SYSTEM PROGRAMMING

Final Project

Elifnur Kabalcı
1801042617

General Part

There is one server file in my project file. If the server file does not exist, the program creates it on its own. I created a client file for the number of tests.

As long as the server code is open, it detects the connection request from any file and acts accordingly.

When we try to connect a client that exceeds the given threadpool size, it gives an error and closes the run line of the client trying to connect.

What am I done?

As long as the server code is open, I monitor the changes in the file with inotify. Thus, it transfers the changes that occur in the connected clients to the server in each round it examines. With Inotfiy, this process is watched live.

Server Part Design Steps

1) I did the checks for the argument received from the user. I assigned them to the values that I set earlier. I opened a string field that will save the connection path as many as the number of clients that can be connected with the connectedPath element. Thus, if an extra client wants to connect, the program will suppress an error.

2) Signal checks were done with a fixed signal handler, same as other assignments.

3) If the connection directory given as the server path is not found in the file location, I created it.

4) I created a socket connection to the specified server directory value. I made this creation inside the if condition and provided its controls. I assigned address statements to provide socket connection. I added listen to the server file direction to which I assigned the socket connection, and I defined a thread for each client and created these threads.

5) I switched to Thread creation method. Here I assigned the data addressing of server fd to socketfd. Afterwards, if the connection is tried to be established despite the maximum number of clients being reached, a warning is given that the socket connections are full.

6) Then “yes” is sent to the client connected with socket fd. There is a part in the client part that will capture and evaluate it. This indicates that the connection has been established. Thus, the total number of clients can now be increased.

7) I often fflush stdout. In such a buffer, there is no accumulation of data to be printed on the screen. In case more than one client is connected, the data to be printed on the screen is not mixed with each other.

8) With recv, a path information is obtained with socketfd. This received information is checked in the for loop. This control is to determine whether the mentioned path is used by another client. Here, “no” indicates that this path is used by someone else. Therefore, the program is closed. If this is not the case, it returns to the client by saying "yes" and continues the program. To save this link, the confirmed path is saved in a directory array.

9) I opened a log file to record the movements made. I printed both the log file and the screen that the server accepted the connection request sent by the client.

10) As long as the server is open, I set up a read loop so that it can read the data from the clients. I split the data read with Read into data with sscanf. So I was able to use this data at separate points.

11) If the command received and read from the user is “file”; The received file size is converted from string to int. Then it is checked whether the file name is server.log. Because this file is created later to save data and it is not required to be copied and deleted to other parts. Therefore, the "no" response is sent to the client and removed from the while.

12) Paths to work on are prepared. The file defined as hidden is opened for reading. After the file is checked, it is opened and read. Then the status of the current path is updated and the file is marked as modified because it has been read. If the opened and read file is not empty, its contents are also copied. Thus, the file is completely moved.

13) If command filedelete, I remove the pointer of the determined path by unlinking the determined path and prevent the pointer garbage by freeing the defined paths.

14) If it is a command folder, I created the direction in the specified path and printed its notifications.

15) If command is folderdelete then remove directory method is called. The specified direction is opened and the directory is read and unlinked until it is finished.

16) If the command is getmissinggfiles, the server tells the client that there are missing files and the sendMissingFiles method is called to complete these missing files. For this method, the directory is examined and different ones are added, similar to the operations we have done before.

Client Part Design Steps

1) The data received from the user is examined and the number of arguments is checked. I defined the signal handler as usual. Then I assigned the arguments to variables.

2) The stat is reset and it is determined whether the specified client area exists in the current location, if not, it is removed from the program. The method is called to connect to the client server.

3) In this method, the socket address is defined in the same way as in the server part. It is checked if the socket could not be defined. Then the server address structure is reset with the I memset method. Then, socket connection is tried to be established by using connection protocols.

4) Trying to connect to the server with Socket, if not, the error is suppressed.

5) If the receiver buffer is “yes”, yes is returned because the maximum number of clients in the server part is not full. Accordingly, information is printed on the screen.

6) If yes is not returned but max statement is displayed, this indicates that the server to which you want to connect has reached the maximum number of clients. If it cannot be connected due to another problem, the socket is closed again and the error is suppressed.

7) I defined Inotify. Thus, it allows us to measure the difference and intervene at all moments when the client is open and connected to the server.

8) While listening to the server with inotify, I read the data from the server with buffer2 with a while loop. Then an event is created with the listened and received data. The information of the event is checked. For example, the length of the event must be greater than zero, the event mask must be a directory, and it must match create.

9) Data about the event is sent to the server using snprintf.

10) With watchfd, the events that inotfiy should monitor are specified. Events such as new file creation, deletion, modification to the file and moving of the file are examined here.

11) A listener is added to the working path by using inotify. Thus, it can send the received data to watchfd and continue the process.

12) I used synchronization methods similar to the server file for the synchronization of the determined changes to each other.

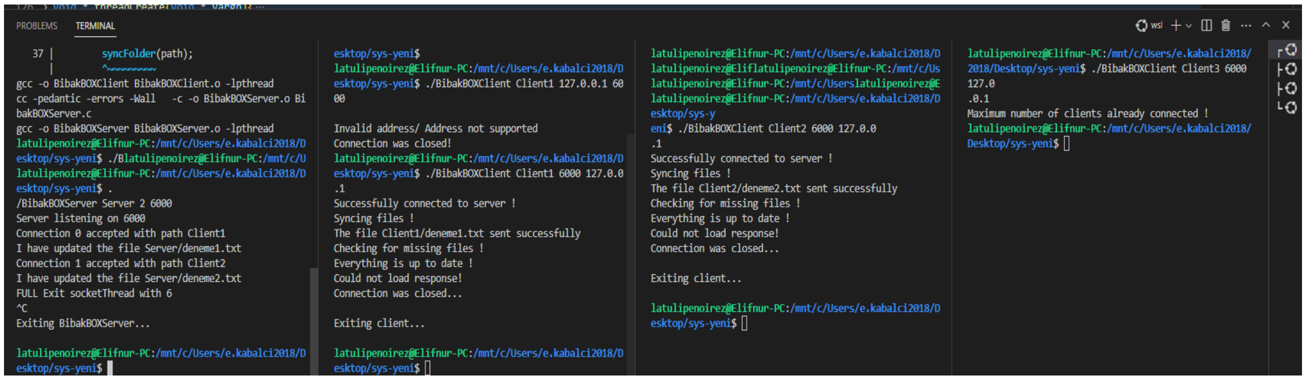
13) I used if conditions to decide which processes specified in watchfd match.

Run Lines:

Server side: `BibakBOXServer [directory] [threadPoolSize] [portnumber]`

Client side: `BibakBOXClient [clientDirPath] [portnumber] (optional)[server ip]`

Outputs

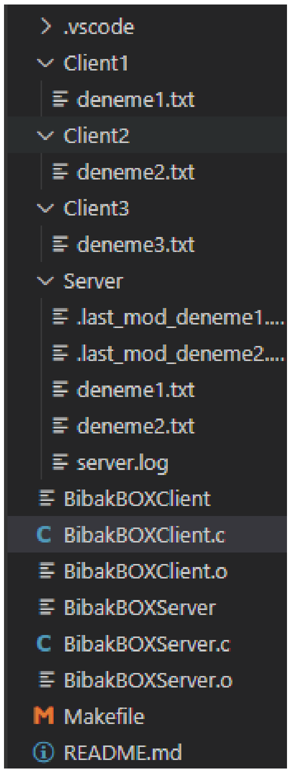


```
   37 |         syncFolder(path);
      |         ^~~~~~~~~~
gcc -o BibakBOXClient BibakBOXClient.o -lpthread
cc -pedantic -errors -Wall   -c -o BibakBOXServer.o Bi
bakBOXServer.c
gcc -o BibakBOXServer BibakBOXServer.o -lpthread
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$ ./Blatulipenoirez@Elifnur-PC:/mnt/c/U
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$ .
/BibakBOXServer Server 2 6000
Server listening on 6000
Connection 0 accepted with path Client1
I have updated the file Server/deneme1.txt
Connection 1 accepted with path Client2
I have updated the file Server/deneme2.txt
FULL Exit socketThread with 6
^C
Exiting BibakBOXServer...

latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$
```

```
esktop/sys-yeni$
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$ ./BibakBOXClient Client1 127.0.0.1 60
00

Invalid address/ Address not supported
Connection was closed!
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$ ./BibakBOXClient Client1 6000 127.0.0
.1
Successfully connected to server !
Syncing files !
The file Client1/deneme1.txt sent successfully
Checking for missing files !
Everything is up to date !
Could not load response!
Connection was closed...

Exiting client...

latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$
```

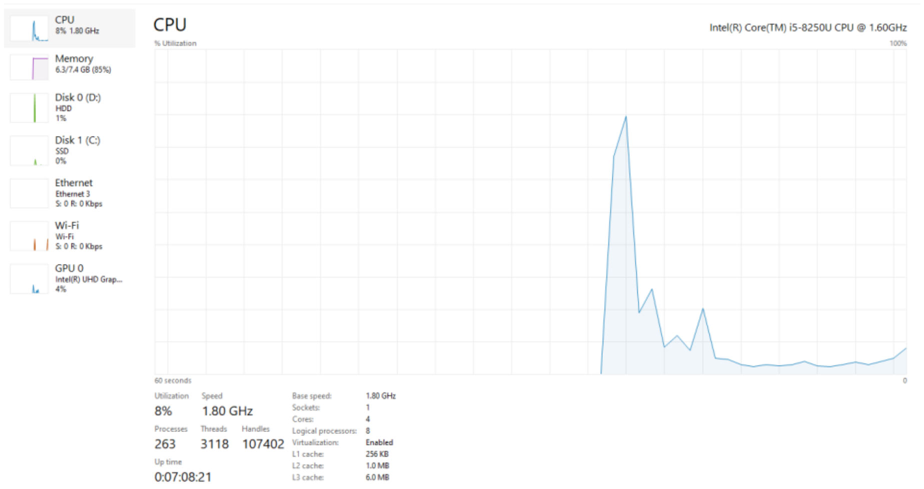
```
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
latulipenoirez@Eliflatulipenoirez@Elifnur-PC:/mnt/c/Us
latulipenoirez@Elifnur-PC:/mnt/c/Userslatulipenoirez@E
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-y
eni$ ./BibakBOXClient Client2 6000 127.0.0
.1
Successfully connected to server !
Syncing files !
The file Client2/deneme2.txt sent successfully
Checking for missing files !
Everything is up to date !
Could not load response!
Connection was closed...

Exiting client...

latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/D
esktop/sys-yeni$
```

```
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/
2018/Desktop/sys-yeni$ ./BibakBOXClient Client3 6000
127.0
.0.1
Maximum number of clients already connected !
latulipenoirez@Elifnur-PC:/mnt/c/Users/e.kabalci2018/
Desktop/sys-yeni$
```



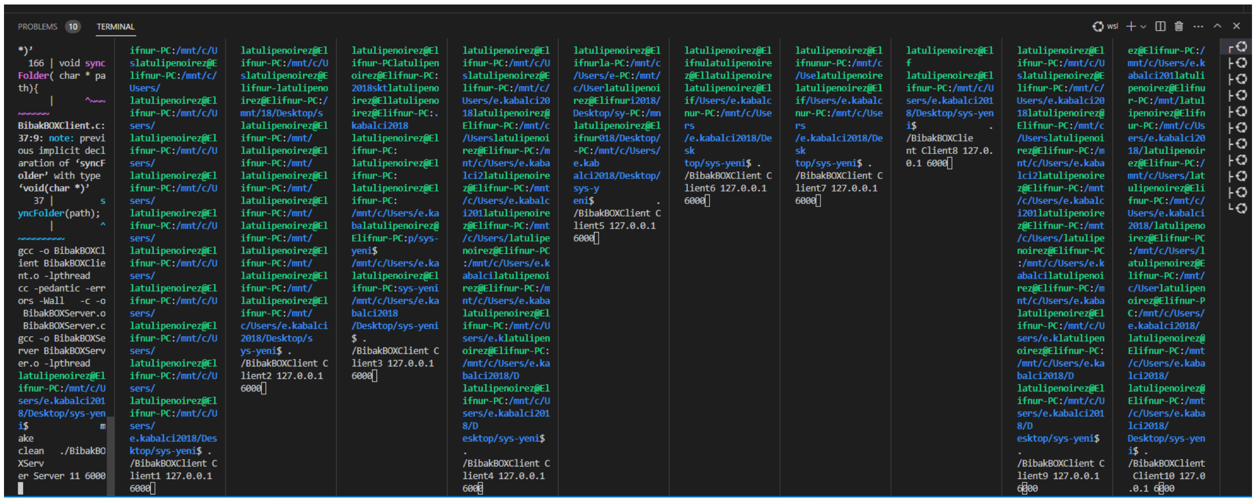
```
> .vscode
v Client1
    deneme1.txt
v Client2
    deneme2.txt
v Client3
    deneme3.txt
v Server
    .last_mod_deneme1....
    .last_mod_deneme2....
    deneme1.txt
    deneme2.txt
    server.log
  BibakBOXClient
  BibakBOXClient.c
  BibakBOXClient.o
  BibakBOXServer
  BibakBOXServer.c
  BibakBOXServer.o
  Makefile
  README.md
```

-> Output of the files



```
Utilization   Speed                 Base speed:          1.80 GHz
8%            1.80 GHz              Sockets:             1
                                    Cores:               4
Processes     Threads    Handles    Logical processors:  8
263           3118       107402     Virtualization:      Enabled
Up time                             L1 cache:            256 KB
0:07:08:21                          L2 cache:            1.0 MB
                                    L3 cache:            6.0 MB
```

➔ CPU using is the last part. Server and 2 client open and it use cpu's %3.



```
*)'
  166 | void sync
Folder( char * pa
th){
      |        ^~~~
~~~~~~
BibakBOXClient.c:
37:9: note: previ
ous implicit decl
aration of 'syncF
older' with type
'void(char *)'
   37 |         s
yncFolder(path);
      |          ^
~~~~~~~~~~
gcc -o BibakBOXCl
ient BibakBOXClie
nt.o -lpthread
cc -pedantic -err
ors -Wall   -c -o
 BibakBOXServer.o
 BibakBOXServer.c
gcc -o BibakBOXSe
rver BibakBOXServ
er.o -lpthread
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/e.kabalci201
8/Desktop/sys-yen
i$              m
ake
clean   ./BibakBO
XServ
er Server 11 6000
█
```

```
ifnur-PC:/mnt/c/U
slatulipenoirez@E
lifnur-PC:/mnt/c/
Users/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/
e.kabalci2018/Des
ktop/sys-yeni$ .
/BibakBOXClient C
lient1 127.0.0.1
6000▯
```

```
latulipenoirez@El
ifnur-PC:/mnt/c/U
slatulipenoirez@E
lifnur-latulipeno
irez@Elifnur-PC:/
mnt/18/Desktop/s
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
latulipenoirez@El
ifnur-PC:/mnt/
c/Users/e.kabalci
2018/Desktop/s
ys-yeni$ .
/BibakBOXClient C
lient2 127.0.0.1
6000▯
```

```
latulipenoirez@El
ifnur-PClatulipen
oirez@Elifnur-PC:
2018sktlatulipeno
irez@Ellatulipeno
irez@Elifnur-PC:.
kabalci2018
latulipenoirez@El
ifnur-PC:
latulipenoirez@El
ifnur-PC:
latulipenoirez@El
ifnur-PC:
/mnt/c/Users/e.ka
balatulipenoirez@
Elifnur-PC:p/sys-
yeni$
/mnt/c/Users/e.ka
latulipenoirez@El
ifnur-PC:sys-yeni
/mnt/c/Users/e.ka
balci2018
/Desktop/sys-yeni
$ .
/BibakBOXClient C
lient3 127.0.0.1
6000▯
```

```
latulipenoirez@El
ifnur-PC:/mnt/c/U
slatulipenoirez@E
lifnur-PC:/mnt/c/
Users/e.kabalci20
18latulipenoirez@
Elifnur-PC:/mnt/c
/Userslatulipenoi
rez@Elifnur-PC:/m
nt/c/Users/e.kaba
lci2latulipenoire
z@Elifnur-PC:/mnt
/c/Users/e.kabalc
i201latulipenoire
z@Elifnur-PC:/mnt
/c/Users/latulipe
noirez@Elifnur-PC
:/mnt/c/Users/e.k
abalcilatulipenoi
rez@Elifnur-PC:/m
nt/c/Users/e.kaba
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/e.klatulipen
oirez@Elifnur-PC:
/mnt/c/Users/e.ka
balci2018/D
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/e.kabalci201
8/D
esktop/sys-yeni$
.
/BibakBOXClient C
lient4 127.0.0.1
6000
```

```
latulipenoirez@El
ifnurla-PC:/mnt/c
/Users/e-PC:/mnt/
c/Userlatulipenoi
rez@Elifnuri2018/
Desktop/sy-PC:/mn
latulipenoirez@El
ifnur018/Desktop/
-PC:/mnt/c/Users/
e.kab
alci2018/Desktop/
sys-y
eni$            .
/BibakBOXClient C
lient5 127.0.0.1
6000▯
```

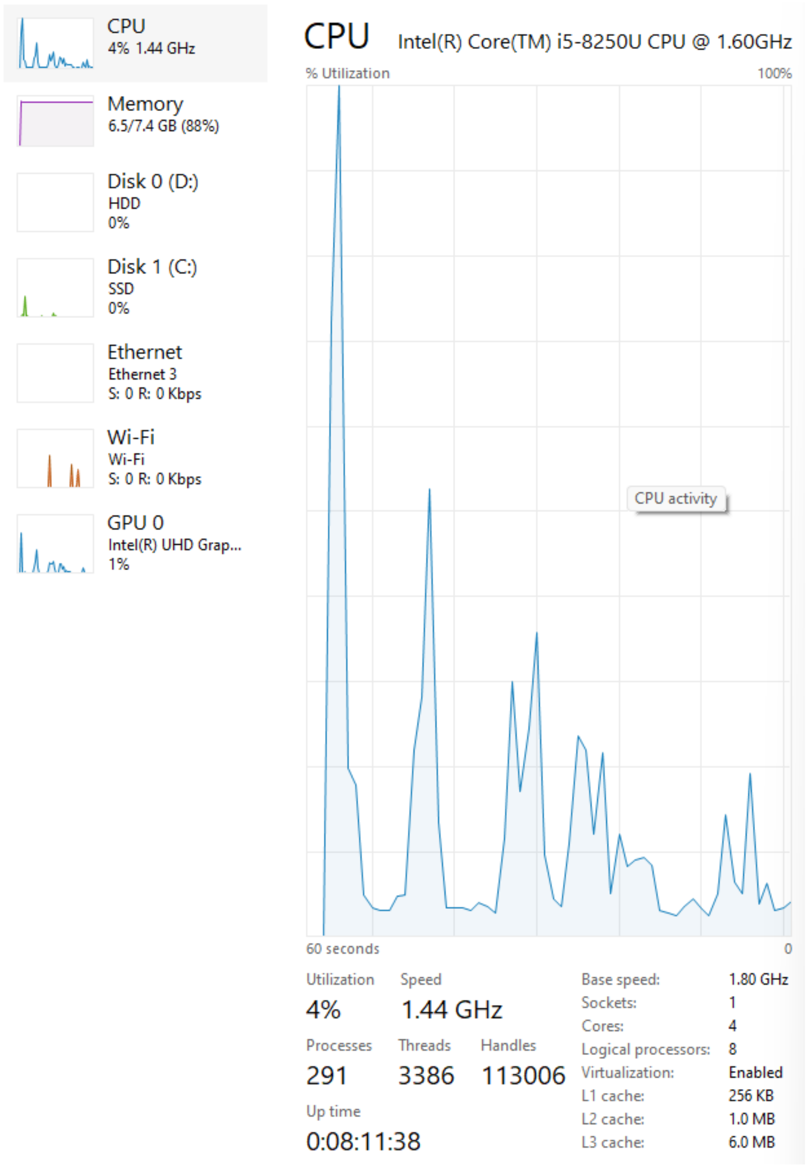
```
latulipenoirez@El
ifnulatulipenoire
z@Ellatulipenoire
latulipenoirez@El
if/Users/e.kabalc
nur-PC:/mnt/c/Use
rs
/e.kabalci2018/De
sk
top/sys-yeni$ .
/BibakBOXClient C
lient6 127.0.0.1
6000▯
```

```
latulipenoirez@El
ifnunur-PC:/mnt/c
/Uselatulipenoire
latulipenoirez@El
if/Users/e.kabalc
nur-PC:/mnt/c/Use
rs
/e.kabalci2018/De
sk
top/sys-yeni$ .
/BibakBOXClient C
lient7 127.0.0.1
6000▯
```

```
latulipenoirez@El
f
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/e.kabalci201
8/Desktop/sys-yen
i$              .
/BibakBOXClie
nt Client8 127.0.
0.1 6000▯
```

```
latulipenoirez@El
ifnur-PC:/mnt/c/U
slatulipenoirez@E
lifnur-PC:/mnt/c/
Users/e.kabalci20
18latulipenoirez@
Elifnur-PC:/mnt/c
/Userslatulipenoi
rez@Elifnur-PC:/m
nt/c/Users/e.kaba
lci2latulipenoire
z@Elifnur-PC:/mnt
/c/Users/e.kabalc
i201latulipenoire
z@Elifnur-PC:/mnt
/c/Users/latulipe
noirez@Elifnur-PC
:/mnt/c/Users/e.k
abalcilatulipenoi
rez@Elifnur-PC:/m
nt/c/Users/e.kaba
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/e.klatulipen
oirez@Elifnur-PC:
/mnt/c/Users/e.ka
balci2018/D
latulipenoirez@El
ifnur-PC:/mnt/c/U
sers/e.kabalci201
8/D
esktop/sys-yeni$
.
/BibakBOXClient C
lient9 127.0.0.1
6000
```

```
ez@Elifnur-PC:/
mnt/c/Users/e.k
abalci201latuli
penoirez@Elifnu
r-PC:/mnt/latul
ipenoirez@Elifn
ur-PC:/mnt/c/Us
ers/e.kabalci20
18/latulipenoir
ez@Elifnur-PC:/
mnt/c/Users/lat
ulipenoirez@Eli
fnur-PC:/mnt/c/
Users/e.kabalci
2018/latulipeno
irez@Elifnur-PC
:/mnt/c/Users/l
atulipenoirez@E
lifnur-PC:/mnt/
c/Userlatulipen
oirez@Elifnur-P
C:/mnt/c/Users/
e.kabalci2018/
latulipenoirez@
Elifnur-PC:/mnt
/c/Users/e.kaba
lci2018/
latulipenoirez@
Elifnur-PC:/mnt
/c/Users/e.kaba
lci2018/
Desktop/sys-yen
i$ .
/BibakBOXClient
 Client10 127.0
.0.1 6000
```



CPU
4% 1.44 GHz

Memory
6.5/7.4 GB (88%)

Disk 0 (D:)
HDD
0%

Disk 1 (C:)
SSD
0%

Ethernet
Ethernet 3
S: 0 R: 0 Kbps

Wi-Fi
Wi-Fi
S: 0 R: 0 Kbps

GPU 0
Intel(R) UHD Grap...
1%

CPU Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz

Utilization	Speed	
4%	1.44 GHz	
Processes	Threads	Handles
291	3386	113006
Up time		
0:08:11:38		

Base speed:	1.80 GHz
Sockets:	1
Cores:	4
Logical processors:	8
Virtualization:	Enabled
L1 cache:	256 KB
L2 cache:	1.0 MB
L3 cache:	6.0 MB

I connected 10 clients at the same time to examine CPU data. Max went up to 24%. The part shown as 100% belongs to the moment the activity monitoring page is opened. Later, it changed to the 12-3 band. Even though it increased to 12% for a moment when I said Ctrl+c, it decreased rapidly.

I record the Turkish videos for explain the code and show the compile and run parts. This is link:

Click: System_final_project_explanation

Or click: https://gtu-my.sharepoint.com/:f:/g/personal/e_kabalci2018_gtu_edu_tr/Ekq-eg9lakZAsMzUVL5kvfsBBi7pJPCwdVszpK8jWCAbFg?e=ZT3cCx