

# Exercício 6

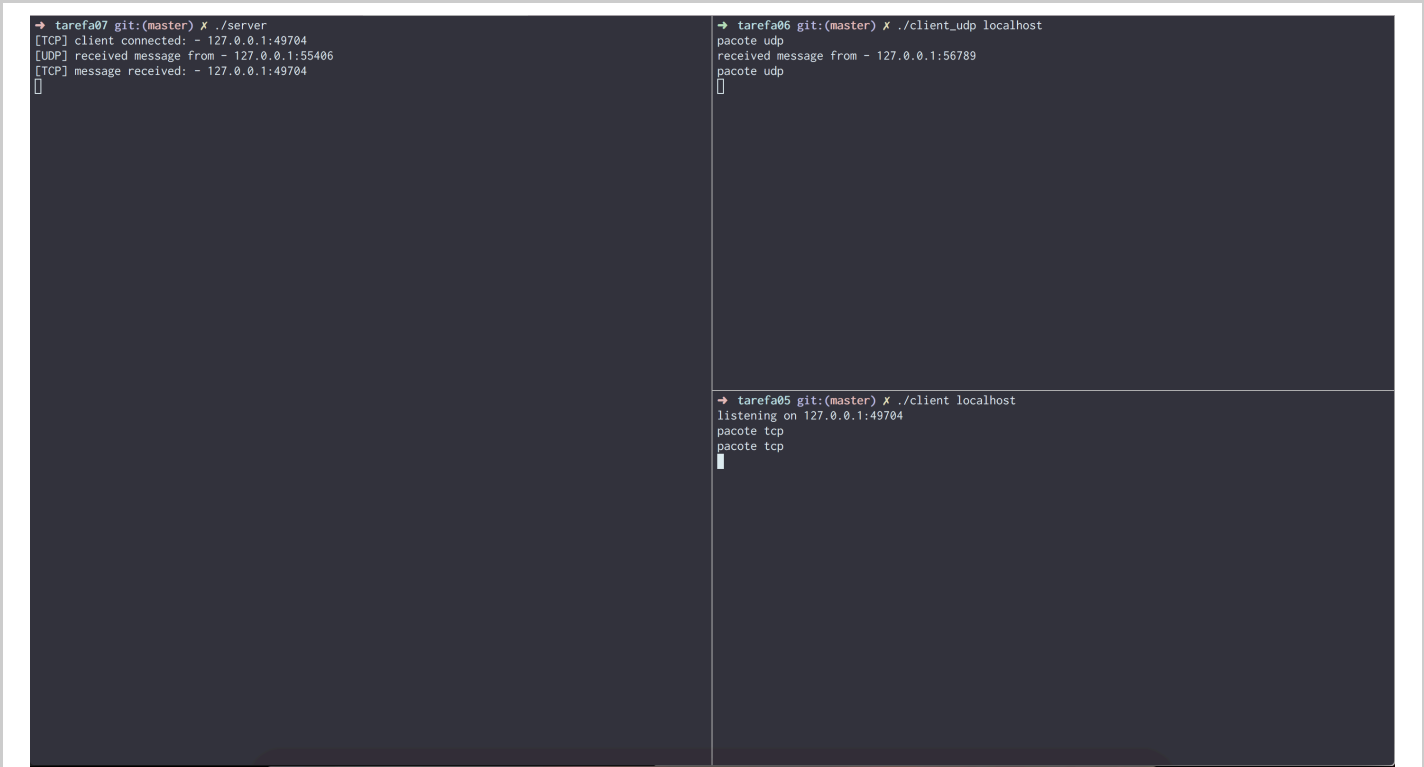
NOME	RA
Renan Camargo de Castro	147775

## Q1

- `int setsockopt(int socket, int level, int option_name, const void *option_value, socklen_t option_len)`: Essa função seta a opção especificada por `option_name` para o valor especificado em `value` no socket especificado pelo fd `socket` e no nível de protocolo especificado por `level`. **Retorna 0, caso sucesso e -1 caso erro.**

## Q2

Foi implementado o servidor, multiplexando os sockets de entrada com Select. Comprova-se o funcionamento com mensagens de log:



Screenshot verificando funcionamento de server TCP/UDP com cliente anterior

## Q3

Também é possível verificar utilizando TCPDUMP e netcat como cliente para comprovar os pacotes recebidos/enviados. Na screenshot a seguir foi utilizado ambos e fica claro os pacotes TCP e UDP. Em vermelho, temos os pacotes TCP e em azul, pacotes UDP.

```
parallels@ubuntu: ~/Desktop/tarefa07
parallels@ubuntu:~/Desktop/tarefa07$ ./server
[TCP] client connected: - 127.0.0.1:42623
[UDP] received message from - 127.0.0.1:46619
[UDP] received message from - 127.0.0.1:46619
[TCP] message received: - 127.0.0.1:42623

parallels@ubuntu:~/Desktop/tarefa07$ netcat -u 127.0.0.1 56789
este
estet3
estet3

parallels@ubuntu:~/Desktop/tarefa07$ ./client localhost tcp
listening on 127.0.0.1:42623
tcp
tcp

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes
18:50:20.692082 IP localhost.42623 > localhost.56789: Flags [S], seq 3088179303,
  win 43690, options [mss 65495,sackOK,TS val 224890 ecr 0,nop,wscale 7], length 0
18:50:20.692097 IP localhost.56789 > localhost.42623: Flags [S.], seq 1120949292,
  ack 3088179304, win 43690, options [mss 65495,sackOK,TS val 224890 ecr 224890,
  nop,wscale 7], length 0
18:50:20.692143 IP localhost.42623 > localhost.56789: Flags [.], ack 1, win 342,
  options [nop,nop,TS val 224890 ecr 224890], length 0
18:50:42.234004 IP localhost.46619 > localhost.56789: UDP, length 6
18:50:42.234126 IP localhost.56789 > localhost.46619: UDP, length 6
18:50:52.394805 IP localhost.46619 > localhost.56789: UDP, length 8
18:50:52.394924 IP localhost.56789 > localhost.46619: UDP, length 8
18:51:00.768186 IP localhost.42623 > localhost.56789: Flags [P.], seq 1:6, ack 1,
  win 342, options [nop,nop,TS val 234909 ecr 224890], length 5
18:51:00.768300 IP localhost.56789 > localhost.42623: Flags [.], ack 6, win 342,
  options [nop,nop,TS val 234909 ecr 234909], length 0
18:51:00.768384 IP localhost.56789 > localhost.42623: Flags [P.], seq 1:6, ack 6,
  win 342, options [nop,nop,TS val 234909 ecr 234909], length 5
18:51:00.768456 IP localhost.42623 > localhost.56789: Flags [.], ack 6, win 342,
  options [nop,nop,TS val 234909 ecr 234909], length 0
```

Screenshot verificando funcionamento de server TCP/UDP com netcat udp e tcpdump