EECS 489 - Winter 2024

Discussion 1

Assignment 1 is out!

- Due Date: Friday, January 26th @ 11:59 pm EDT
  - Autograder + Repos will be released soon, sorry for the delay!
- Start early and have fun!
- Make sure to fill out the GitHub info form ASAP when it is out!
  - This is your only way you will be able to submit to the autograder

Today

___

- Socket Programming
    - Stepping through functions
    - Walking through code example

Socket Programming: Introduction

———

- How does a server and a client communicate over the network?

- Host (Server) ------ Network ------ Host (Client)


- Sockets will save the day!
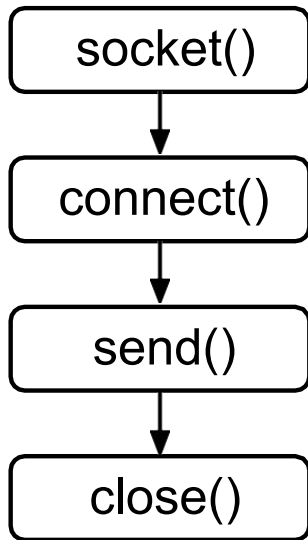
Socket Programming: Introduction

———

- ● What is a socket?
  - ○ A socket is a communication endpoint at the hosts (applications) for communication flow
- ● Why do we use sockets?
  - ○ A socket provides an API for exchanging data
    - ■ Can be on different machines, or even the same machine! (Different processes)

Socket Programming: Server Side
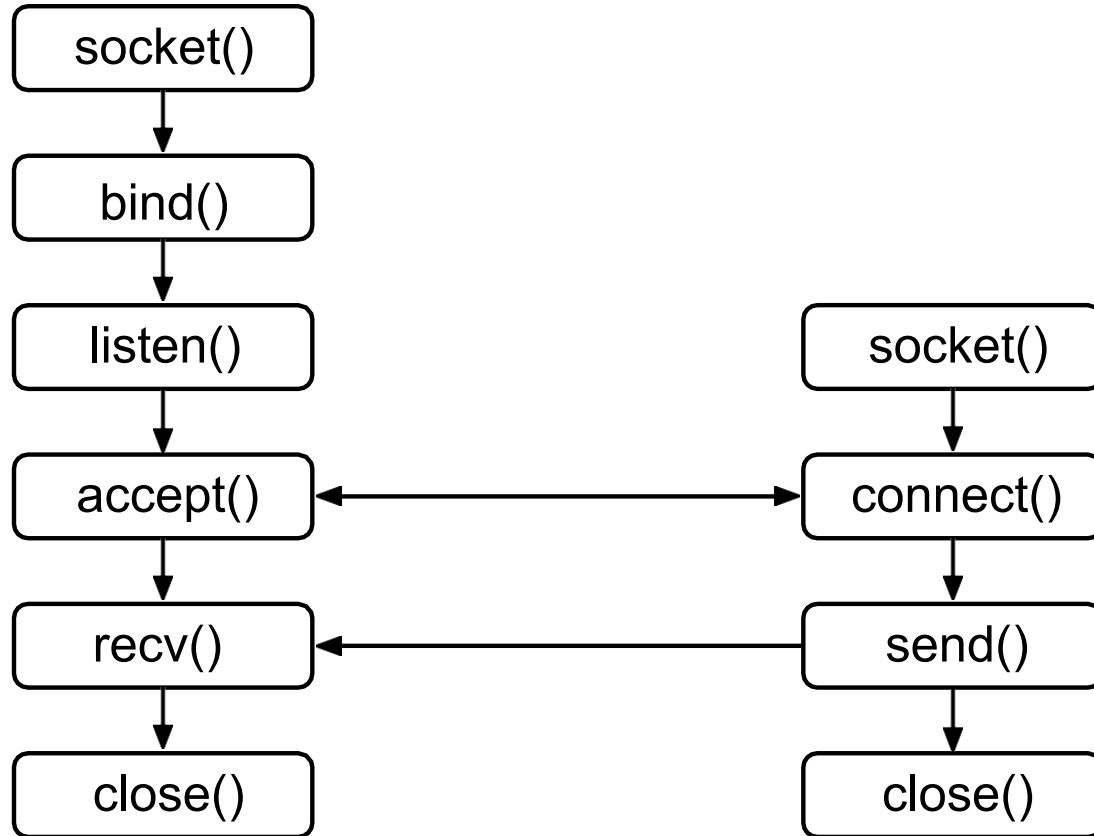
– – –

```
socket()
   ↓
bind()
   ↓
listen()
   ↓
accept()
   ↓
recv()
   ↓
close()
```
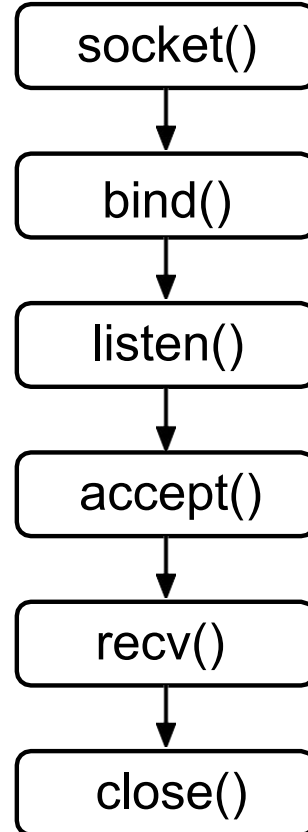
Socket Programming: Client Side

- - -

```
socket()
   ↓
connect()
   ↓
send()
   ↓
close()
```

Socket Programming: Complete Flow

– – –

Socket Programming: socket()

———

- Creates a socket
- Returns a file descriptor that is used to refer to the socket object created
- sockfd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

int socket(int *domain,* **int** *type,* **int** *protocol***);**

Socket Programming: bind()

———

- Binds an address (unique local name) to a socket
  - Sockets are unnamed when initially created
  - Returns 0 if successful, or -1 if error
- bind(sockfd, (struct sockaddr *) &addr, sizeof(addr));

```
int bind(int sockfd, const struct sockaddr *addr,
         socklen_t addrlen);
```

- NOTE: Do not put `using namespace std;` to avoid having issues with C++'s std::bind function

Socket Programming: socket addr

———

- Server:
  - struct sockaddr_in addr;
  - memset(&addr, 0, sizeof(addr));
  - addr.sin_family = AF_INET;
  - addr.sin_addr.s_addr = INADDR_ANY;
  - addr.sin_port = htons(port);
- Client:
  - struct hostent* host = gethostbyname(hostname); // server hostname
  - struct sockaddr_in addr;
  - addr.sin_family = AF_INET;
  - memcpy(&(addr->sin_addr), host->h_addr, host->h_length);
  - addr.sin_port = htons(server_port);

Quick Aside: Byte Order

———

- Internet: Big Endian
- Host: Machine Dependent

- Refresher: Represent the hex value **b34f**
  - Big Endian: **b34f**
  - Little Endian: **4fb3**

Function	Description
htons()	host to network short
htonl()	host to network long
ntohs()	network to host short
ntohl()	network to host long

Socket Programming: listen()

___

- Listens for connections on a socket
  - Takes in max number of pending connections allowed (backlog)
  - Returns 0 upon successful completion, or -1 if error
- listen(sockfd, 10);

```
int listen(int sockfd, int backlog);
```

Socket Programming: connect()

- Initiate a connection on a socket
  - Returns 0 if successful, or -1 if error
- socklen_t addr_len = sizeof(addr);
- connect(sockfd, (struct sockaddr *) &addr, &addr_len);

```
int connect(int sockfd, const struct sockaddr *addr,
            socklen_t addrlen);
```

Socket Programming: accept()

- Accepts a connection on a socket
  - Returns the file descriptor of the accepted socket, or -1 if error
- socklen_t addr_len = sizeof(addr);
- int connectfd = accept(sockfd, (struct sockaddr *) addr, &addr_len);

```
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

Socket Programming: send()

———

- Sends a message on a socket
  - Returns the number of bytes read in, or -1 if error
- bytes_sent = send(connectfd, buffer, len, 0);
  - Last part is a flag, look at manual pages for different ones

ssize_t send(int *sockfd*, const void *buf, size_t *len*, int *flags*);

Socket Programming: recv()

———

- Receive a message from a socket
  - Returns the number of bytes read in, or -1 if error
- bytes_recv = recv(connectfd, buffer, len, MSG_WAITALL);
  - Last part is a flag, look at manual pages for different ones

```
ssize_t recv(int sockfd, void *buf, size_t len, int flags);
```

Socket Programming: Resources

———

- [Beej's Guide to Network Programming](#)
- [Discussion Socket Example](#) (from EECS 482)
- Linux manual pages
  - `man socket`

Useful Libraries

———

- `#include <arpa/inet.h>`   `// htons(), ntohs()`
- `#include <netdb.h>`   `// gethostbyname(), struct hostent`
- `#include <netinet/in.h>`   `// struct sockaddr_in`
- `#include <stdio.h>`   `// perror(), fprintf()`
- `#include <string.h>`   `// memcpy()`
- `#include <sys/socket.h>`   `// getsockname()`
- `#include <unistd.h>`   `// stderr`
- `#include <time.h>`   `//time(&time_t)`

Wrap-Up

———

- Thanks for coming!
- Make sure to start Assignment 1 soon!
- Any last questions?