Assignment 3 - OS

Part A:

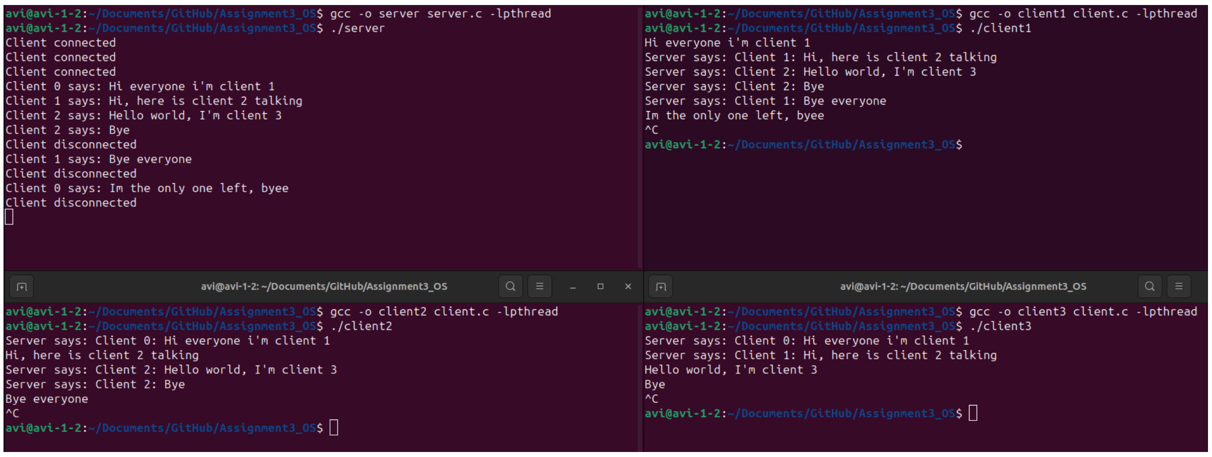
In this part, we will implement a client-server chat using async I/O which means threads.

The server has $n + 1$ sockets, which n is the number of clients that are connected to the server, and 1 more for the thread that is listening for incoming connections.

If there is a new connection, the number of threads increases by 1, and if one client ends his process, the number of threads decreases by 1.

The client has 2 threads - one for connecting with the server and another for listening to the keyboard (to prevent busy waiting).

Execution example with 3 clients:



avi@avi-1-2: ~/Documents/GitHub/Assignment3_OS
avi@avi-1-2: ~/Documents/GitHub/Assignment3_OS

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ gcc -o server server.c -lpthread
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ ./server
Client connected
Client connected
Client connected
Client 0 says: Hi everyone i'm client 1
Client 1 says: Hi, here is client 2 talking
Client 2 says: Hello world, I'm client 3
Client 2 says: Bye
Client disconnected
Client 1 says: Bye everyone
Client disconnected
Client 0 says: Im the only one left, byee
Client disconnected
```

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ gcc -o client1 client.c -lpthread
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ ./client1
Hi everyone i'm client 1
Server says: Client 1: Hi, here is client 2 talking
Server says: Client 2: Hello world, I'm client 3
Server says: Client 2: Bye
Server says: Client 1: Bye everyone
Im the only one left, byee
^C
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$
```

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ gcc -o client2 client.c -lpthread
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ ./client2
Server says: Client 0: Hi everyone i'm client 1
Hi, here is client 2 talking
Server says: Client 2: Hello world, I'm client 3
Server says: Client 2: Bye
Bye everyone
^C
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$
```

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ gcc -o client3 client.c -lpthread
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ ./client3
Server says: Client 0: Hi everyone i'm client 1
Server says: Client 1: Hi, here is client 2 talking
Hello world, I'm client 3
Bye
^C
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$
```

The compiling task is straightforward.

For the server, you should write:

```
gcc -o server server.c -lpthread
```

The `-lpthread` command is for the POSIX thread library.

For the client, you should write:

```
gcc -o client client.c -lpthread
```

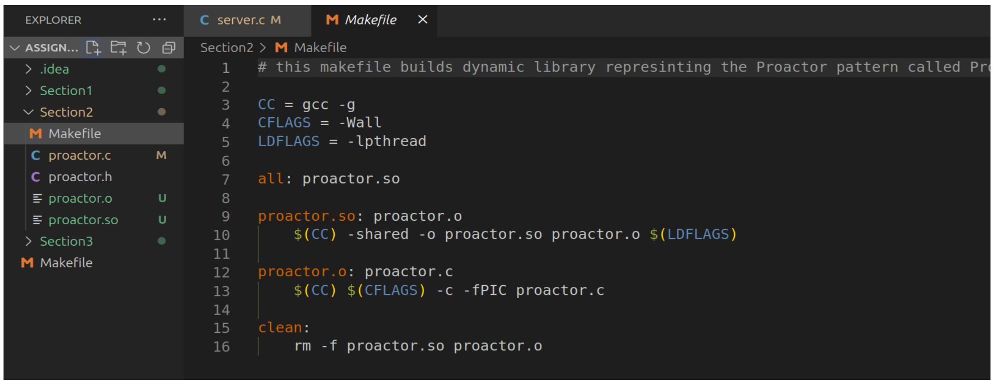
Part B:

In this part, we implement a general proactor for listening to one or more sockets (client connections).

If there is a new connection, the proactor will handle it by creating a new thread that takes care of that client.

The number of threads isn't known, it changes due to the number of connections.

This part is for the general proactor and for creating the proactor library.

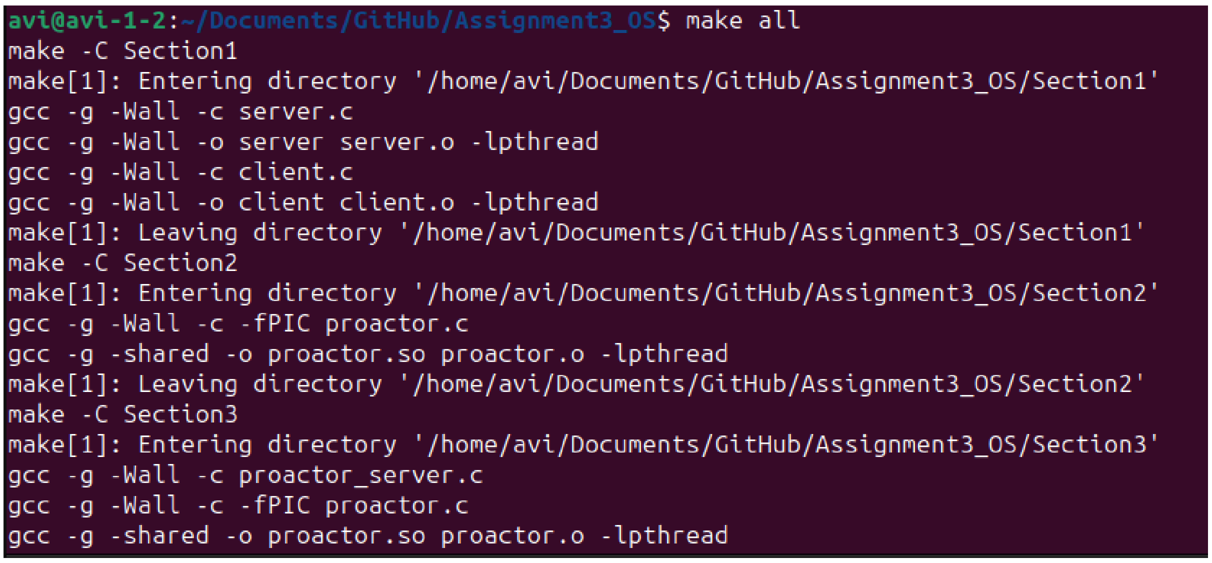


```
# this makefile builds dynamic library represinting the Proactor pattern called Pr

CC = gcc -g
CFLAGS = -Wall
LDFLAGS = -lpthread

all: proactor.so

proactor.so: proactor.o
	$(CC) -shared -o proactor.so proactor.o $(LDFLAGS)

proactor.o: proactor.c
	$(CC) $(CFLAGS) -c -fPIC proactor.c

clean:
	rm -f proactor.so proactor.o
```

Part C:

In this part, we will run the server from Part A with the general proactor implemented in Part B.

First of all, run `make all` to compile the necessary files in each section:

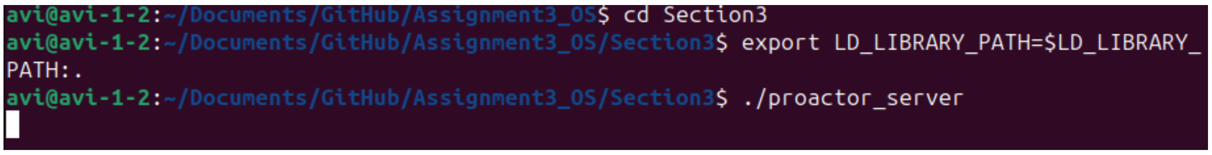


```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ make all
make -C Section1
make[1]: Entering directory '/home/avi/Documents/GitHub/Assignment3_OS/Section1'
gcc -g -Wall -c server.c
gcc -g -Wall -o server server.o -lpthread
gcc -g -Wall -c client.c
gcc -g -Wall -o client client.o -lpthread
make[1]: Leaving directory '/home/avi/Documents/GitHub/Assignment3_OS/Section1'
make -C Section2
make[1]: Entering directory '/home/avi/Documents/GitHub/Assignment3_OS/Section2'
gcc -g -Wall -c -fPIC proactor.c
gcc -g -shared -o proactor.so proactor.o -lpthread
make[1]: Leaving directory '/home/avi/Documents/GitHub/Assignment3_OS/Section2'
make -C Section3
make[1]: Entering directory '/home/avi/Documents/GitHub/Assignment3_OS/Section3'
gcc -g -Wall -c proactor_server.c
gcc -g -Wall -c -fPIC proactor.c
gcc -g -shared -o proactor.so proactor.o -lpthread
```

After that, get into Section 3 in each terminal with `cd Section3`.

and run these commands for the server:

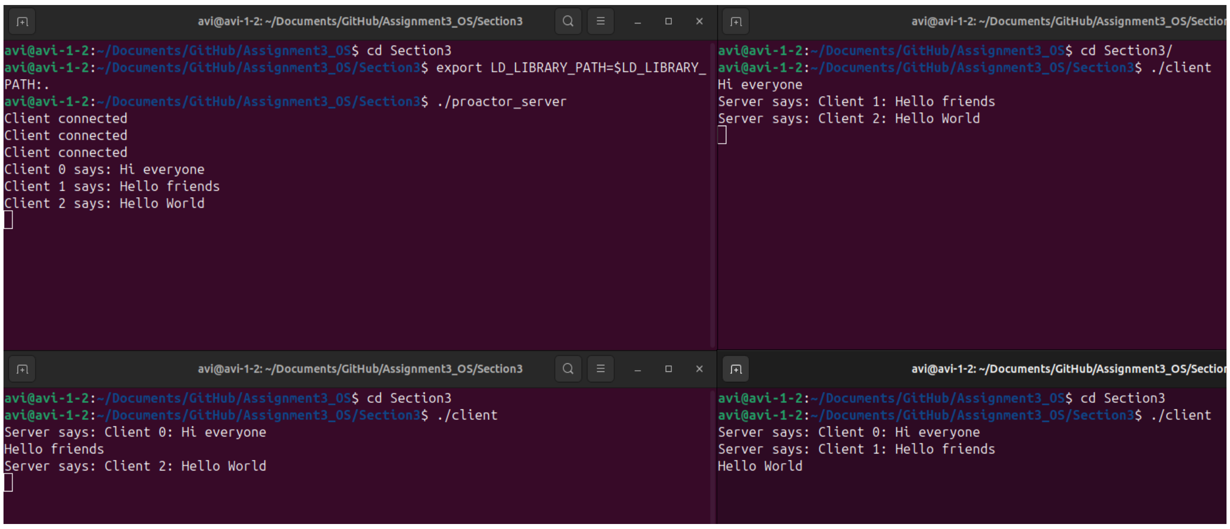
```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
./proactor_server
```



```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ cd Section3
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ export LD_LIBRARY_PATH=$LD_LIBRARY_
PATH:.
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ ./proactor_server
```

Now, to run the clients, we need to run this command:

```
./client
```



avi@avi-1-2: ~/Documents/GitHub/Assignment3_OS/Section3
avi@avi-1-2: ~/Documents/GitHub/Assignment3_OS/Sectio
avi@avi-1-2: ~/Documents/GitHub/Assignment3_OS/Section3
avi@avi-1-2: ~/Documents/GitHub/Assignment3_OS/Sectio

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ cd Section3
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ export LD_LIBRARY_PATH=$LD_LIBRARY_
PATH:.
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ ./proactor_server
Client connected
Client connected
Client connected
Client 0 says: Hi everyone
Client 1 says: Hello friends
Client 2 says: Hello World
```

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ cd Section3/
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ ./client
Hi everyone
Server says: Client 1: Hello friends
Server says: Client 2: Hello World
```

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ cd Section3
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ ./client
Server says: Client 0: Hi everyone
Hello friends
Server says: Client 2: Hello World
```

```
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS$ cd Section3
avi@avi-1-2:~/Documents/GitHub/Assignment3_OS/Section3$ ./client
Server says: Client 0: Hi everyone
Server says: Client 1: Hello friends
Hello World
```