

# Client Server Application

## Thomas O'Brien

# An exercise in TCP communications via Unix programming

# Justification

- To develop an in-depth understanding of the TCP/IP stack
  - Relevant to Site Reliability/DevOps Engineering

# Server

- Establishes ownership over available port on some UNIX based machine
  - Listens for an incoming TCP request to make a 3 way handshake
  - Decodes incoming files and stores them at a given existing directory

# Client

- Attempts to make a simulated TCP connection with a server
    - Utilizes a TCPHeader struct
  - Sends encoded files from the client side to the server side
  - Returns whether or not the file was sent over successfully

# Technologies

- Golang is a powerful systems programming language gaining popularity in modern tech stacks
  - Unix programming libraries are utilized for lower level programming



```
ThomasObrint@MacBook-Pro-3:~/Documents/Stuff/School/SS/Semester_2/CSIS_402/project/client-server-application]: go run unix-client.go hello.  
ThomasObrint@MacBook-Pro-3:~/Documents/Stuff/School/SS/Semester_2/CSIS_402/project/client-server-application]: cat testingNewFile.txt  
hello world!  
I'm just testing this out  
to see if I can read a file!  
ThomasObrint@MacBook-Pro-3:~/Documents/Stuff/School/SS/Semester_2/CSIS_402/project/client-server-application]:
```

ThomasObrien@MacBook-Pro-3: [/tmp]

```
|Thomas0brien@MacBook-Pro-3:~/Documents/Stuff/School/SS/Semester_2/CMSI_402/p  
|ject/client-server-application]: go run unix-server.go  
|Let's listen  
|Found connection
```

[402] 0: bash

"MacBook-Pro-3.local" 14:56 23-Apr-