CS341#26 Client Review. TCP Server

Fun with python3 -m http.server (SimpleHTTPServer ), and netcat

1. **TCP Client (Review) + IPv6**

1.1 What are the steps to setting up a client TCP socket?

i. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ii. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ iii \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

struct addrinfo {

int ai\_flags

int ai\_family

int ai\_socktype

int ai\_protocol

socklen\_t ai\_addrlen

struct sockaddr \*ai\_addr

char \*ai\_canonname

struct addrinfo \*ai\_next

}

1.2 How many addrinfo structs does getaddrinfo return? Why?

1.3 How do I get a string error with getaddrinfo returns?

1.4 What is AF\_INET6?

1.5 What is 0:0:0:0:0:0:0:1?

1.6 Using getaddrinfo how do I ask for stream-based https IP4?

int startclient() {

struct addrinfo hints, \*result;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

hints.ai\_family = \_\_\_\_\_\_\_\_\_\_\_\_\_\_

hints.ai\_socktype = \_\_\_\_\_\_\_\_\_\_\_\_\_\_

int result = \_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_) ?

}

1.7 For each addrinfo what do you call next?

1.8 Can you bind() a client socket? Why would you want to?

**2. TCP SERVER**

2.1 What is a passive socket? How do you specify it?

2.2 Why would I create one?

2.3 If you don't bind what do you get?

2.4 What is htons? ntohs? Why/when do we need them?

struct sockaddr\_in stSockAddr;

int SocketFD = socket(PF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

memset(&stSockAddr, 0, sizeof(stSockAddr));

stSockAddr.sin\_family = AF\_INET;

stSockAddr.sin\_port = htons(1100);

stSockAddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

2.5 **Important!** What are the "*four calls*"? What is their order? And what is their purpose?

i\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ii \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ iii \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ iv \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <sys/types.h>

#include <sys/socket.h>

#include <netdb.h>

#include <unistd.h>

#include <arpa/inet.h>

//plus string.h, stdlib.h stdio.h

int main(int argc, char\*\* argv) { // TCP Server

int s;

int sock\_fd = **socket**(AF\_INET, SOCK\_STREAM, 0);

struct addrinfo hints, \*result;

memset(&hints, 0, sizeof(struct addrinfo));

hints.ai\_family = AF\_INET;

hints.ai\_socktype = SOCK\_STREAM;

hints.ai\_flags = AI\_PASSIVE;

s = **getaddrinfo**(NULL, "1234", &hints, &result);

if (s != 0) {

fprintf(stderr, "getaddrinfo: %s\n", gai\_strerror(s));

exit(1);

}

if ( **bind**(sock\_fd, result->ai\_addr, result->ai\_addrlen) != 0 ) {

perror("bind()"); exit(1);

}

if ( **listen**(sock\_fd, 10) != 0) {

perror("listen()"); exit(1);

}

struct sockaddr\_in \* result\_addr = **(struct sockaddr\_in\*)** result->ai\_addr;

printf("Listening on file descriptor %d, port %d\n", sock\_fd, **ntohs**(result\_addr->sin\_port));

printf("Waiting for connection...\n");

int **client\_fd** = **accept**(sock\_fd, NULL, NULL);

printf("Connection made: client\_fd=%d\n", client\_fd);

char buffer[1000];

int len = read(**client\_fd**, buffer, 999);

printf("Read %d chars\n", len);

if( len >0) {

buffer[len] = '\0';

printf("%s\n", buffer);

}

return 0;

}

Limitations: No SIGPIPE support. Single threaded. No port reuse.

If there's time...

What is a 'honey pot? What is epoll? What is select?