Networking

Create simple programs that communicate through the network

Objectives

- Learn the basics of how the internet works
- Learn the basics of networking
- Be familiar with the vocabulary of networking
- Extend and rewrite and existing App to enable networking
- Try out stuff and have some experience before jumping into the deep end...

Materials & Resources

Environment

- Windows operating system
- CodeBlocks on windows

Training

Material	Time
IP addresses and DNS	6:44
Packet, routers, and reliability	6:25
HTTP and HTML	7:06
Encryption and public keys	6:39
The OSI Model Demystified	18:40
Hub, Switch or Router? Network Devices Explained	7:39
TCP: Transmission control protocol Networking tutorial (12 of 13)	8:28
TCP and UDP	3:06
UDP and TCP: Comparison of Transport Protocols	11:34

Bonus task: Explain someone how the internet works! To your grandma, dad, kid, neighbor in the elevator, cat, you name it.

The following materials could be used during the project. At home please start reading it only if you have a lot's of time.

Material	Time
winsock2.h usage	-
Creating a TCP Client in C++	22:11
Creating a TCP Server in C++	30:30

Material Review

- OSI model
  - MAC vs. IP and DNS
- What are packets?
- What does the
  - Hub
  - Switch
  - Router
  - Gateway do?
- What is HTTP and HTML?
- What is encryption and public keys?
- What is a server?
- What is a client?
- What is TCP and UDP?
- What are the differences?

Excercises

First you need to be link to the linker the ws2_32.lib:

- Project --> project build options --> Linker settings
- Under the Link libraries, push the Add button, and type `ws2_32` --> click OK--> and OK again

Basics (mandatory) tasks

- Ping Google
- Create a server

Advanced task

- Extend the server

Make sure to comment each section in the code that does something significant! Also make sure that you handle errors.

Ping Google

Create a client app that pings `google.com` and gets the latency. Print out the result!

- ping.c

Hint: Use the built-in windows ping command with the system function!

Create a server

- Create the server app that responds to the communication from the client app. It simply sends back what it has received.
- Create a client app that connects to your server app. It sends a message to the server, waits for the response and prints out the response. Yepp, the server app is running on your computer also, but the message goes out and comes in from the internet.
- send_msg.c

Extend the server

Extend your server application so that every interaction is logged into a file. Make sure to use something unique in the log file for each entry, for example use timestamps.

- Extended_server.c

Individual Workshop Review

Please follow the styleguide: Our C styleguide

- Is the directory structure and the name of the files correct?
- Are the includes placed on the top of the files?
- Is the indentation good in each file?
- Is there unnecessary code?
- Can you find unnecessary code in comments?
- Is there unnecessary code duplication?
- Are there unnecessary empty blocks?
- Can you spot unused variables?
- Is the commit message meaningful?

Solutions

Solutions