



ft_irc

인터넷 릴레이 채팅

요약:

이 프로젝트는 나만의 IRC 서버를 만드는 것입니다.

실제 IRC 클라이언트를 사용하여 서버에 연결하고 테스트합니다.

*인터넷은 연결된 컴퓨터가 서로 상호 작용할 수 있도록 하는 견고한 표준 프로토콜에
의해 지배됩니다.*

항상 알아두면 좋은 정보입니다.

버전: 7

콘텐츠

I	소개	2
II	일반 규칙	3
III	필수 부분	4
III.1	요구 사항.....	5
III.2	MacOS 전용	6
III.3	테스트 예	6
IV	보너스 부품	7
V	제출 및 동료 평가	8

1장 소개

인터넷 릴레이 채팅 또는 IRC는 인터넷의 텍스트 기반 커뮤니케이션 프로토콜입니다. 공개 또는 비공개로 실시간 메시징을 제공합니다. 사용자는 쪽지를 주고받고 그룹 채널에 참여할 수 있습니다.

IRC 클라이언트는 채널에 참여하기 위해 IRC 서버에 연결합니다. IRC 서버는 서로 연결되어 네트워크를 형성합니다.

2장 일반 규칙

- 프로그램은 어떤 상황에서도(메모리가 부족한 경우에도) 충돌해서는 안 되며, 예기치 않게 종료되어서는 안 됩니다.
이 경우 프로젝트는 작동하지 않는 것으로 간주되며 성적은 다음과 같이 표시됩니다.
0.
- 소스 파일을 컴파일할 메이크파일을 제출해야 합니다. 다시 링크해서는 안 됩니다.
- 메이크파일에는 최소한 규칙이 포함되어야 합니다:
(NAME), all, clean, fclean, re.
- c++와 -Wall -Wextra -Werror 플래그를 사용하여 코드를 컴파일합니다.
- 코드는 **C++ 98 표준**을 준수해야 합니다. 그런 다음 -std=c++98 플래그를 추가하면 여전히 컴파일됩니다.
- 항상 가능한 한 많은 C++ 기능을 사용하여 개발하도록 노력하세요(예를 들어 <string.h>보다 <cstring>). C 함수를 사용할 수 있지만 가능하면 항상 C++ 버전을 선호합니다.
- 외부 라이브러리 및 부스트 라이브러리는 금지되어 있습니다.

제3장 필수 부분

프로그램 이름	ircserv
파일 제출	메이크파일, *.h, hpp}, *.cpp, *.hpp, *.ipp, 선택적 구성 파일
메이크파일	이름, 모두, 깨끗하다, 깨끗하다, 다시
인수	포트입니다: 수신 포트 비밀번호 : 연결 비밀번호
외부 함수.	C++ 98의 모든 것. 소켓, 닫기, 설정소켓옵트, getsockname, getprotobyname, 게호스트바이네임, getaddrinfo, freeaddrinfo, 바인드, 연결, 청취, 수락, htons, htonl, ntohs, ntohl, inet_addr, inet_ntoa, 보내기, 수신, 신호, lseek, fstat, fcntl, 폴 (또는 등가)
Libft 인증	n/a
설명	C++ 98의 IRC 서버

C++ 98로 IRC 서버를 개발해야 합니다.

클라이언트를 개발해서는 안 됩니다.

서버 간 통신을 처리해서는 안 됩니다.

실행 파일은 다음과 같이 실행됩니다:

```
./ircserv <포트> <암호>
```

- 포트: 포트: IRC 서버가 들어오는 IRC 연결을 수신 대기할 포트 번호입니다.
- 비밀번호: 연결 비밀번호입니다. 서버에 연결을 시도하는 모든 IRC 클라이언트가 이 비밀번호를 입력해야 합니다.



주제와 평가 척도에 poll()이 언급되어 있더라도 select(), kqueue() 또는 epoll()과 같은 동등한 함수를 사용할 수 있습니다.

III.1 요구 사항

- 서버는 동시에 여러 클라이언트를 처리할 수 있어야 하며 중단되지 않아야 합니다.
- 포킹은 허용되지 않습니다. 모든 I/O 작업은 차단되지 않아야 합니다.
- 이러한 모든 작업(읽기, 쓰기뿐만 아니라 듣기 등)을 처리하는 데는 하나의 poll()(또는 이에 상응하는 함수)만 사용할 수 있습니다.



비차단 파일 기술자를 사용해야 하므로 poll()(또는 이에 상응하는 함수)없이 읽기/받기 또는 쓰기/보내기 함수를 사용할 수 있으며 서버가 차단되지 않습니다.
하지만 시스템 리소스를 더 많이 소비하게 됩니다.
따라서 poll()(또는 이에 상응하는 함수)를 사용하지 않고 파일 설명자로 읽

- 여러 IRC 클라이언트가 존재합니다. 그 중 하나를 **참조** 클라이언트로 선택해야 합니다. 참조 클라이언트는 평가 프로세스 중에 사용됩니다.
- 참조 클라이언트는 오류 없이 서버에 연결할 수 있어야 합니다.
- 클라이언트와 서버 간의 통신은 TCP/IP(v4 또는 v6)를 통해 이루어져야 합니다.
- 참조 클라이언트를 서버에서 사용하는 것은 공식 IRC 서버에서 사용하는 것과 비슷해야 합니다. 하지만 다음 기능만 구현하면 됩니다:
 - 참조 클라이언트를 사용하여 인증, 닉네임, 사용자 아이디 설정, 채널 참여, 비공개 메시지 송수신 등의 기능을 사용할 수 있어야 합니다.
 - 한 클라이언트에서 채널로 보내는 모든 메시지는 채널에 참여한 다른 모든 클라이언트에게 전달되어야 합니다.
 - 운영자와 일반 사용자가 있어야 합니다.
 - 그런 다음 채널 운영자 전용 명령을 구현해야 합니다:
 - * KICK - 채널에서 클라이언트 내보내기
 - * 초대 - 고객을 채널에 초대합니다.
 - * 토픽 - 채널 토픽 변경 또는 보기
 - * 모드 - 채널의 모드를 변경합니다:
 - i: 초대 전용 채널 설정/제거하기
 - t: TOPIC 명령의 채널 운영자 제한을 설정/해제합니다.
 - k: 채널 키(비밀번호) 설정/해제
 - o: 채널 운영자 권한 부여/취소

팅

- I: 채널에 대한 사용자 제한 설정/해제
- 물론 깔끔한 코드를 작성해야 합니다.

III.2 MacOS 전용



MacOS는 다른 유닉스 OS와 같은 방식으로 `write()`를 구현하지 않으므로 `fcntl()`을 사용할 수 있습니다.

다른 유닉스 OS와 유사한 동작을 얻으려면 비차단 모드에서 파일 기술자를 사용해야 합니다.



그러나 다음과 같은 경우에만 `fcntl()`을 사용할 수 있습니다:

`fcntl(fd, F_SETFL, O_NONBLOCK);`

다른 것들은 금지되어 있습니다.

III.3 테스트 예제

가능한 모든 오류와 문제(부분 데이터 수신, 낮은 대역폭 등)를 완벽하게 확인합니다.

서버로 보내는 모든 내용을 올바르게 처리하는지 확인하기 위해 다음과 같이 `nc`를 사용하여 간단한 테스트를 수행할 수 있습니다:

```
\$> NC 127.0.0.1 6667
com^Dman^Dd
\$>
```

'com', 'man', 'd\n' 등 여러 부분으로 나누어 명령을 보내려면 `ctrl+D`를 사용합니다.

명령을 처리하려면 먼저 수신된 패킷을 집계하여 다시 빌드해야 합니다.

4장 보너스 부분

다음은 IRC 서버에 추가할 수 있는 추가 기능으로 실제 IRC 서버와 더욱 비슷하게 보이도록 할 수 있습니다:

- 파일 전송을 처리합니다.
- 봇입니다.



보너스 부분은 필수 부분이 완벽한 경우에만 평가됩니다. 완벽하다는 것은 필수 부분을 완벽하게 완료하고 오작동 없이 작동한다는 의미입니다. 모든 필수 요건을 통과하지 못한 경우 보너스 부분은 전혀 평가되지 않습니다.

5장

제출 및 동료 평가

과제는 평소처럼 Git 저장소에 제출하세요. 방어 기간 동안에는 저장소 내의 작업만 평가됩니다. 주저하지 말고 파일 이름을 다시 한 번 확인하여 정확한지 확인하세요.

제출하지 않고 채점하지 않더라도 프로젝트에 대한 테스트 프로그램을 만드는 것이 좋습니다. 이러한 테스트는 특히 방어 중에 서버를 테스트하는 데 유용할 수 있지만, 언젠가 다른 `ft_irc`를 평가해야 하는 경우 동료의 서버를 테스트하는 데도 유용할 수 있습니다. 실제로 평가 과정에서 필요한 테스트는 무엇이든 자유롭게 사용할 수 있습니다.



평가 과정에서 참조 클라이언트가 사용됩니다.

