

# CSCI 4061 Discussion 12

4/16/18



UNIVERSITY OF MINNESOTA  
**Driven to Discover<sup>SM</sup>**

# Overview

- Sockets
  - Client
  - Server
- Exercise



# Sockets

- IP Connection abstraction.
- Uniquely identified via an address and port number.
- Linux treats a socket as a file descriptor.



# Server Connection Steps

1. Create a socket
2. Bind it to an address
3. Listen for a connection
4. Accept a connection
5. Read/Write as needed
6. Close connection when done



# Server Connection Calls

**// Create a socket. Returns an file descriptor if successful, else -1.**

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

**// Bind it to an address.**

**int bind(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);**

**// Listen for a connection.**

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

**// Accept a connection.**

**int accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen);**

**// Close a connection.**

**close(int sockfd);**



# Client Connection Steps

1. Create a socket
2. Connect to a remote address
3. Read/Write as needed
4. Close connection when done



# Client Connection Calls

**// Create a socket. Returns an file descriptor if successful, else -1.**

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

**// Connect to a remote host.**

**int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);**

**// Close a connection.**

**close(int sockfd);**



# Socket I/O

- Opening a socket returns a file descriptor, which can be read and written to just like a file.
- Sockets also have their own specific functions for I/O.

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

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





# Programming Assignment 4

- Due April 27th
- Covers sockets, networking material.
- Read the document carefully, ask questions if you have them.



# Basic Idea

- Client reads votes files, sends requests such as adding votes and getting the results from the server.
- The server maintains a representation of the region tree, with the current vote totals, and responds to client(s).



# Notes and Help

- You need to consider multiple clients, on either your local machine, or one on the network.
- Your requests and responses need to follow the defined format **exactly**.
- We will release test cases, along with executables of client and server to test against soon.



# Exercise

- Two files are provided for today, a server and a client.
- Simply fill in the necessary function calls where the 'TODO' comments are.
- The client application takes in a file name and sends it to the server, which then prints the contents.

