



ft\_irc

Internet Relay Chat

요약:

이 프로젝트는 자신만의 IRC 서버를 만드는 것을 목표로 합니다. 실제 IRC 클라이언트를 사용하여 직접 만든 서버에 접속하고 테스트할 예정입니다. 인터넷은 서로 상호 작용하는 연결된 컴퓨터들을 위한 견고한 표준 프로토콜로 운영되고 있습니다. 이러한 표준들을 알고 있으면 좋습니다.

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

인터넷 릴레이 챗, 또는 IRC는 인터넷에서 사용되는 텍스트 기반의 커뮤니케이션 프로토콜입니다. 실시간으로 진행되는 메시지 교환을 가능하게 하며, 이는 공개적으로 또는 개인적으로 이루어질 수 있습니다. 사용자들은 서로 직접 메시지를 주고받거나 그룹 채널에 참여할 수 있습니다.

IRC 클라이언트는 채널에 참여하기 위해 IRC 서버에 연결되며, 이러한 서버들은 서로 연결되어 하나의 네트워크를 형성합니다.

# Chapter II

## General rules

- 프로그램은 어떠한 상황에서도 충돌하거나 갑작스럽게 종료되어서는 안 됩니다. 이런 일이 발생하면 프로젝트는 기능하지 않는 것으로 간주되고, 점수는 0점이 됩니다.
- 소스 파일을 컴파일할 Makefile을 제출해야 하며, 재링크 없이 작동해야 합니다.
- Makefile은 \$(NAME), all, clean, fclean, re 규칙을 반드시 포함해야 합니다.
- 코드는 c++와 함께 -Wall -Wextra -Werror 플래그를 사용해 컴파일해야 합니다.
- C++ 98 표준을 준수해야 하며, -std=c++98 플래그를 추가해도 컴파일이 가능해야 합니다.
- 가능한 한 많은 C++ 기능을 활용하려고 노력하고, 가능하면 항상 C++ 버전의 함수를 사용하세요. (예를 들어, <cstring>을 <string.h>대신 사용하세요.) C 함수를 사용할 수는 있지만, 가능하다면 항상 C++ 버전을 선호해야 합니다.
- 모든 외부 라이브러리와 Boost 라이브러리의 사용은 금지됩니다.

# Chapter III

## Mandatory Part

Program name	ircserv
Turn in files	Makefile, *.{h, 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

IRC 서버는 C++ 98로 개발되어야 합니다.

클라이언트 개발은 포함되지 **않아야** 합니다.

서버 간 통신은 포함되지 **않아야** 합니다.

실행 파일은 다음과 같이 지정된 포트와 비밀번호를 사용하여 실행됩니다.:

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

- port: IRC 서버가 들어오는 IRC 연결을 위해 대기할 포트 번호입니다.
- password: 연결 비밀번호로, 서버에 접속하려는 모든 IRC 클라이언트는 이 비밀번호가 필요합니다.



poll()을 사용할 수 있으며 select(), kqueue(), epoll()과 같은 동등한 기능도 사용 가능합니다.

### III.1 Requirements

- 서버는 여러 클라이언트를 동시에 처리할 수 있어야 하며, 절대 멈추지 않아야 합니다.
- 포킹은 허용되지 않으며, 모든 I/O 작업은 **논블로킹**으로 처리되어야 합니다.
- read, write, listen 등을 포함한 모든 작업을 처리하기 위해 **단 하나의 poll()** (또는 동등한 기능)만 사용해야 합니다.



논블로킹 파일 디스크립터를 사용하면 poll() 없이 read/recv 또는 write/send를 사용할 수 있지만, 서버가 블로킹되지 않으며 시스템 자원을 더 많이 소모합니다.

따라서 poll() (또는 그에 상응하는 기능) 없이 파일 디스크립터에서 read/recv 또는 write/send를 시도할 경우 점수는 0점입니다.

- 여러 IRC 클라이언트 중 하나를 기준 클라이언트로 선택해야 합니다. 이 클라이언트는 평가 과정에서 사용됩니다.
- 기준 클라이언트는 서버에 오류 없이 연결되어야 합니다.
- 클라이언트와 서버 간의 통신은 **TCP/IP(v4 또는 v6)**를 통해 이루어져야 합니다.
- 기준 클라이언트를 이용해 서버에 접속하는 경험은 공식 IRC 서버와 유사해야 합니다. 다만, 아래 서술된 기능들만 구현하면 됩니다:
  - 인증하기, 닉네임과 사용자 이름 설정하기, 채널에 참여하기, 개인 메시지 보내기 및 받기 등을 기준 클라이언트를 통해 수행할 수 있어야 합니다.
  - 한 클라이언트가 채널에 보낸 모든 메시지는 채널에 참여한 다른 모든 클라이언트에게 전달되어야 합니다.
  - **운영자**와 일반 사용자가 있어야 합니다.
  - 채널 **운영자**를 위한 특정 명령어를 구현해야 합니다.:

\* KICK - 채널에서 클라이언트를 추방

\* INVITE - 클라이언트를 채널에 초대

\* TOPIC - 채널의 주제를 변경하거나 확인

\* MODE - 채널의 모드 변경

· i: 초대 전용 채널 설정/해제

· t: TOPIC 명령의 권한을 운영자 전용으로 제한 설정/해제

· k: 채널 키(비밀번호) 설정/해제

· o: 채널 운영자 권한 부여/박탈

- 1: 채널의 사용자 수 제한 설정/해제
- 깨끗하고 정돈된 코드 작성이 중요합니다.

## III.2 For MacOS only



MacOS에서는 다른 Unix 운영체제와는 다르게 `write()`가 구현되어 있지 않기 때문에, `fcntl()` 사용이 허용됩니다. Unix 운영체제와 비슷한 동작을 위해, 파일 디스크립터를 논블로킹 모드로 사용해야 합니다.



`fcntl()`은 오직 `fcntl(fd, F_SETFL, O_NONBLOCK)` 형태로만 사용할 수 있으며, 다른 어떤 플래그도 사용해서는 안 됩니다.

## III.3 Test example

가능한 모든 오류와 문제(부분적인 데이터 수신, 낮은 대역폭 등)를 철저히 검증해야 합니다.

서버가 보낸 모든 내용을 올바르게 처리하는지 확인하기 위해서는 `nc`를 사용한 간단한 테스트를 할 수 있습니다.:

```
\$> nc 127.0.0.1 6667
com~Dman~Dd
\$>
```

`ctrl+D`를 이용하여 'com', 'man', 'd\n' 등의 명령을 여러 부분으로 나눠서 보낼 수 있습니다.

명령을 처리하기 위해, 수신된 패킷들을 모아 전체 명령을 재구성해야 합니다.

# Chapter IV

## Bonus part

실제 IRC 서버와 더욱 비슷하게 만들고 싶다면, 추가할 수 있는 기능들이 있습니다.:

- 파일 전송 처리 기능
- 봇 기능



보너스 파트는 필수 부분이 완벽하게 수행되었을 때만 평가됩니다. 여기서 '완벽함'은 필수 부분이 전체적으로 수행되고 어떤 기능적 문제도 없음을 의미합니다. 만약 필수 요구사항 중 하나라도 미충족된 경우, 보너스 파트는 평가되지 않습니다.



# Chapter V

## Submission and peer-evaluation

평소와 같이 Git 저장소에 과제를 제출합니다. 방어 평가는 저장소 내부의 작업만을 대상으로 합니다. 파일 이름을 잘 확인하여 오류가 없는지 확인하세요.

프로젝트에 대한 테스트 프로그램을 만드는 것이 권장되지만, 이는 제출되거나 평가되지 않습니다. 이러한 테스트는 서버를 평가하거나 미래에 다른 `ft_irc` 프로젝트를 평가할 때 유용할 수 있습니다. 평가 과정에서 필요한 모든 테스트를 사용할 수 있습니다.



평가 과정에서 기준 클라이언트가 사용될 것입니다.



16D85ACC441674FBA2DF65190663F432222F81AA0248081A7C1C1823F7A96F0B74495  
15056E97427E5B22F07132659EC8D88B574BD62C94BB654D5835AAD889B014E078705  
709F6E02