# EECS 489 - WN 23

Discussion 1

# A1 is out

- Course website https://github.com/morleydragon/eecs489
- Please register your github username ASAP. https://forms.gle/a79fPrA7XpzaBH9P6
- Due Fri., 27th Jan. 11:59:59 p.m.
  - 1 late day for A1 only
- Get yourself familiar with the basic socket programming
- START EARLY!
- Have fun : - )
- Sign up group for A2/3/4 https://forms.gle/Mi3tsZzcNoWH1XPW7

# Plan

- Introduction to socket programming
- A Demo

# Socket Programming: Intro

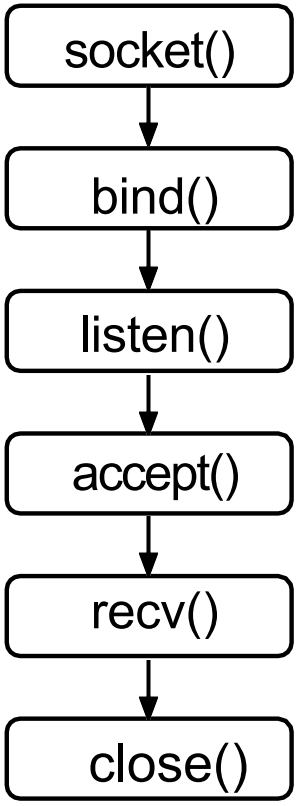
What is a socket?

- Communication endpoint at the end hosts.

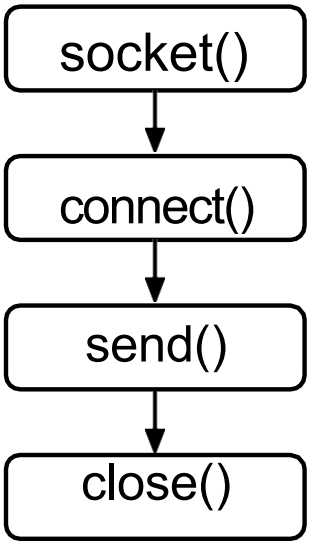
Why do we use a socket?

- It provides an application programming interface to exchange data between processes on the same or different machines.

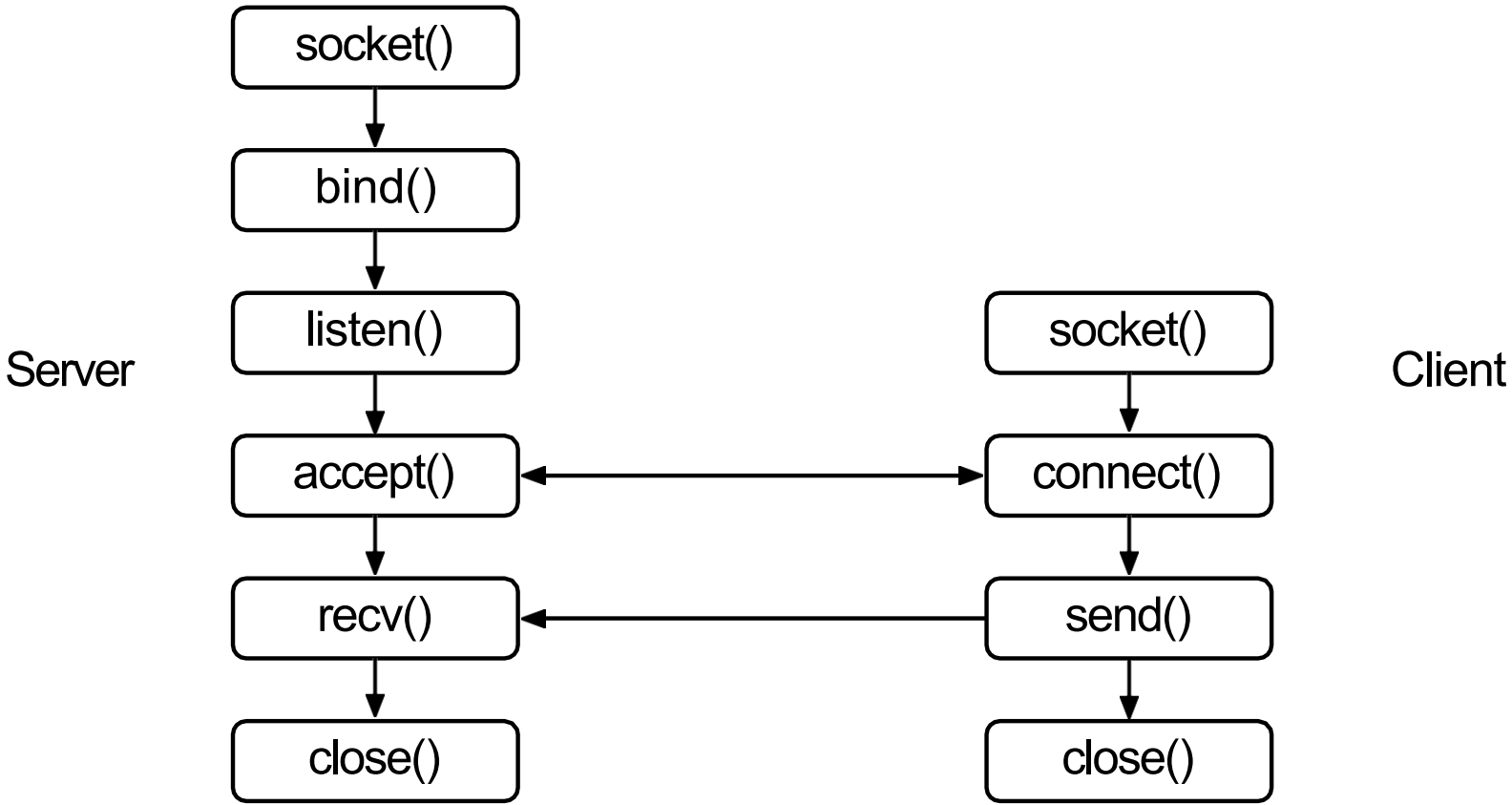
# Socket Programming: Server Side



# Socket Programming: Client Side



# Socket Programming: Complete Flow



# Socket Programming: socket()

Create a socket

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

sockfd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

# Socket Programming: bind()

Bind an address to a socket

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

bind(sockfd, (struct sockaddr *) &addr, sizeof(addr));

# 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 *server=gethostbyname(hostname);
struct sockaddr_in addr;
memset(&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = *(unsigned long *) server->h_addr_list[0];
addr.sin_port = htons(server_port);
```

# Byte Order

Internet Convention: Big Endian
Host: machine-dependent

represent the two-byte hex: `b34f`
Big-Endian: `b34f`
Small-Endian: `4fb3`

| Function | Description |
|---|---|
| `htons()` | `h`ost `to` `n`etwork `s`hort |
| `htonl()` | `h`ost `to` `n`etwork `l`ong |
| `ntohs()` | `n`etwork `to` `h`ost `s`hort |
| `ntohl()` | `n`etwork `to` `h`ost `l`ong |

# Socket Programming: listen()

Listen for connections on a socket

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

listen(sockfd, 10);

# Socket Programming: connect()

Initiate a connection on a socket

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

socklen_t addr_len = sizeof(addr);
connect(socked, (struct sockaddr *) &addr, addr_length);

# Socket Programming: accept()

Accept a connection on a socket

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

socklen_t addr_len = sizeof(addr);
int conn = accept(sockfd, (struct sockaddr *) &addr, &addr_len);

# Socket Programming: send()

Send a message on a socket

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

send(conn, buffer, MSG_SIZE, MSG_NOSIGNAL);

# Socket Programming: recv()

Receive a message on a socket

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

byte_recved = recv(conn, buffer, len, MSG_NOSIGNAL);

# Socket Programming:A Demo

# Socket Programming: Resources

- Beej's Guide to Network Programming
- eecs482/bgreeves-socket-example: Sockets example from discussion
- man is always your good friend
    - man socket

# Useful Libraries for A1

```
#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)
```

# Thanks

Have a good one!