Advanced Programming in the UNIX Environment

Week 09, Segment 4:
socket(PF_INET6, SOCK_STREAM, 0)

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

Terminal — 159×35

```
                                        rip = "unknown";
                                } else {
                                        printf("Client (%s) sent: \"%s\"", rip,
 buf);
                                }
                        }
                } while (rval != 0);
                (void)close(fd);
        }

        /* NOTREACHED */
}
:q
jschauma@apue$ ./send panix.netmeister.org 61174
jschauma@apue$ ./send panix.netmeister.org 61174
jschauma@apue$ ./send panix.netmeister.org 61174
connecting stream socket: Connection refused
jschauma@apue$ ./send panix.netmeister.org 61172
jschauma@apue$ ./send panix.netmeister.org 61172
connecting stream socket: Connection refused
jschauma@apue$ ./send panix.netmeister.org 61175
connecting stream socket: Connection refused
jschauma@apue$ ./send panix.netmeister.org 61165
jschauma@apue$ telnet panix.netmeister.org 61165
Trying 2001:470:30:84:e276:63ff:fe72:3900...
Connected to panix.netmeister.org.
Escape character is '^]'.
hello there!
How are you?
These messages all use the same connection.
^]
telnet> quit
Connection closed.
jschauma@apue$
```

0 bash

```
4:e276:63ff:fe72:3900.61165: Flags [.], ack 1, win 4140, options [nop,nop,TS va
l 1 ecr 1], length 0
19:45:22.286340 IP6 2601:87:4281:1e00:e12f:43b9:949b:1e0a.65508 > 2001:470:30:8
4:e276:63ff:fe72:3900.61165: Flags [P.], seq 1:15, ack 1, win 4140, options [no
p,nop,TS val 27 ecr 1], length 14
19:45:22.509455 IP6 2001:470:30:84:e276:63ff:fe72:3900.61165 > 2601:87:4281:1e0
0:e12f:43b9:949b:1e0a.65508: Flags [.], ack 15, win 4140, options [nop,nop,TS v
al 28 ecr 27], length 0
19:45:25.740418 IP6 2601:87:4281:1e00:e12f:43b9:949b:1e0a.65508 > 2001:470:30:8
4:e276:63ff:fe72:3900.61165: Flags [P.], seq 15:29, ack 1, win 4140, options [n
op,nop,TS val 34 ecr 28], length 14
19:45:25.965975 IP6 2001:470:30:84:e276:63ff:fe72:3900.61165 > 2601:87:4281:1e0
0:e12f:43b9:949b:1e0a.65508: Flags [.], ack 29, win 4140, options [nop,nop,TS v
al 34 ecr 34], length 0
19:45:35.078432 IP6 2601:87:4281:1e00:e12f:43b9:949b:1e0a.65508 > 2001:470:30:8
4:e276:63ff:fe72:3900.61165: Flags [P.], seq 29:74, ack 1, win 4140, options [n
op,nop,TS val 53 ecr 34], length 45
19:45:35.303842 IP6 2001:470:30:84:e276:63ff:fe72:3900.61165 > 2601:87:4281:1e0
0:e12f:43b9:949b:1e0a.65508: Flags [.], ack 74, win 4140, options [nop,nop,TS v
al 53 ecr 53], length 0
19:45:41.900355 IP6 2601:87:4281:1e00:e12f:43b9:949b:1e0a.65508 > 2001:470:30:8
4:e276:63ff:fe72:3900.61165: Flags [F.], seq 74, ack 1, win 4140, options [nop,
nop,TS val 66 ecr 53], length 0
19:45:41.931527 IP6 2001:470:30:84:e276:63ff:fe72:3900.61165 > 2601:87:4281:1e0
0:e12f:43b9:949b:1e0a.65508: Flags [.], ack 75, win 4140, options [nop,nop,TS v
al 66 ecr 66], length 0
19:45:41.933108 IP6 2001:470:30:84:e276:63ff:fe72:3900.61165 > 2601:87:4281:1e0
0:e12f:43b9:949b:1e0a.65508: Flags [F.], seq 1, ack 75, win 4140, options [nop,
nop,TS val 66 ecr 66], length 0
19:45:41.933153 IP6 2601:87:4281:1e00:e12f:43b9:949b:1e0a.65508 > 2001:470:30:8
4:e276:63ff:fe72:3900.61165: Flags [.], ack 2, win 4139, options [nop,nop,TS va
l 66 ecr 66], length 0
```

3 bash

Sockets: Streams in the Internet6 Domain

- connections are asymmetrical: one process requests a connection, the other process accepts the request
- one socket is created for each accepted request
- mark socket as willing to accept connections using `listen(2)`
- pending connections are then `accept(2)`ed
- `accept(2)` will block if no connections are available
- each connection requires a full handshake

Questions

- Update `streamwrite.c` to use `send(2)` instead of `write(2)`. What's the difference?
- Update both programs to handle dual-stack environment, *i.e.*, hosts with both an IPv4 and an IPv6 address.
- Run our stream reader, then connect to the open port multiple times simultaneously (e.g., using `telnet(1)` or `nc(1)`) - how does the reader handle this scenario?
- What happens if a client connects, sends a message, and disconnect while still in the backlog?
- What happens when more clients connect than you have marked as the BACKLOG in `listen(2)`? Keep a `tcpdump(8)` to observe the packets…