

Laboratório 6

4) a função `Accept` retorna um novo file descriptor referente ao socket que conecta-se à um cliente que requisitou conexão com o servidor. Assim, as chamadas de `Close` encerram apenas o socket referente a este cliente. O servidor continua escutando no socket `listenfd`. A conexão de um cliente não afeta a conexão dos outros, logo, o encerramento de um cliente não interfere na conexão dos demais.

5) podemos utilizar o comando `ps ax -f | grep servidor` para capturar todos os processos referentes ao executável *servidor*, que é o nosso servidor:

```
rafael      5279   4409    0 15:53 pts/11    Sl+      0:26 vim servidor.c
rafael      7909   5338    0 18:06 pts/23    S+       0:00 ./servidor 1400
rafael      8019   7909    0 18:07 pts/23    S+       0:00 ./servidor 1400
rafael      8078   7909    0 18:07 pts/23    S+       0:00 ./servidor 1400
rafael      8139   7909    0 18:08 pts/23    S+       0:00 ./servidor 1400
rafael      8253   8141    0 18:11 pts/27    S+       0:00 grep servidor
```

A segunda coluna é o process identifier (PID) do processo, enquanto a terceira é o PPID (parent process identifier). Notamos os processos 8019, 8078 e 8139 são todos filhos de 7909, que é o processo pai.

Como segunda opção podemos também utilizar `pstree -p 7909`, que retorna uma estrutura semelhante à:

```
servidor(7909)  -----servidor(8019)
                  |--servidor(8078)
                  |--servidor(8139)
```

comprovando a relação de pai-filho entre os processos.

6) Executando o servidor e um cliente e em seguida executando `netstat -p -t | grep 9999`, obtemos:

Proto	Recv-Q	Send-Q	Local Address	Foreign Address
State	PID/Program name			
tcp	0	0	localhost:36040	localhost:9999
ESTABLISHED	8953/cliente			
tcp	0	0	localhost:9999	localhost:36040
ESTABLISHED	8954/servidor			

Em seguida, terminamos ambos servidor e cliente, e reexecutamos `netstat -p -t | grep 9999`:

tcp	0	0	localhost:36040	localhost:9999
TIME_WAIT	-			

Observamos que a conexão que fica no estado `TIME_WAIT` após o encerramento da conexão é a do cliente. Isso condiz com a implementação realizada, pois é o lado que realiza o término da conexão ativamente, ao digitar uma das keywords de saída *exit*, *bye*, *sair* ou *quit* ou pressionar `Ctrl+D`, que simula um *end of file*, resultando na primeira chamada de `close()` do protocolo TCP.

Os testes que realizamos para verificar se o cliente estava executando a saída do comando corretamente e saindo com certas strings (sair, bye, quit), se o servidor imprimia o IP, PORTA e string enviadas pelo cliente, assim como os momentos de conexão e desconexão foram:

```
rafael@BlackHawk: ~/unicamp/sshic/lab6.2
21:27:~/unicamp/sshic/lab6.2$ ./servidor 1400
Listening on port 1400...
2015-11-09 21:28:01>177.220.84.244:40210 connected.
177.220.84.244:402105 echo $PPID
2015-11-09 21:28:33>127.0.0.1:43407 connected.
127.0.0.1:434075 echo $PPID
2015-11-09 21:28:46>177.220.84.244:40210 disconnected.
2015-11-09 21:29:33>177.220.84.244:40212 connected.
177.220.84.244:402125 head -n 2 cliente.c
2015-11-09 21:32:00>177.220.84.244:40212 disconnected.
127.0.0.1:434075 echo $PPID
2015-11-09 21:32:32>127.0.0.1:43407 disconnected.
2015-11-09 21:34:24>127.0.0.1:43423 connected.
127.0.0.1:434235 ls
2015-11-09 21:34:40>127.0.0.1:43423 disconnected.
^C
21:34:~/unicamp/sshic/lab6.2$

rafael@BlackHawk: ~/unicamp/sshic/lab6.2
21:24:~/unicamp/sshic/lab6.2$ ./cliente 177.220.84.244 1400
Connected to server: 177.220.84.244:40210
echo $PPID
local input:echo $PPID
server answer:
7349
21:28:~/unicamp/sshic/lab6.2$ ./cliente 127.0.0.1 1400
Connected to server: 127.0.0.1:43423
ls
local input:ls
server answer:
act.sh
cliente
cliente.c
errorHandling.c
log.txt
Makefile
rai06228.odt
screenshot.png
servidor
servidor.c
bye
local input:bye
Encerrando conexão com o servidor...
21:34:~/unicamp/sshic/lab6.2$

rafael@BlackHawk: ~/unicamp/sshic/lab6.2
21:25:~/unicamp/sshic/lab6.2$ ./cliente 127.0.0.1 1400
Connected to server: 127.0.0.1:43407
echo $PPID
local input:echo $PPID
server answer:
7353
echo $PPID
local input:echo $PPID
server answer:
7353
sair
local input:sair
Encerrando conexão com o servidor...
21:32:~/unicamp/sshic/lab6.2$

rafael@BlackHawk: ~/unicamp/sshic/lab6.2
21:29:~/unicamp/sshic/lab6.2$ ./cliente 177.220.84.244 1400
Connected to server: 177.220.84.244:40212
head -n 2 cliente.c
local input:head -n 2 cliente.c
server answer:
#include <sys/socket.h>
#include <sys/types.h>
exit
local input:exit
Encerrando conexão com o servidor...
21:32:~/unicamp/sshic/lab6.2$
```