CS241 Lawrence Angrave L25 – TCP Server

struct addrinfo {

int ai\_flags;

int ai\_family;

int ai\_socktype;

int ai\_protocol;

socklen\_t ai\_addrlen;

struct sockaddr \*ai\_addr;

char \*ai\_canon name;

struct addrinfo \*ai\_next;

};

Client Review : What are the steps to setting up a client TCP socket?

1. 2. 3.

How many addrinfo structs does getaddrinfo return? Why?

How do I get a string error with getaddrinfo returns?

What is AF\_INET6?

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

Using getaddrinfo wow would I request stream-based protocol for https?

int startserver() {

struct addrinfo hints, \*result;

?

hints.ai\_family =

hints.ai\_socktype =

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

}

For each addrinfo what do you call next?

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

TCP SERVER

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

Why would I create one?

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

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);

What are the "four calls "? What is their order? And what is their purpose?

#include <sys/types.h>

#include <sys/socket.h>

#include <netdb.h>

#include <unistd.h>

#include <arpa/inet.h>

int main(int argc, char\*\* argv)

{

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));

//inet\_ntoa(result\_addr->sin\_addr),

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);

buffer[len] = '\0';

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

printf("===\n");

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

return 0;

}

(What is a 'honey pot?)

What is epoll?