



ft\_irc

## Internet Relay Chat

*Summary:*

*This project is about creating your own IRC server.*

*You will use an actual IRC client to connect to your server and test it.*

*Internet is ruled by solid standards protocols that allow connected computers to interact with each other.*

*It's always a good thing to know.*

*Version: 8*

# Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>General rules</b>	<b>3</b>
<b>III</b>	<b>Mandatory Part</b>	<b>4</b>
III.1	Requirements . . . . .	5
III.2	For MacOS only . . . . .	6
III.3	Test example . . . . .	6
<b>IV</b>	<b>Bonus part</b>	<b>7</b>
<b>V</b>	<b>Submission and peer-evaluation</b>	<b>8</b>

# Chapter I

## Introduction

**Internet Relay Chat** or IRC is a text-based communication protocol on the Internet. It offers real-time messaging that can be either public or private. Users can exchange direct messages and join group channels.

IRC clients connect to IRC servers in order to join channels. IRC servers are connected together to form a network.

# Chapter II

## General rules

- Your program should not crash in any circumstances (even when it runs out of memory), and should not quit unexpectedly.  
If it happens, your project will be considered non-functional and your grade will be 0.
- You have to turn in a **Makefile** which will compile your source files. It must not relink.
- Your **Makefile** must at least contain the rules:  
`$(NAME)`, `all`, `clean`, `fclean` and `re`.
- Compile your code with `c++` and the flags `-Wall -Wextra -Werror`
- Your code must comply with the **C++ 98 standard**. Then, it should still compile if you add the flag `-std=c++98`.
- Try to always develop using the most C++ features you can (for example, choose `<cstring>` over `<string.h>`). You are allowed to use C functions, but always prefer their C++ versions if possible.
- Any external library and Boost libraries are forbidden.

# Chapter III

## Mandatory Part

Program name	ircserv
Turn in files	Makefile, *.{h, hpp}, *.cpp, *.tpp, *.ipp, an optional configuration file
Makefile	NAME, all, clean, fclean, re
Arguments	port: The listening port password: The connection password
External functs.	Everything in C++ 98. socket, close, setsockopt, getsockname, getprotobyname, gethostbyname, getaddrinfo, freeaddrinfo, bind, connect, listen, accept, htons, htonl, ntohs, ntohl, inet_addr, inet_ntoa, send, recv, signal, sigaction, lseek, fstat, fcntl, poll (or equivalent)
Libft authorized	n/a
Description	An IRC server in C++ 98

You have to develop an IRC server in C++ 98.

You **mustn't** develop a client.

You **mustn't** handle server-to-server communication.

Your executable will be run as follows:

```
./ircserv <port> <password>
```

port: The port number on which your IRC server will be listening to for incoming IRC connections.

password: The connection password. It will be needed by any IRC client that tries to connect to your server.

Even if poll() is mentionned in the subject and the evaluation scale, you can use any equivalent such as select(), kqueue(), or epoll().









