

Задание 1

Необходимые знания

1. TCP и TCP/IP
2. TCP vs UDP
3. Системный вызов `socket`
4. Системный вызов `bind`
5. Системный вызов `listen`
6. Системный вызов `accept`
7. Системный вызов `recv`
8. Системный вызов `send`
9. Системный вызов `close`
10. Системный вызов `connect`

В предыдущей лабораторной работе вы распараллеливали вычисление факториала по модулю с помощью потоков. В этой работе вы пойдете еще дальше: вы распараллелите эту работу еще и между серверами.

Необходимо закончить `client.c` и `server.c`:

Клиент в качестве аргументов командной строки получает `k`, `mod`, `servers`, где `k` это факториал, который необходимо вычислить ($k! \% \text{mod}$), `servers` это путь до файла, который содержит сервера (`ip:port`), между которыми клиент будет распараллеливать соединения.

Сервер получает от клиента "кусоч" своих вычислений и `mod`, в ответ отправляет клиенту результат этих вычислений.

```
ubuntu@ubuntu: ~/osis/lab6/src
ubuntu@ubuntu:~/osis/lab6/src$ ./client --k 1000 --mod 5 --servers servers.txt
Final result: 0
ubuntu@ubuntu:~/osis/lab6/src$ ./client --k 5 --mod 7 --servers servers.txt
Final result: 1
ubuntu@ubuntu:~/osis/lab6/src$

ubuntu@ubuntu:~/osis/lab6/src
ubuntu@ubuntu:~/osis/lab6/src$ gcc -pthread -o server server.c multmodulo.c
ubuntu@ubuntu:~/osis/lab6/src$ gcc -pthread -o client client.c multmodulo.c
ubuntu@ubuntu:~/osis/lab6/src$ ./server --port 20001 --tnum 4

ubuntu@ubuntu:~/osis/lab6/src
ubuntu@ubuntu:~/osis/lab6/src$ ./server --port 20002 --tnum 4
```

Задание 2

Создать makefile для программ клиента и сервера

Задание 3

Найти дублирующийся код в двух приложениях и вынести его в библиотеку. Добавить изменения в makefile.

Эти задания я объединил

```
M makefile
1  CC = gcc
2  CFLAGS = -pthread
3  SERVER = server
4  CLIENT = client
5
6  K ?= 1000
7  MOD ?= 5
8  SERVERS_FILE ?= servers.txt
9
10 all: $(SERVER) $(CLIENT)
11
12 $(SERVER): server.o multmodulo.o
13 |     $(CC) $(CFLAGS) -o $@ $^
14
15 $(CLIENT): client.o multmodulo.o
16 |     $(CC) -o $@ $^
17
18 %.o: %.c multmodulo.h
19 |     $(CC) -c $<
20
21 clean:
22 |     rm -f $(SERVER) $(CLIENT) *.o
23
24 run-server1:
25 |     ./server --port 20001 --tnum 4
26
27 run-server2:
28 |     ./server --port 20002 --tnum 4
29
30 run-client:
31 |     ./client --k $(K) --mod $(MOD) --servers $(SERVERS_FILE)
32
33 .PHONY: all clean run-server1 run-server2 run-client
```

```
ubuntu@ubuntu: ~/osis/lab6/src
ubuntu@ubuntu:~/osis/lab6/src$ make clean
rm -f server client *.o
ubuntu@ubuntu:~/osis/lab6/src$ make
gcc -c server.c
gcc -c multmodulo.c
gcc -pthread -o server server.o multmodulo.o
gcc -c client.c
gcc -o client client.o multmodulo.o
ubuntu@ubuntu:~/osis/lab6/src$ make run-server1
./server --port 20001 --tnum 4
[ ]

ubuntu@ubuntu:~/osis/lab6/src$ make run-server2
./server --port 20002 --tnum 4
[ ]

ubuntu@ubuntu:~/osis/lab6/src$ make run-client
./client --k 1000 --mod 5 --servers servers.txt
Final result: 0
ubuntu@ubuntu:~/osis/lab6/src$ make run-client K=5 MOD=7 SERVERS_FILE=servers.txt
./client --k 5 --mod 7 --servers servers.txt
Final result: 1
ubuntu@ubuntu:~/osis/lab6/src$ make run-client K=6 MOD=10 SERVERS_FILE=servers.txt
./client --k 6 --mod 10 --servers servers.txt
Final result: 0
ubuntu@ubuntu:~/osis/lab6/src$
```