November 22 at 23h55.
- The source code and the exec files of the server (websrv.c and websrv) as well as the index.html should be zipped in one file. Please propose a small html file.

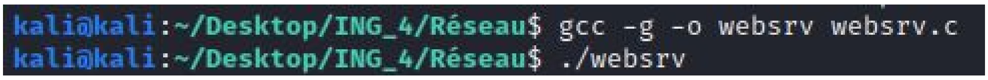
II.Web server

1.Skeleton

Q1: explain why we are using the library <netinet/in.h >

This library defines multiple structures that will be helpful in the development of our web server. It is necessary to define IP addresses, ports, ICPM, TCP and UDP.
It also defines functions such as `connect ()`, `sendmsg ()` and `sendto ()`.

Q2: Add the following line,compile and execute the server code. What is the required Linux command line and the name of the Compiler.



```
kali@kali:~/Desktop/ING_4/Réseau$ gcc -g -o websrv websrv.c
kali@kali:~/Desktop/ING_4/Réseau$ ./websrv
```

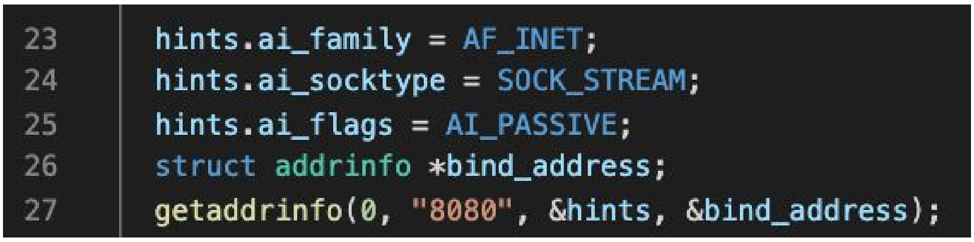
The name of the compiler is `gcc`.

2.Configuring local address and creating the socket

Q3: What would be the suitable domain in our case the domain, the type and the port number?

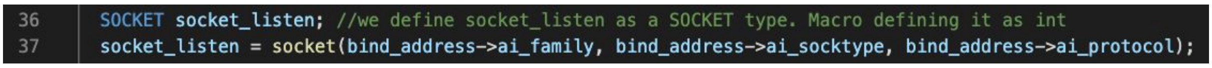
In our case, we should use :

- domain : `AF_INET` because we are looking for an IPv4 address.
- type : `SOCK_STREAM` because we're going to be using TCP. (For UDP it would have been `SOCK_DGRAM`).
- port number : `8080` is the port number of a service.



```c
hints.ai_family = AF_INET;
hints.ai_socktype = SOCK_STREAM;
hints.ai_flags = AI_PASSIVE;
struct addrinfo *bind_address;
getaddrinfo(0, "8080", &hints, &bind_address);
```

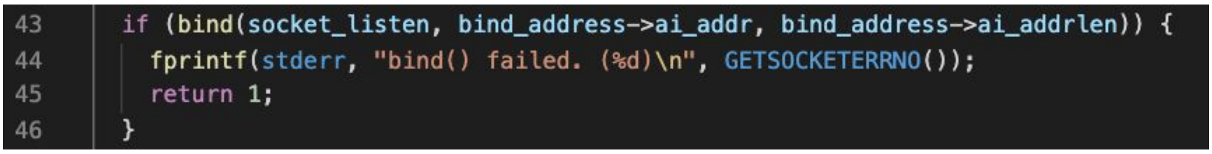
Q4: Complete the instruction of line 37.



```c
36  SOCKET socket_listen; //we define socket_listen as a SOCKET type. Macro defining it as int
37  socket_listen = socket(bind_address->ai_family, bind_address->ai_socktype, bind_address->ai_protocol);
```

3. Socket binding

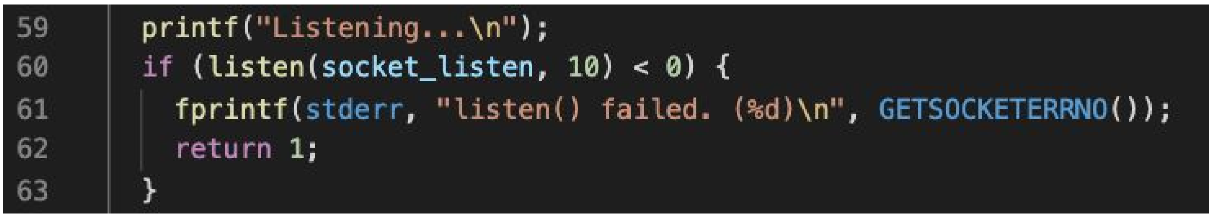
Q5: Add the source code to manage the error of the binding function.



```c
if (bind(socket_listen, bind_address->ai_addr, bind_address->ai_addrlen)) {
  fprintf(stderr, "bind() failed. (%d)\n", GETSOCKETERRNO());
  return 1;
}
```

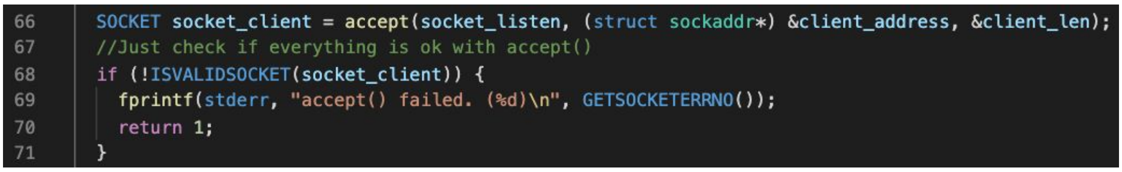
4. Socket listening

Q6: Here you should add your code to manage the error of socket listening.



```c
printf("Listening...\n");
if (listen(socket_listen, 10) < 0) {
  fprintf(stderr, "listen() failed. (%d)\n", GETSOCKETERRNO());
  return 1;
}
```

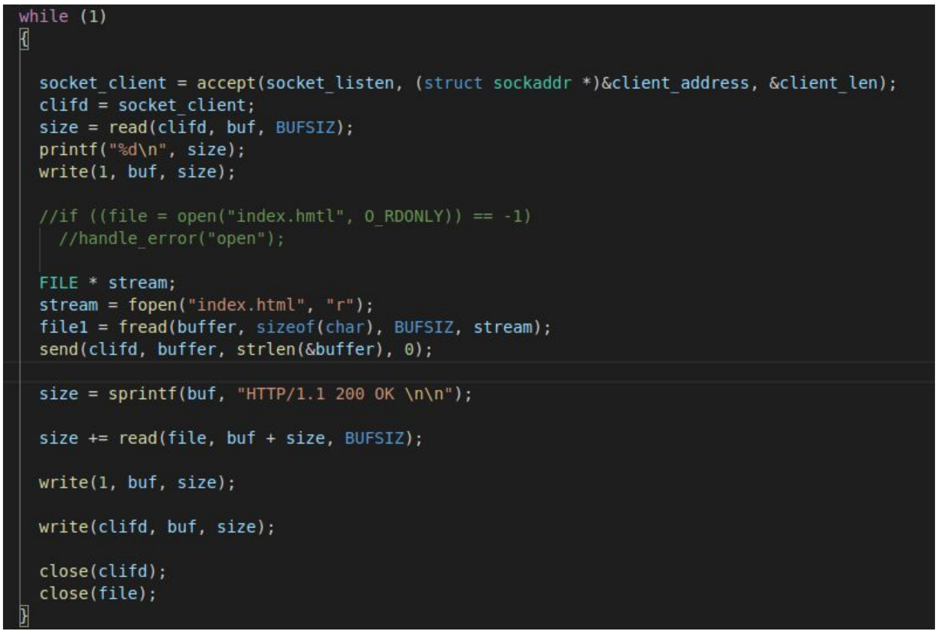
5. Socket acceptance



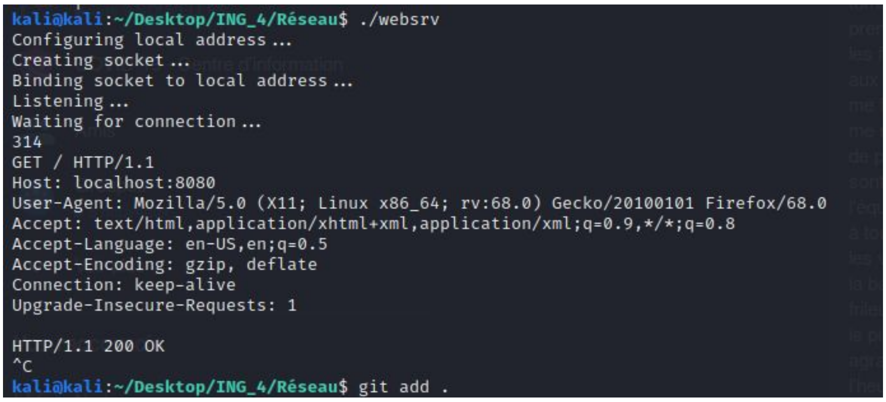
```c
SOCKET socket_client = accept(socket_listen, (struct sockaddr*) &client_address, &client_len);
//Just check if everything is ok with accept()
if (!ISVALIDSOCKET(socket_client)) {
  fprintf(stderr, "accept() failed. (%d)\n", GETSOCKETERRNO());
  return 1;
}
```

III. Connection test

Q7: Now, you will surround your accept function call by an infinite while loop (a server program should never quit until it is interrupted). Add the following code to make your server program able to serve an "index.html" file through the HTTP protocol. The “index.html” could be the simple html file.

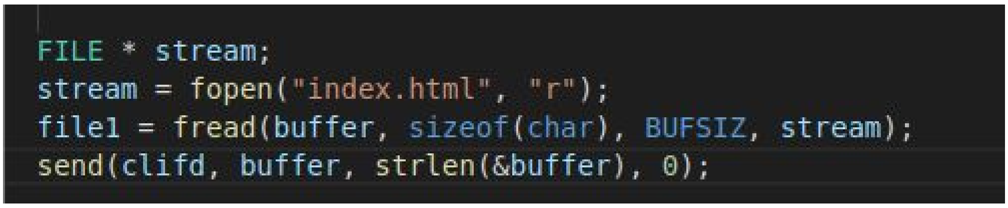


```c
while (1)
{

  socket_client = accept(socket_listen, (struct sockaddr *)&client_address, &client_len);
  clifd = socket_client;
  size = read(clifd, buf, BUFSIZ);
  printf("%d\n", size);
  write(1, buf, size);

  //if ((file = open("index.hmtl", O_RDONLY)) == -1)
    //handle_error("open");

  FILE * stream;
  stream = fopen("index.html", "r");
  file1 = fread(buffer, sizeof(char), BUFSIZ, stream);
  send(clifd, buffer, strlen(&buffer), 0);

  size = sprintf(buf, "HTTP/1.1 200 OK \n\n");

  size += read(file, buf + size, BUFSIZ);

  write(1, buf, size);

  write(clifd, buf, size);

  close(clifd);
  close(file);
}
```



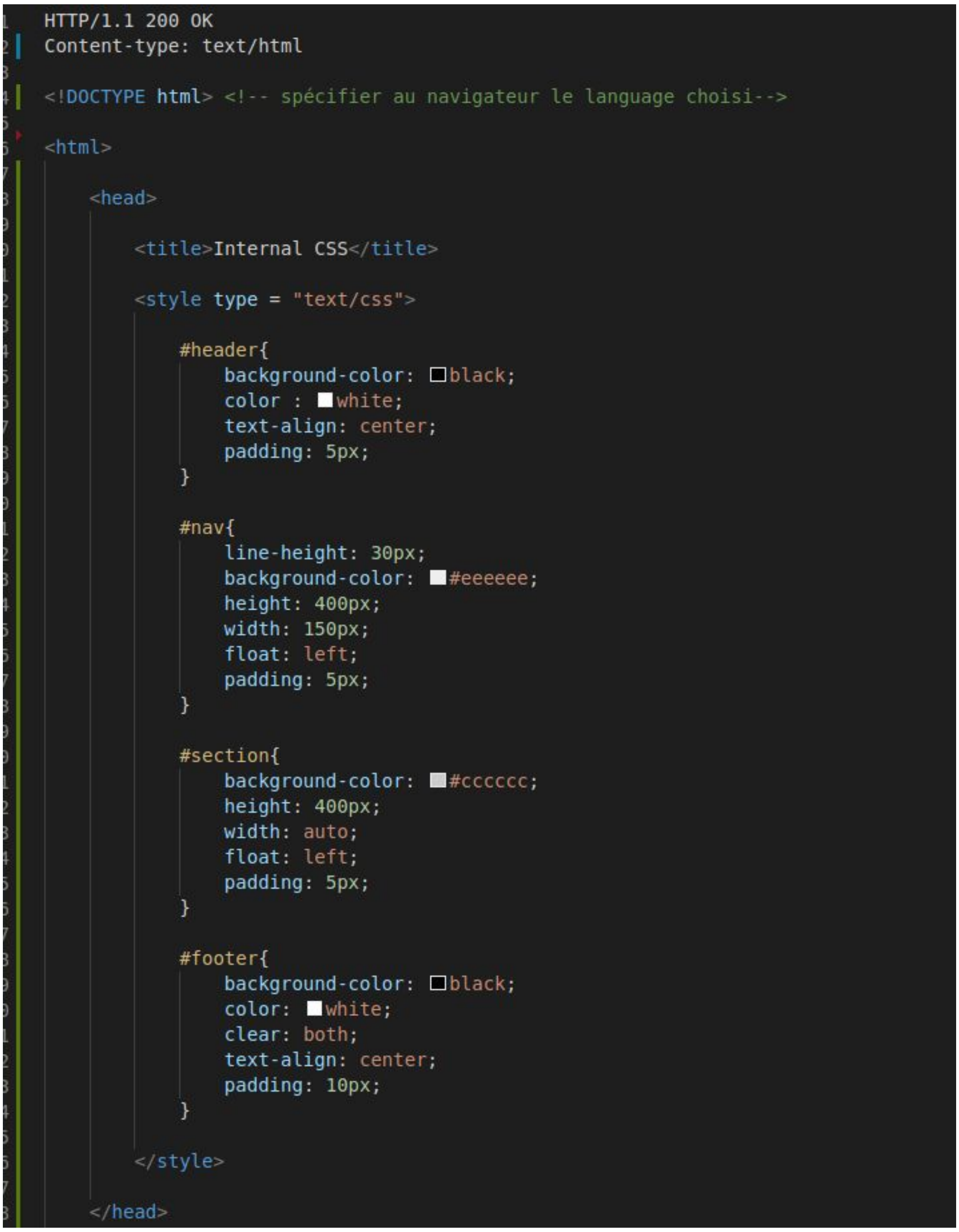
```
kali@kali:~/Desktop/ING_4/Réseau$ ./websrv
Configuring local address ...
Creating socket ...
Binding socket to local address ...
Listening ...
Waiting for connection ...
314
GET / HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
^C
kali@kali:~/Desktop/ING_4/Réseau$ git add .
```

Q8 (Bonus) : The last objective is to implement the send function to share html page with the client.

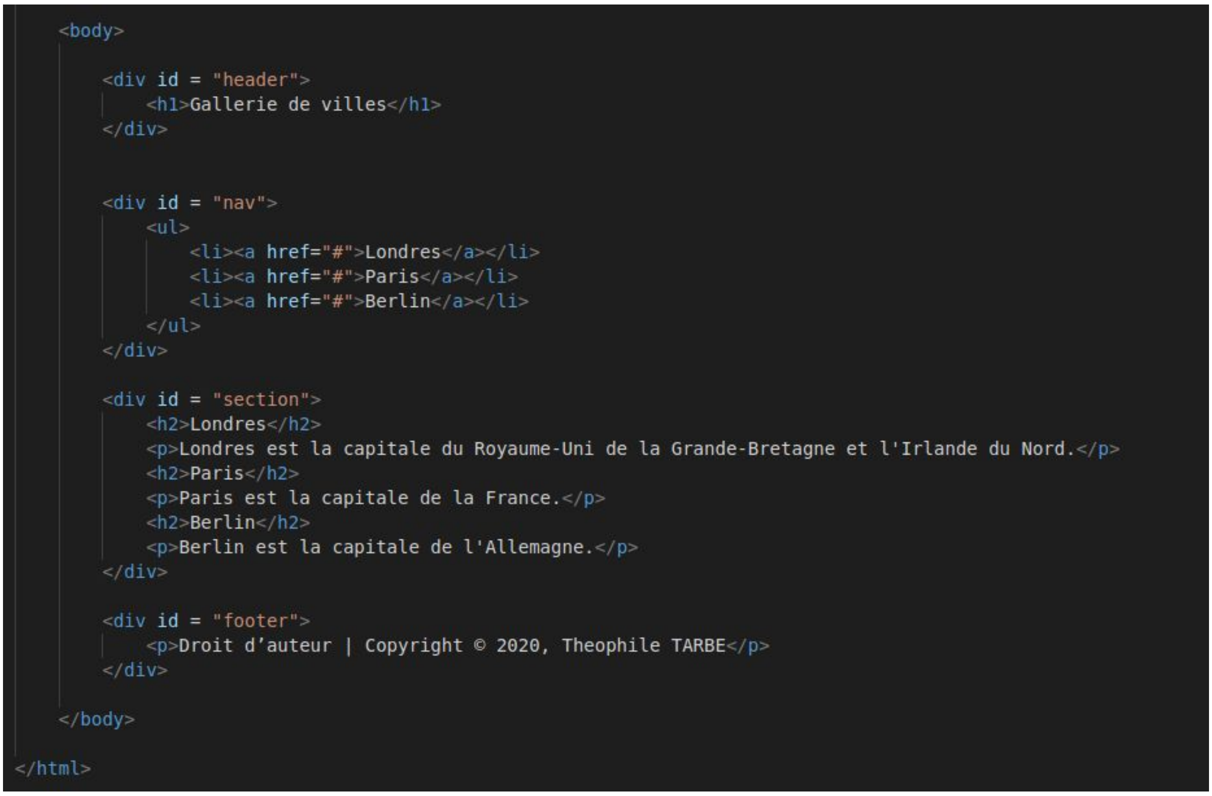


```
FILE * stream;
stream = fopen("index.html", "r");
file1 = fread(buffer, sizeof(char), BUFSIZ, stream);
send(clifd, buffer, strlen(&buffer), 0);
```

<u>Web_server.c</u>

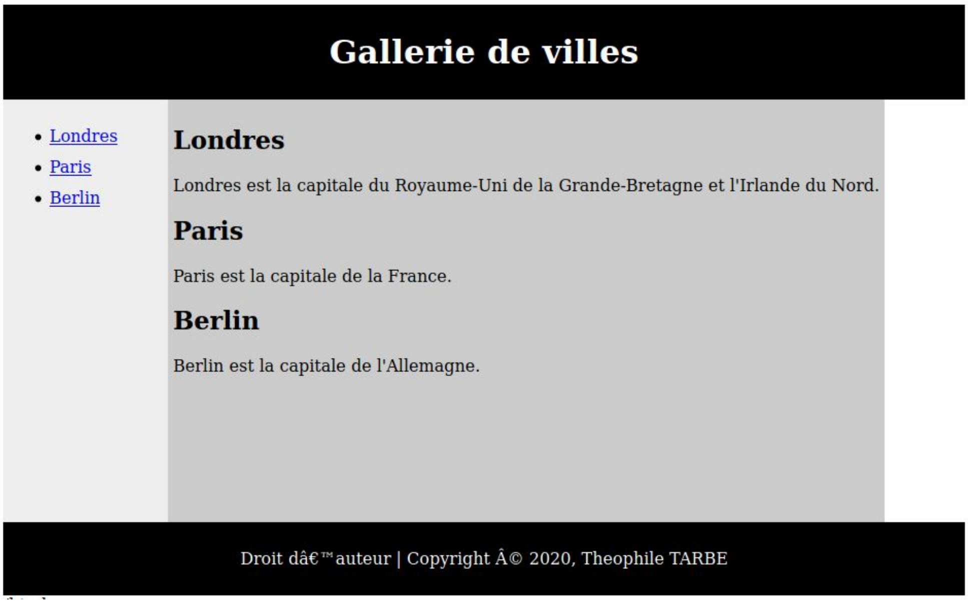


```
HTTP/1.1 200 OK
Content-type: text/html

<!DOCTYPE html> <!-- spécifier au navigateur le language choisi-->

<html>

    <head>

        <title>Internal CSS</title>

        <style type = "text/css">

            #header{
                background-color: ■black;
                color : ■white;
                text-align: center;
                padding: 5px;
            }

            #nav{
                line-height: 30px;
                background-color: ■#eeeeee;
                height: 400px;
                width: 150px;
                float: left;
                padding: 5px;
            }

            #section{
                background-color: ■#cccccc;
                height: 400px;
                width: auto;
                float: left;
                padding: 5px;
            }

            #footer{
                background-color: ■black;
                color: ■white;
                clear: both;
                text-align: center;
                padding: 10px;
            }

        </style>

    </head>
```



```
    <body>

        <div id = "header">
            <h1>Gallerie de villes</h1>
        </div>


        <div id = "nav">
            <ul>
                <li><a href="#">Londres</a></li>
                <li><a href="#">Paris</a></li>
                <li><a href="#">Berlin</a></li>
            </ul>
        </div>

        <div id = "section">
            <h2>Londres</h2>
            <p>Londres est la capitale du Royaume-Uni de la Grande-Bretagne et l'Irlande du Nord.</p>
            <h2>Paris</h2>
            <p>Paris est la capitale de la France.</p>
            <h2>Berlin</h2>
            <p>Berlin est la capitale de l'Allemagne.</p>
        </div>

        <div id = "footer">
            <p>Droit d’auteur | Copyright © 2020, Theophile TARBE</p>
        </div>

    </body>

</html>
```

index.html



Gallerie de villes

- Londres
- Paris
- Berlin

Londres

Londres est la capitale du Royaume-Uni de la Grande-Bretagne et l'Irlande du Nord.

Paris

Paris est la capitale de la France.

Berlin

Berlin est la capitale de l'Allemagne.

Droit dâ€™auteur | Copyright Â© 2020, Theophile TARBE

localhost