OPERATING SYSTEMS: SESSION5

JANUARY 2023

TecnoCampus Escuela Superior Politécnica

Centro adscrito a la upf. Universitat Pompeu Fabra Barcelona

TecnoCampus 10 años

C FUNCTIONS FOR SOCKETS: SERVER

- On the SERVER SIDE we have to:
  - Have a SOCKET-FILE DESCRIPTOR (**SOCKET**)
  - Prepare the LINK between the INTERNET CONNECTION and THE SERVER PROCESS (**BIND**)
  - Wait for a CONNECTION REQUEST (**ACCEPT**)
  - Close the CONNECTION (**CLOSE**)

C FUNCTIONS FOR SOCKETS: SERVER-SOCKET-FILE DESCRIPTOR

```
int serversock, clientsock;
/* Create TCP socket */
serversock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
if (serversock < 0) {
    err_sys("Error socket");
}
```

C FUNCTIONS FOR SOCKETS: SERVER-SOCKET-BINDING

- We need to BIND the socket to SERVER PROCESS:
  - Inform the OS that when a PACKET has DESTINATION IP (SERVER IP) AND DESTINATION PORT (SERVER PORT) it should be linked with the OPENED SOCKET
  - So the SERVER process will be able to manage the information

  - THAT'S THE BINDING PROCESS

C FUNCTIONS FOR SOCKETS: SERVER-SOCKET-BINDING

- We need a SOCKET DESCRIPTOR and LOCAL ADDRESS (SERVER) CONFIGURATION

```
/* Set information for sockaddr_in structure */
memset(&echoserver, 0, sizeof(echoserver));       /* we reset memory */
echoserver.sin_family = AF_INET;                  /* Internet/IP */
echoserver.sin_addr.s_addr = htonl(INADDR_ANY);   /* ANY address */
echoserver.sin_port = htons(atoi(argv[1]));       /* server port */
```

```
/* Bind socket */
result = bind(serversock, (struct sockaddr *) &echoserver, sizeof(echoserver));
if (result < 0) {
    err_sys("Error bind");
}
```

C FUNCTIONS FOR SOCKETS: SERVER-SOCKET-BINDING

- PORT must be UNIQUE

```
echoserver.sin_port = htons(atoi(argv[1]));        /* server port */
```

- ADDRESS could be:
  - ANY ADDRESS/INTERFACE from SERVER DEVICE

```
echoserver.sin_addr.s_addr = htonl(INADDR_ANY);    /* ANY address */
```

  - THE UNIQUELY CONFIGURED ADDRESS/INTERFACE from SERVER

```
echoserver.sin_addr.s_addr = inet_addr(argv[1]);   /* IP address */
```

C FUNCTIONS FOR SOCKETS: SERVER-SOCKET-BINDING

- If you have more than ONE IP
  - You can “accept” connection requests from more than one IP (interface)
  - Or just only one

C FUNCTIONS FOR SOCKETS: SERVER-ACCEPT CONNECTION REQUEST

- The SERVER is ready to wait for a CONNECTION REQUEST

- If the SERVER ACCEPT the connection the SOCKET will be ALREADY CONFIGURED
  - To have the ability to manage more than one connection at the same IP+PORT …
  - The OS will create a **NEW SOCKET DESCRIPTOR** with THE 5-TUPLE PARAMETERS to be used on the current CONNECTION
  - The former SOCKET could be used to ACCEPT a second CONNECTION REQUEST

C FUNCTIONS FOR SOCKETS: SERVER-ACCEPT CONNECTION REQUEST

```
/* Wait for a connection from a client */
clientsock = accept(serversock, (struct sockaddr *) &echoclient, &clientlen);
if (clientsock < 0) {
    err_sys("Error accept");
}
```

WHEN	SOCKET DESCRIPTOR	PROTOCOL	LOCAL-IP	LOCAL-PORT	REMOTE-IP	REMOTE-PORT
BEFORE ACCEPT	serversock	TCP	SERVER-IP	SERVER-PORT	NOT SET	NOT SET
	clientsock	TCP	SERVER-IP	SERVER-PORT	NOT SET	NOT SET
AFTER ACCEPT	serversock	TCP	SERVER-IP	SERVER-PORT	NOT SET	NOT SET
	clientsock	TCP	SERVER-IP	SERVER-PORT	CLIENT-IP	CLIENT-PORT

- You can ONLY use SOCKET DESCRIPTOR clientsock (THE NEW ONE) BECAUSE IT IS THE ONLY ONE WITH 5-TUPLE PARAMETERS

C FUNCTIONS FOR SOCKETS: SERVER-LISTEN

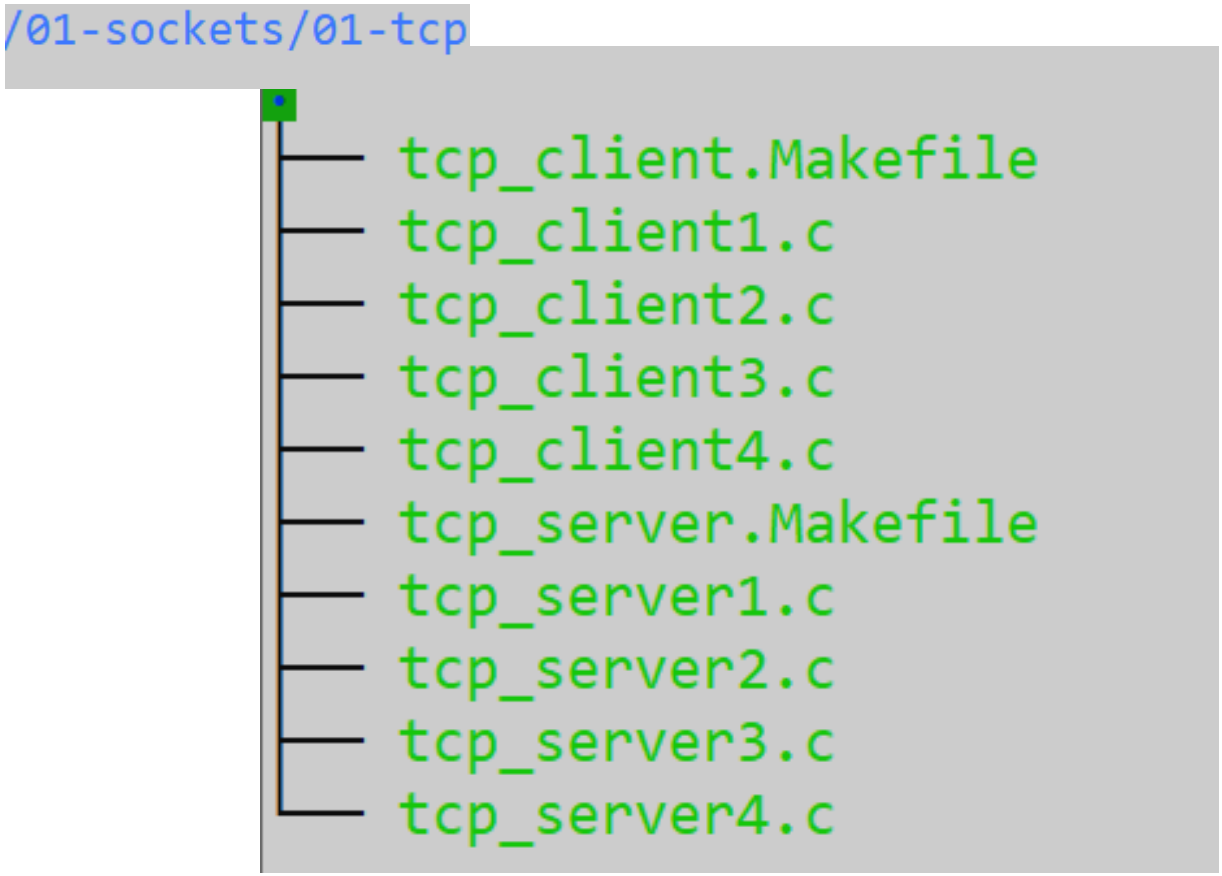
- What could happen if you get a second REQUEST before the first one is ACCEPTED?
  - Do you have a kind of QUEUE: CONNECTION REQUEST in queue…. Waiting to be processed and managed
  - OS will manage the QUEUE

```
/* Listen socket */
result = listen(serversock, MAXPENDING);
if (result < 0) {
    err_sys("Error listen");
}
```

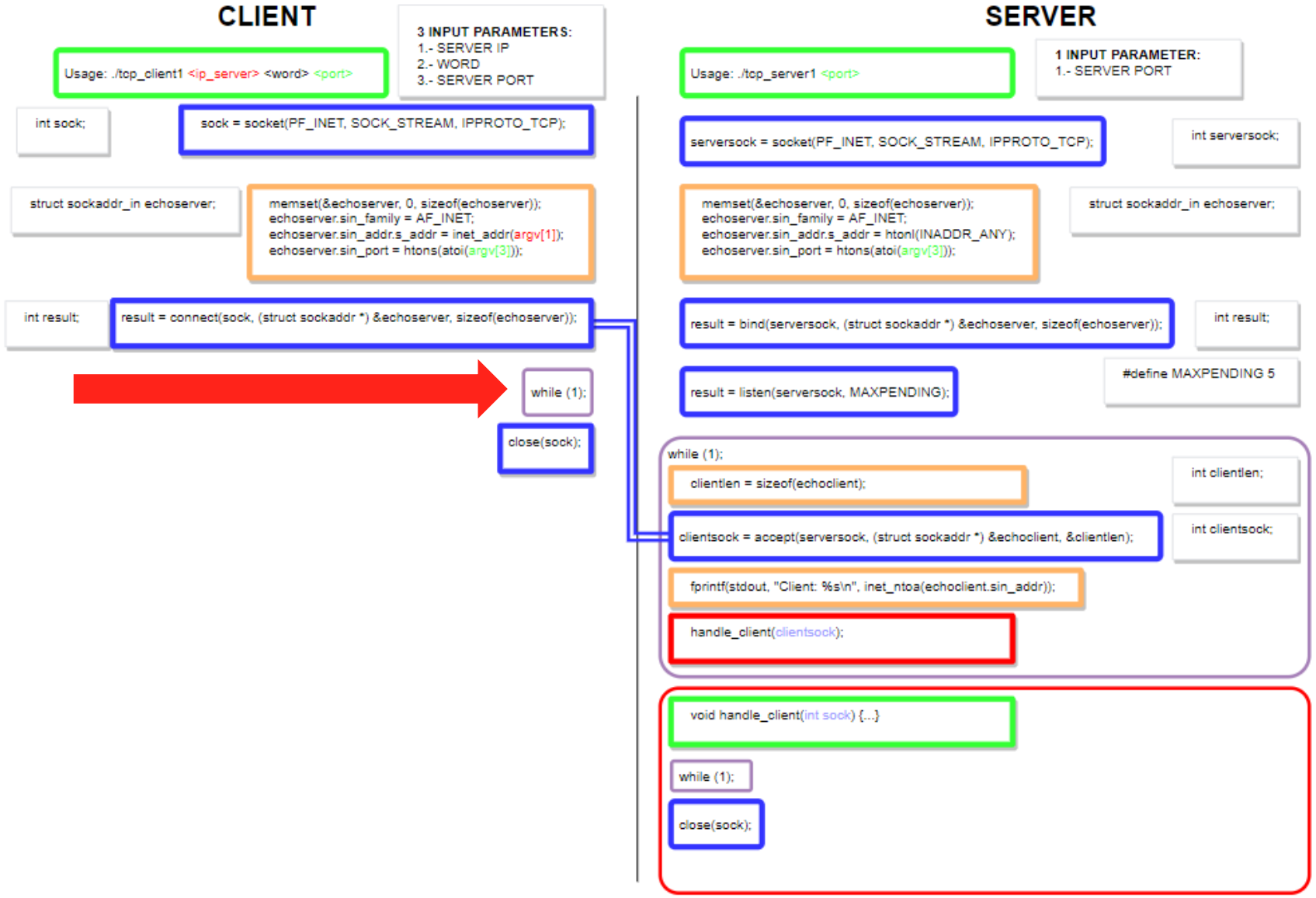
```
LISTEN(2)                                Linux Programmer's Manual

NAME
       listen - listen for connections on a socket
```

C FUNCTIONS FOR SOCKETS: EXAMPLES



C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)



C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- CLIENT:
  - 1.- REQUEST a connection to server
  - 2.- LOOP (doing NOTHING)

- SERVER:
  - 1.- BIND socket
  - 2.- LOOP (ready to accept a client request)
    - 2A.- ACCEPT connection request from client
    - 2B.- PRINF client information
    - 2C.- HANDLE CLIENT CONNECTION
      - 2C-1.- LOOP (doing NOTHING)

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- OBJECTIVE:
  - CLIENT-SERVER can be connected
  - CLIENT-SERVER do not share information

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- **nmap**:
  - CONNECTIONS MONITORING
- Execute SERVER (but not CLIENT):

```
LJG:./tcp_server1 6000 &
[1] 114
LJG:
```

- CHECK for PROCESS (background):

```
LJG:ps 114
  PID TTY      STAT   TIME COMMAND
  114 tty1     S      0:00 ./tcp_server1 6000
LJG:
```

- INSTALL **nmap**:

```
sudo apt install nmap
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- nmap:
  - CONNECTIONS MONITORING
- nmap –sT 127.0.0.1 –p 6000: SCANNING PORT 6000 LOCALHOST TCP

ON WSL1

```
LJG:nmap -sT 127.0.0.1 -p 6000
Warning: Nmap may not work correctly on Windows Subsystem for Linux.
```

- DEVASC:

```
SERVER(01-tcp):./tcp_server1 6000 &
[1] 2759
SERVER(01-tcp):
```

```
SERVER(01-tcp):ps -aux | grep tcp_server
devasc      2759  0.0  0.0   2356   516 pts/0    S    20:19   0:00 ./tcp_server1 6000
```

```
SERVER(01-tcp):nmap -sT 127.0.0.1

Command 'nmap' not found, but can be installed with:

sudo snap install nmap  # version 7.91, or
sudo apt  install nmap  # version 7.80+dfsg1-2build1

See 'snap info nmap' for additional versions.

SERVER(01-tcp):
```

```
SO: nmap -sT 127.0.0.1 -p 1000
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-15 14:12 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00031s latency).

PORT     STATE SERVICE
1000/tcp open  cadlock

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
SO:
```

NO PROBLEMS ON WSL2

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- nmap:
  - CONNECTIONS MONITORING
- nmap –sT 127.0.0.1 –p 6000: SCANNING PORT 6000 LOCALHOST TCP

```
SERVER(01-tcp):nmap -sT 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-22 20:27 UTC
Client: 127.0.0.1
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0014s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
631/tcp  open  ipp
6000/tcp open  X11
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 0.74 seconds
SERVER(01-tcp):
```

- PORT 6000: X11?

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- nmap: (again now on port 16000)

```
SERVER(01-tcp):ps -aux | grep tcp_server | grep 16000
devasc      3160  0.0  0.0   2356   576 pts/0    S    20:29   0:00 ./tcp_server1 16000
SERVER(01-tcp):nmap -sT 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-22 20:29 UTC
Client: 127.0.0.1
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0012s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
631/tcp   open  ipp
6000/tcp  open  X11
8081/tcp  open  blackice-icecap
16000/tcp open  fmsas

Nmap done: 1 IP address (1 host up) scanned in 1.04 seconds
SERVER(01-tcp):
```

- PORT 16000: fmsas?

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- nmap: (again now on port 45321)

```
SERVER(01-tcp):./tcp_server1 45321 &
[3] 3170
```

```
SERVER(01-tcp):ps -aux | grep tcp_server | grep 45321
devasc      3170  0.0  0.0   2356   584 pts/0    S    20:30   0:00 ./tcp_server1 45321
```

```
SERVER(01-tcp):nmap -sT 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-22 20:31 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00057s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
631/tcp   open  ipp
6000/tcp  open  X11
8081/tcp  open  blackice-icecap
16000/tcp open  fmsas

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
```

```
SERVER(01-tcp):nmap -sT 127.0.0.1 -p 45321
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-22 20:31 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00014s latency).

PORT      STATE SERVICE
45321/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
Client: 127.0.0.1
SERVER(01-tcp):
```

- PORT 45321: unknown

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- **Netstat** (PORT CONNECTION MONITORING)

```
SERVER(01-tcp):netstat -tulpn | grep tcp_server1
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        2      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      2759/./tcp_server1
tcp        1      0 0.0.0.0:16000           0.0.0.0:*               LISTEN      3160/./tcp_server1
tcp        0      0 0.0.0.0:45321           0.0.0.0:*               LISTEN      3170/./tcp_server1
SERVER(01-tcp):
```

- BIND 1: LOCAL (0.0.0.0:6000) REMOTE (0.0.0.0:*) LISTEN PID=2759 ./tcp_server1
- BIND 2: LOCAL (0.0.0.0:16000) REMOTE (0.0.0.0:*) LISTEN PID=3160 ./tcp_server1
- BIND 3: LOCAL (0.0.0.0:45321) REMOTE (0.0.0.0:*) LISTEN PID=3170 ./tcp_server1

- WE HAVE 3 SERVERS READY TO ACCEPT CONNECTION REQUESTS…

```
SO: netstat -putona
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name     Timer
tcp        0      0 0.0.0.0:1000            0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
SO:
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- CONNECTION REQUEST (CLIENT)

```
CLIENT(01-tcp):./tcp_client1 127.0.0.1 hello 6000 &
[1] 3226
CLIENT(01-tcp):
```

```
SERVER(01-tcp):ps -aux | grep tcp
root        1559  8.4  2.9 699804 118048 tty7    Ssl+ 19:05   8:37 /usr/lib/xorg/Xorg -core :0 -seat seat0 -a
uth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
devasc      2759 47.7  0.0   2488   516 pts/0    R    20:19  13:00 ./tcp_server1 6000
devasc      3160 61.0  0.0   2488   576 pts/0    R    20:29  10:27 ./tcp_server1 16000
devasc      3170 57.2  0.0   2488   584 pts/0    R    20:30   9:03 ./tcp_server1 45321
devasc      3226 46.7  0.0   2356   516 pts/1    R    20:43   1:19 ./tcp_client1 127.0.0.1 hello 6000
devasc      3235  0.0  0.0   9032   660 pts/0    S+   20:46   0:00 grep --color=auto tcp
SERVER(01-tcp):
```

```
SERVER(01-tcp):netstat -tulpn | grep tcp_server1
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        3      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      2759/./tcp_server1
tcp        1      0 0.0.0.0:16000           0.0.0.0:*               LISTEN      3160/./tcp_server1
tcp        0      0 0.0.0.0:45321           0.0.0.0:*               LISTEN      3170/./tcp_server1
SERVER(01-tcp):
```

- WHERE IS THE CONNECTION (ESTABLISHED CONNECTION?)

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- CONNECTION REQUEST (CLIENT)

```
SERVER(01-tcp):ss -4 state established
Netid      Recv-Q      Send-Q           Local Address:Port           Peer Address:Port      Process
tcp        0           0                    127.0.0.1:x11               127.0.0.1:39846
tcp        0           0                    127.0.0.1:39846             127.0.0.1:x11
SERVER(01-tcp):
```

```
SERVER(01-tcp):ss -4 -n state established
Netid      Recv-Q      Send-Q           Local Address:Port           Peer Address:Port      Process
tcp        0           0                    127.0.0.1:6000              127.0.0.1:39846
tcp        0           0                    127.0.0.1:39846             127.0.0.1:6000
SERVER(01-tcp):
```

- THE CONNECTION IS:
  - 127.0.0.1:39846 (CLIENT) – 127.0.0.1:6000 (SERVER)

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- KILL PROCESSES: you can kill ALL “YOUR PROCESSES”: **KILL -9 -1**

```
SERVER(01-tcp):ps -aux | grep tcp
root        1559  8.2  2.9 699804 118048 tty7    Ssl+ 19:05   8:56 /usr/lib/xorg/Xorg -core :0 -seat seat0 -a
uth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
devasc      2759 47.7  0.0   2488   516 pts/0    R    20:19  16:31 ./tcp_server1 6000
devasc      3160 57.0  0.0   2488   576 pts/0    R    20:29  13:59 ./tcp_server1 16000
devasc      3170 54.1  0.0   2488   584 pts/0    R    20:30  12:33 ./tcp_server1 45321
devasc      3226 47.5  0.0   2356   516 pts/1    R    20:43   4:51 ./tcp_client1 127.0.0.1 hello 6000
devasc      3270  0.0  0.0   9032   660 pts/0    S+   20:53   0:00 grep --color=auto tcp
SERVER(01-tcp):SERVER(01-tcp):kill -9 2759
SERVER(01-tcp):
[1]   Killed                  ./tcp_server1 6000
SERVER(01-tcp):kill -9 3160
SERVER(01-tcp):kill -9 3170
[2]-  Killed                  ./tcp_server1 16000
SERVER(01-tcp):kill -9 3226
[3]+  Killed                  ./tcp_server1 45321
SERVER(01-tcp):SERVER(01-tcp):ps -aux | grep tcp
root        1559  8.2  2.9 699804 118048 tty7    Rsl+ 19:05   9:07 /usr/lib/xorg/Xorg -core :0 -seat seat0 -a
uth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
devasc      3274  0.0  0.0   9032   732 pts/0    R+   20:56   0:00 grep --color=auto tcp
SERVER(01-tcp):ss -4 -n state established
Netid      Recv-Q      Send-Q            Local Address:Port            Peer Address:Port      Process
SERVER(01-tcp):netstat -tulpn | grep tcp_server1
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.) SERVER(01-tcp):nmap -sT 127.0.0.1 -p 6000,16000,45321
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-22 20:57 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00034s latency).

PORT      STATE  SERVICE
6000/tcp  closed X11
16000/tcp closed fmsas
45321/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.56 seconds
SERVER(01-tcp):
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- IP ADDRESS?

```
SERVER(01-tcp):ifconfig
dummy0: flags=195<UP,BROADCAST,RUNNING,NOARP>  mtu 1500
        inet 192.0.2.1  netmask 255.255.255.255  broadcast 0.0.0.0
        inet6 fe80::74a7:38ff:fe2d:33a1  prefixlen 64  scopeid 0x20<link>
        ether 76:a7:38:2d:33:a1  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 770 (770.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe90:deaf  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:90:de:af  txqueuelen 1000  (Ethernet)
        RX packets 74721  bytes 108821846 (108.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14496  bytes 896267 (896.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.5  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe2f:a32b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:2f:a3:2b  txqueuelen 1000  (Ethernet)
        RX packets 709  bytes 76502 (76.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 104  bytes 16101 (16.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6150  bytes 311834 (311.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6150  bytes 311834 (311.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

SERVER(01-tcp):
```

```
SERVER(01-tcp):ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:90:de:af brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 79346sec preferred_lft 79346sec
    inet6 fe80::a00:27ff:fe90:deaf/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:2f:a3:2b brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.5/24 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 433sec preferred_lft 433sec
    inet6 fe80::a00:27ff:fe2f:a32b/64 scope link
       valid_lft forever preferred_lft forever
4: dummy0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether 76:a7:38:2d:33:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/32 scope global dummy0
       valid_lft forever preferred_lft forever
    inet 192.0.2.2/32 scope global dummy0
       valid_lft forever preferred_lft forever
    inet 192.0.2.3/32 scope global dummy0
       valid_lft forever preferred_lft forever
    inet 192.0.2.4/32 scope global dummy0
       valid_lft forever preferred_lft forever
    inet 192.0.2.5/32 scope global dummy0
       valid_lft forever preferred_lft forever
    inet6 fe80::74a7:38ff:fe2d:33a1/64 scope link
       valid_lft forever preferred_lft forever
SERVER(01-tcp):
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- CHECKING IP ADDRESS?

```
SERVER(01-tcp):ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.103 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.103/0.103/0.103/0.000 ms
SERVER(01-tcp):ping -c 2 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.107 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.083 ms

--- 10.0.2.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.083/0.095/0.107/0.012 ms
SERVER(01-tcp):ping -c 3 192.168.56.5
PING 192.168.56.5 (192.168.56.5) 56(84) bytes of data.
64 bytes from 192.168.56.5: icmp_seq=1 ttl=64 time=0.119 ms
64 bytes from 192.168.56.5: icmp_seq=2 ttl=64 time=0.077 ms
64 bytes from 192.168.56.5: icmp_seq=3 ttl=64 time=0.217 ms

--- 192.168.56.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.077/0.137/0.217/0.058 ms
```

```
SERVER(01-tcp):ping -c 1 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
64 bytes from 192.0.2.1: icmp_seq=1 ttl=64 time=0.081 ms

--- 192.0.2.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.081/0.081/0.081/0.000 ms
SERVER(01-tcp):ping -c 1 192.0.2.2
PING 192.0.2.2 (192.0.2.2) 56(84) bytes of data.
64 bytes from 192.0.2.2: icmp_seq=1 ttl=64 time=0.223 ms

--- 192.0.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.223/0.223/0.223/0.000 ms
SERVER(01-tcp):ping -c 1 192.0.2.3
PING 192.0.2.3 (192.0.2.3) 56(84) bytes of data.
64 bytes from 192.0.2.3: icmp_seq=1 ttl=64 time=0.083 ms

--- 192.0.2.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.083/0.083/0.083/0.000 ms
SERVER(01-tcp):ping -c 1 192.0.2.4
PING 192.0.2.4 (192.0.2.4) 56(84) bytes of data.
64 bytes from 192.0.2.4: icmp_seq=1 ttl=64 time=0.084 ms

--- 192.0.2.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.084/0.084/0.084/0.000 ms
SERVER(01-tcp):ping -c 1 192.0.2.5
PING 192.0.2.5 (192.0.2.5) 56(84) bytes of data.
64 bytes from 192.0.2.5: icmp_seq=1 ttl=64 time=0.084 ms

--- 192.0.2.5 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.084/0.084/0.084/0.000 ms
SERVER(01-tcp):
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- WORKING WITH IP DIFFERENT FROM LOCALHOST (127.0.0.1):

```
SERVER(01-tcp):./tcp_server1
Usage: ./tcp_server1 <port>
SERVER(01-tcp):./tcp_server1 6000 &
[1] 3305
SERVER(01-tcp):Client: 10.0.2.15

SERVER(01-tcp):
```

```
CLIENT(01-tcp):./tcp_client1 10.0.2.15 hello 6000 &
[1] 3306
CLIENT(01-tcp):
```

```
SERVER(01-tcp):ss -4 -n state established
Netid       Recv-Q      Send-Q           Local Address:Port            Peer Address:Port
tcp         0           0                    10.0.2.15:52738              10.0.2.15:6000
tcp         0           0                    10.0.2.15:6000               10.0.2.15:52738
SERVER(01-tcp):
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- CAN WE HAVE MORE THAN ONE CONNECTION WITH THE SERVER?:
  - IMPOSSIBLE!
  - WHY?
    - ONCE THE SERVER ACCEPT THE CONNECTION FROM THE CLIENT THE SERVER WILL EXECUTE **handle_client()** FUNCTION
    - **handle_client()** FUNCTION HAS A WHILE(1) LOOP. SO THE PROGRAM IS ALWAYS THERE. IT WILL NEVER TRY TO ACCEPT A NEW CONNECTION REQUEST

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- **NETCAT (NC)** CONNECTION TO THE SERVER

```
SERVER(01-tcp):./tcp_server1 7000 &
[1] 3344
SERVER(01-tcp):
SERVER(01-tcp):nc -vn 127.0.0.1 7000 &
[2] 3347
SERVER(01-tcp):Connection to 127.0.0.1 7000 port [tcp/*] succeeded!
Client: 127.0.0.1

SERVER(01-tcp):ss -4 -n state established
Netid      Recv-Q      Send-Q           Local Address:Port              Peer Address:Port
tcp        0           0                    127.0.0.1:56494                127.0.0.1:7000
tcp        0           0                    127.0.0.1:7000                 127.0.0.1:56494

[2]+  Stopped                 nc -vn 127.0.0.1 7000
SERVER(01-tcp):
```

```
SERVER(01-tcp):nc -vn 127.0.0.1 6000 &
[3] 3349
SERVER(01-tcp):nc: connect to 127.0.0.1 port 6000 (tcp) failed: Connection refused

[3]-  Exit 1                  nc -vn 127.0.0.1 6000
SERVER(01-tcp):
```

```
SERVER(01-tcp):ps
    PID TTY          TIME CMD
   2261 pts/0    00:00:00 bash
   3344 pts/0    00:01:47 tcp_server1
   3347 pts/0    00:00:00 nc
   3350 pts/0    00:00:00 ps
SERVER(01-tcp):
```

C FUNCTIONS FOR SOCKETS: EXAMPLE 1 (tcp_server1 – tcp_client1)

- **NETCAT (NC)** CONNECTION TO THE SERVER

```
SERVER(01-tcp):./tcp_server1 7000 &
[1] 3344
SERVER(01-tcp):
SERVER(01-tcp):nc -vn 127.0.0.1 7000 &
[2] 3347
SERVER(01-tcp):Connection to 127.0.0.1 7000 port [tcp/*] succeeded!
Client: 127.0.0.1

SERVER(01-tcp):ss -4 -n state established
Netid      Recv-Q      Send-Q           Local Address:Port              Peer Address:Port
tcp        0           0                    127.0.0.1:56494                127.0.0.1:7000
tcp        0           0                    127.0.0.1:7000                 127.0.0.1:56494

[2]+  Stopped                 nc -vn 127.0.0.1 7000
SERVER(01-tcp):
```

```
SERVER(01-tcp):nc -vn 127.0.0.1 6000 &
[3] 3349
SERVER(01-tcp):nc: connect to 127.0.0.1 port 6000 (tcp) failed: Connection refused

[3]-  Exit 1                  nc -vn 127.0.0.1 6000
SERVER(01-tcp):
```

```
SERVER(01-tcp):ps
    PID TTY          TIME CMD
   2261 pts/0    00:00:00 bash
   3344 pts/0    00:01:47 tcp_server1
   3347 pts/0    00:00:00 nc
   3350 pts/0    00:00:00 ps
SERVER(01-tcp):
```