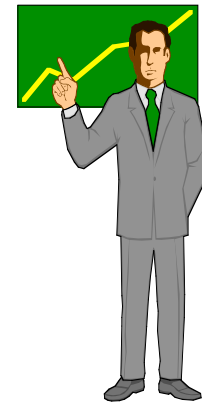


CS437: Week 1 Tutorial

Socket Programming.

Goal: Introduce the Unix socket programming and tell you all you need for Exercise 1!

We will describe how to use UDP and TCP.
And we will review working code that you can use for you exercise.





Protocols: UDP and TCP

- **UDP: User Datagram Protocol**
 - Computers on the internet can send short messages to each other
 - **Unreliable:** Messages may be lost, reordered, or duplicated
 - **Connectionless:** Similar to sending letters in the mail
- **TCP: Transmission Control Protocol**
 - Computers can send long streams of data to each other
 - TCP guarantees that messages are delivered to the receiver in the order that they were sent
 - Recovers lost messages
 - Even slows down when the network is congested
 - **Connection oriented:** We need to set up the connection before we send.



The important functions we will use!

- `socket()`: Creates a socket, a communications endpoint
 - Sockets are used to send and receive messages for UDP and TCP
- `setsockopt()`: Set options on a socket
- `bind()`: Associate a socket with an address
- `gethostname()`: Get the name of the machine where the program is running
- `gethostbyname()`: Get the the address of the machine having a particular name
 - This function can use DNS to look up the IP address of ugrad12
- `sendto()`: Sends information on a socket to another machine
 - We need to specify the socket we want to use to send AND where we want to send the data -- the ip address and the port.
- `recvfrom()`: Receives information on a socket
 - We need to specify the socket
- `listen()`: Listen for machines trying to connect to this machine
 - Our `net_server` listens for connection attempts coming from clients
- `connect()`: Establish a connection with another machine
 - We need to connect to use TCP!
 - The other machine needs to be listening for connection attempts.
- `accept()`: Accept a connection
- `send()`: Send data over a connection
 - This function is used for TCP
- `recv()`: Read data from a connection
 - This can be used for TCP or UDP, but it usually used for TCP.
- `select()`: Multiplexing IO operations



Our Sample Programs

- UDP:
 - `ucast.c`: This program uses UDP to send and receive messages. We can send and receive using a single program.
- TCP:
 - `net_server.c`: This program listens for TCP connection attempt from `net_clients` and reads information sent by connected clients.
 - `net_client.c`: This program connects to a `net_server` on another machine and sends information to it.
- You can use these programs when you program Exercise 1. They include all of the functions associated with sending and receiving messages that you need to complete the exercise. However, there is a lot of work that you need to handle, so start early.