# Client/Server Chat Program
# User Guide

Wilson Sue & Charmie Jung
November 18, 2023

## Purpose

The chat.c program's purpose is to allow you to communicate between a server and clients. Also, this program helps us to understand the flow of TCP traffic. Utilizing Wireshark, we are able to see the traffic flow of the communication within the program and between different operating systems.

## Installing

### Obtaining

```
git clone https://github.com/CharminJungBCIT/Project3980.git
```

### Building

```
Cd ./Project3980

chmod u+x ./generate-flags.sh

chmod u+x ./generate-cmakelists.sh

chmod u+x ./change-compiler.sh

chmod u+x ./build.sh

./generate-flags.sh

./generate-cmakelists.sh

./change compiler.sh -c gcc
Or
./change compiler.sh -c clang

./build.sh
```

## Features

- Once Server address has been set and client has connected,
  You can type anything on one either mode and it will show what you've typed on the other mode.
- Additionally, you can use io redirection to send a message with a text file.

- Ctrl-d to close the server connection on the client's end.

## Built-in Commands

Project3980 supports the following built-in commands:

| Command | Purpose |
|---|---|
| chmod u+x | Gaining access to the shell files and commands |
| generate-Cmakelists.sh | Running the Cmakelists that is compatible with the operating system |
| generate-flags.sh | Running flags for potential errors |
| change-compiler.sh (gcc or clang) | Running the program with a compiler for error handling |
| build.sh | Running the program for any errors and success |

## Limitations

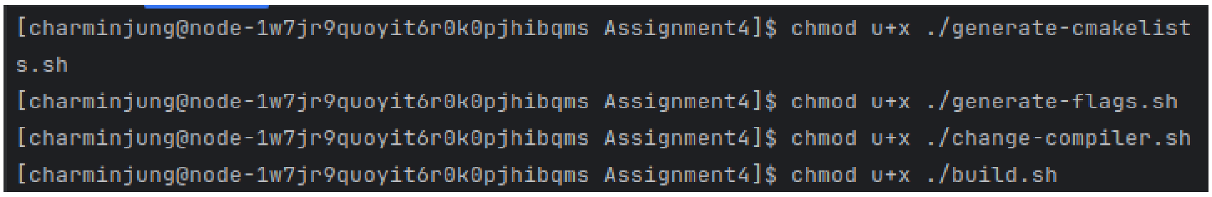
- The program must have an IP address and port
- Ip address and port must be the same
- The server must first establish a connection, then the client can connect.

The following command types are not supported:

- Pipelines (|)
- Lists (;, &, &,&, ||)
- Compound (if, case, until, while, for, ())
- Functions

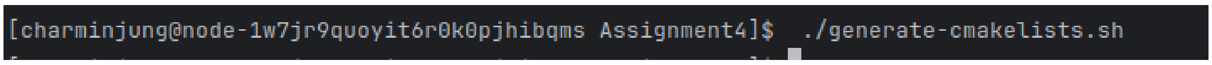
# Examples

chmod u+x commands



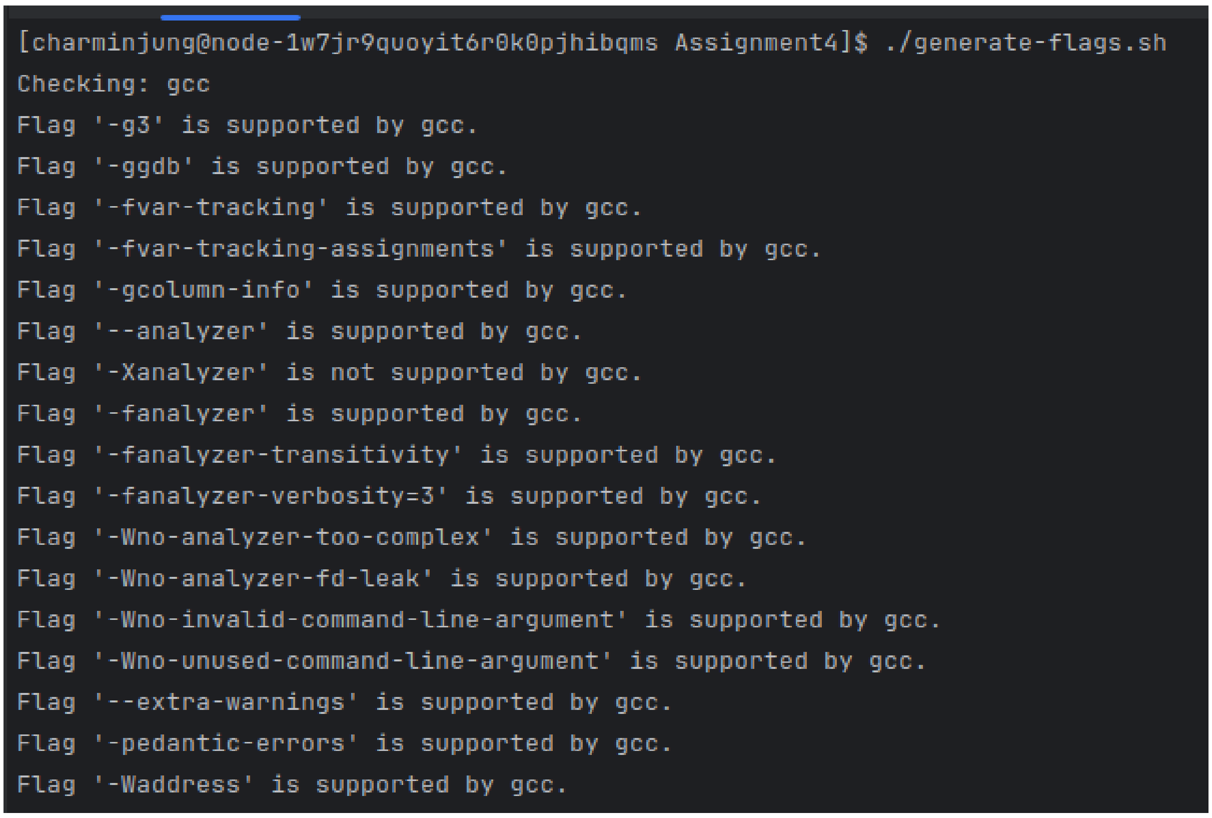
```
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ chmod u+x ./generate-cmakelist
s.sh
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ chmod u+x ./generate-flags.sh
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ chmod u+x ./change-compiler.sh
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ chmod u+x ./build.sh
```

Running ./generate-cmakelists.sh

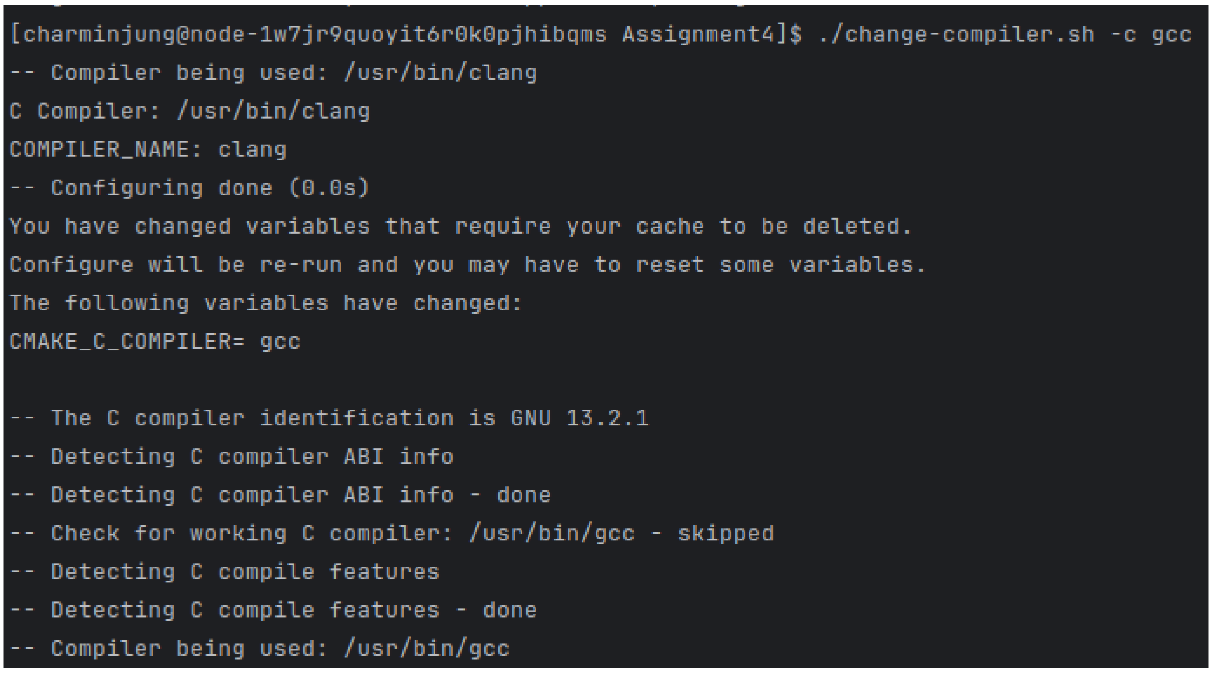


```
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$  ./generate-cmakelists.sh
```

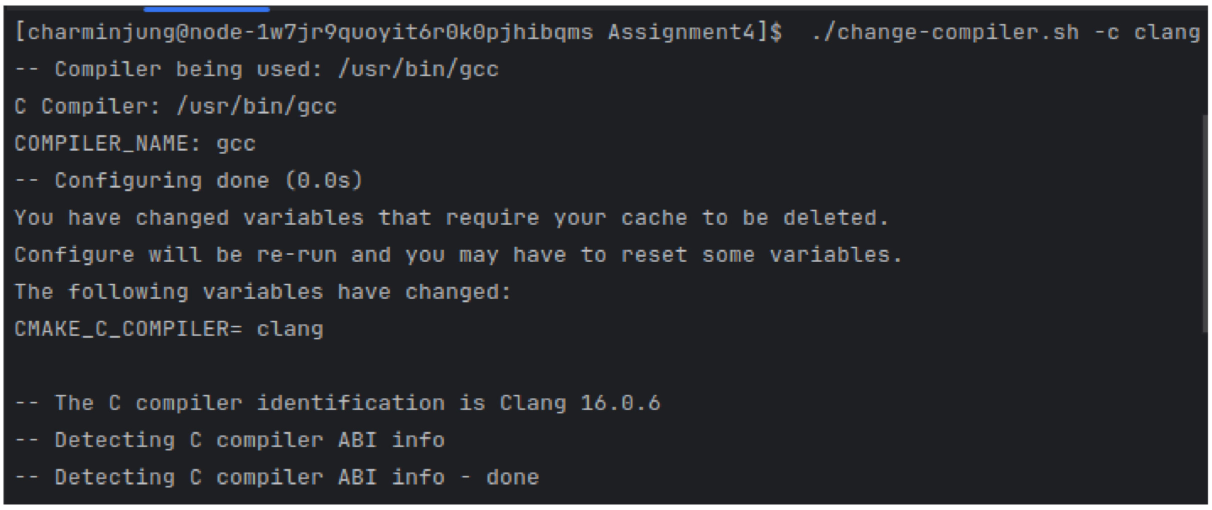
## Running the generate-flags shell


```
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ ./generate-flags.sh
Checking: gcc
Flag '-g3' is supported by gcc.
Flag '-ggdb' is supported by gcc.
Flag '-fvar-tracking' is supported by gcc.
Flag '-fvar-tracking-assignments' is supported by gcc.
Flag '-gcolumn-info' is supported by gcc.
Flag '--analyzer' is supported by gcc.
Flag '-Xanalyzer' is not supported by gcc.
Flag '-fanalyzer' is supported by gcc.
Flag '-fanalyzer-transitivity' is supported by gcc.
Flag '-fanalyzer-verbosity=3' is supported by gcc.
Flag '-Wno-analyzer-too-complex' is supported by gcc.
Flag '-Wno-analyzer-fd-leak' is supported by gcc.
Flag '-Wno-invalid-command-line-argument' is supported by gcc.
Flag '-Wno-unused-command-line-argument' is supported by gcc.
Flag '--extra-warnings' is supported by gcc.
Flag '-pedantic-errors' is supported by gcc.
Flag '-Waddress' is supported by gcc.
```

## Running change-compiler shell gcc

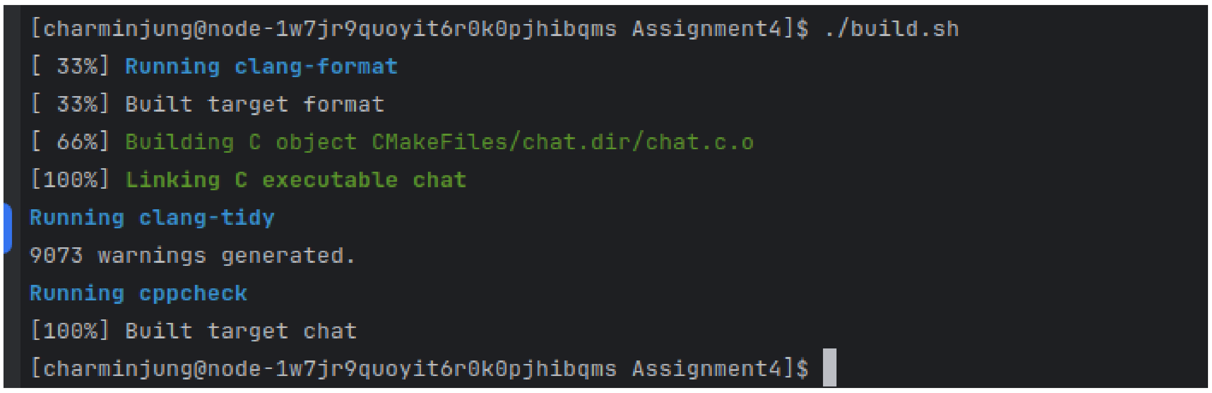

```
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ ./change-compiler.sh -c gcc
-- Compiler being used: /usr/bin/clang
C Compiler: /usr/bin/clang
COMPILER_NAME: clang
-- Configuring done (0.0s)
You have changed variables that require your cache to be deleted.
Configure will be re-run and you may have to reset some variables.
The following variables have changed:
CMAKE_C_COMPILER= gcc

-- The C compiler identification is GNU 13.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/gcc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Compiler being used: /usr/bin/gcc
```

## Running Change compiler shell clang



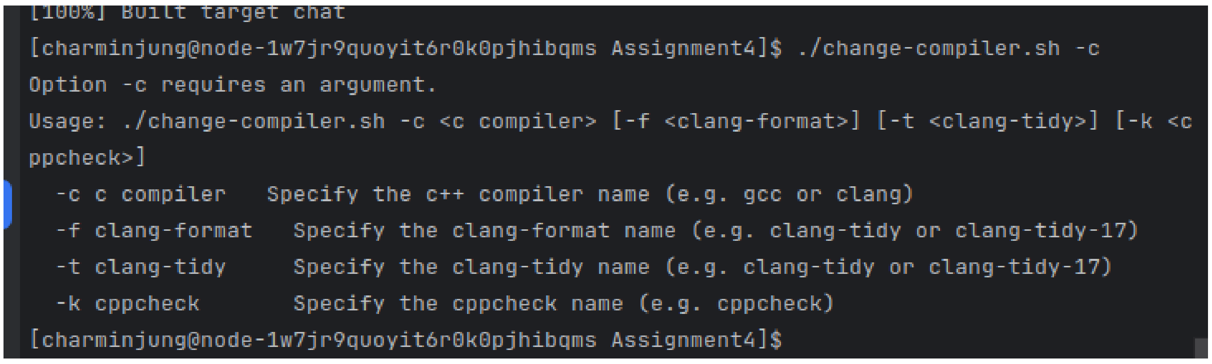
```
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$  ./change-compiler.sh -c clang
-- Compiler being used: /usr/bin/gcc
C Compiler: /usr/bin/gcc
COMPILER_NAME: gcc
-- Configuring done (0.0s)
You have changed variables that require your cache to be deleted.
Configure will be re-run and you may have to reset some variables.
The following variables have changed:
CMAKE_C_COMPILER= clang

-- The C compiler identification is Clang 16.0.6
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
```

## Running Build shell



```
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ ./build.sh
[ 33%] Running clang-format
[ 33%] Built target format
[ 66%] Building C object CMakeFiles/chat.dir/chat.c.o
[100%] Linking C executable chat
Running clang-tidy
9073 warnings generated.
Running cppcheck
[100%] Built target chat
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$
```

## Invalid command



```
[100%] Built target chat
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$ ./change-compiler.sh -c
Option -c requires an argument.
Usage: ./change-compiler.sh -c <c compiler> [-f <clang-format>] [-t <clang-tidy>] [-k <c
ppcheck>]
  -c c compiler   Specify the c++ compiler name (e.g. gcc or clang)
  -f clang-format   Specify the clang-format name (e.g. clang-tidy or clang-tidy-17)
  -t clang-tidy     Specify the clang-tidy name (e.g. clang-tidy or clang-tidy-17)
  -k cppcheck       Specify the cppcheck name (e.g. cppcheck)
[charminjung@node-1w7jr9quoyit6r0k0pjhibqms Assignment4]$
```