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/
Jan Schaumann  

CS631 - Advanced Programming in the UNIX Environment

```c

rip = "unknown";
} else {
  printf("Client (%s) sent: ",as", rip,

  buf);
}
}

while (rval != 0) {
  (void)close(fd);

  /* NOTREACHED */
}

jcschaumapue$ ./send panix.netmeister.org 61174
jcschaumapue$ ./send panix.netmeister.org 61174
jcschaumapue$ ./send panix.netmeister.org 61174
jcschaumapue$ ./send panix.netmeister.org 61174
jcschaumapue$ ./send panix.netmeister.org 61172
jcschaumapue$ ./send panix.netmeister.org 61172
jcschaumapue$ ./send panix.netmeister.org 61175
jcschaumapue$ ./send panix.netmeister.org 61165
jcschaumapue$ telnet panix.netmeister.org 61165
Trying 2001:47030:384:2e276:63ff:fe72:3900...
Connected to panix.netmeister.org.
Escape character is '\x03'.
Hello there!
How are you?
These messages all use the same connection.
}
telnet> quit
Connection closed.

jcschaumapue$
```

Jan Schaumann  

2020-10-30
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…