



ft_irc

インターネット・リレー・チャット

概要

このプロジェクトは、あなた自身のIRCサーバーを作るためのものです。

実際のIRCクライアントを使ってサーバーに接続し、テストする。

インターネットは、接続されたコンピューターが相互に作用することを可能にする

強固な標準プロトコルによって支配されている。

知っておくに越したことはない。

バージョン: 8

内容

I	はじめに	2
II	一般規定	3
III	必須パート	4
III.1	必要条件	5
III.2	For MacOS only	6
III.3	テスト例	6
IV	ボーナス・パート	7
V	提出と相互評価	8

第一章 はじめに

インターネット・リレー・チャットまたはIRCは、インターネット上のテキストベースの通信プロトコルです。公開または非公開のリアルタイム・メッセージングを提供する。ユーザーは直接メッセージを交換したり、グループチャンネルに参加することができます。

IRCクライアントはチャンネルに参加するためにIRCサーバーに接続する。IRCサーバーはネットワークを形成するために接続されます。

第II章 一般規定

- あなたのプログラムは、どんな状況でも（たとえメモリー不足になっても）クラッシュしてはならないし、予期せず終了してはならない。
その場合、あなたのプロジェクトは機能しないとみなされ、あなたの成績は次のようになります。
0.
- ソースファイルをコンパイルするMakefileを提出しなければならない。再リンクしてはならない。
- Makefileには少なくともルールが含まれていなければならない：
(NAME)、all、clean、fclean、re。
- c++と-Wall -Wextra -Werrorフラグでコードをコンパイルする。
- あなたのコードはC++ 98標準に準拠していなければならない。それなら、-std=c++98フラグを追加してもコンパイルできるはずだ。
- 常に、できる限りC++の機能を使用して開発するようにしてください（たとえば、次のように選択します）。
<cstring> over <string.h>）。C関数を使用することは許されているが、可能であれば常にC++版を使用すること。
- 外部ライブラリやBoostライブラリの使用は禁止されている。

第III章 必須パー

ト

プログラム名	イールサーバ
ファイルの提出	Makefile, *.{h, hpp}, *.cpp, *.hpp, *.ipp、 オプションの設定ファイル
メイクファイル	名前、すべて、クリーン、Fクリーン、再
論争	ポート: リスニングポート パスワード: 接続パスワード
外部機能。	すべてはC++ 98で。 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 (またはそ れに相当するもの)
リフト公認	該当なし
説明	C++ 98 による IRC サーバー

C++98でIRCサーバを開発しなければならない。ク

ライアントを開発してはならない。

サーバ間の通信を処理してはならない。

実行ファイルは次のように実行される:

`./ircserv <ポート> <パスワード>`

- ポート番号: IRCサーバがIRC接続をリッスンするポート番号。
- password: 接続パスワード。あなたのサーバに接続しようとするIRCクライアントが必要

とします。



`poll()`が主題と評価尺度で言及されていても、`select()`、`kqueue()`、`epoll()`などの同等のものを使用することができる。

III.1 必要条件

- サーバーは複数のクライアントを同時に処理でき、決してハングアップしないものでなければならない。
- フォークは許されない。すべてのI/O操作は**ノンブロッキング**でなければならない。
- **1つの** poll() (または同等のもの) だけが、これらすべての操作 (読み込み、書き込みだけでなく、リッスンなど) を処理するために使用できる。



ノンブロッキングのファイル・ディスクリプタを使用する必要があるため、poll() (または同等のもの) を使用せずにread/recvまたはwrite/send関数を使用することが可能であり、サーバーがブロッキングすることはありません。しかし、より多くのシステムリソースを消費することになる。

したがって、poll() (または同等のもの) を使用せずにファイル記述子で読み取り/再読み取りや書き込み/送信を行おうとすると、成績は0となる。

- IRCクライアントはいくつか存在する。そのうちの1つを**リファレンス**として選択する必要があります。あなたのリファレンスクライアントは評価プロセスで使用されます。
- リファレンス・クライアントは、エラーが発生することなくサーバーに接続できなければなりません。
- クライアントとサーバー間の通信は、TCP/IP (v4またはv6) を介して行われなければならない。
- あなたのリファレンス・クライアントをあなたのサーバーで使用することは、公式IRCサーバーで使用するのと同様でなければなりません。ただし、以下の機能だけは実装する必要があります:
 - リファレンスクライアントを使用して、認証、ニックネーム、ユーザー名の設定、チャンネルへの参加、プライベートメッセージの送受信ができなければなりません。
 - あるクライアントからチャンネルに送信されたすべてのメッセージは、チャンネルに参加した他のすべてのクライアントに転送されなければならない。

- オペレーターや常連ユーザーがいなければならない。
- それから、チャンネル・オペレーター特有のコマンドを実装しなければならない：
 - * KICK - クライアントをチャンネルから退出させます。
 - * INVITE - クライアントをチャンネルに招待します。
 - * TOPIC - チャンネルのトピックの変更と表示
 - * MODE - チャンネルのモードを変更します：
 - i: 招待専用チャンネルの設定/削除
 - t: チャンネルオペレーターに対するTOPICコマンドの制限を設定/解除する
 - k: チャンネルキー（パスワード）の設定/解除
 - o: チャンネル・オペレータの権限を与える／奪う

- l:チャンネルへのユーザー制限の設定/解除

- もちろん、きれいなコードを書くことが求められる。

III.2 MacOSのみ



MacOSは他のUnix OSと同じようにwrite()を実装していないので、fcntl()を使うことが許されている。

他のUnix OSと同様の動作を得るためには、ノンブロッキング・モードでファイル・ディスクリプタを使用しなければならない。



fcntl(fd, F_SETFL, O_NONBLOCK);
それ以外の旗は禁止されている。

III.3 テスト例

ありとあらゆるエラーや問題（部分的なデータの受信、低帯域幅など）を検証する。

あなたのサーバーが、あなたがサーバーに送信したものすべてを正しく処理していることを確認するには、ncを使用して次のような簡単なテストを行うことができます：

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

ctrl+Dを使って、'com'、'man'、'dn'の順にコマンドを送る。

コマンドを処理するためには、まず受信したパケットを集約して再構築する必要がある。

第IV章 ボーナス

・パート

ここでは、IRCサーバーに追加できる機能を紹介します：

- ・ ファイル転送を処理する。
- ・ ボットだ。



ボーナス・パートは、必須パートがパーフェクトである場合にのみ査定される。パーフェクトとは、必須パートが統合的に行われ、不具合なく動作することを意味します。必須条件をすべてクリアしていない場合、ボーナス・パートはまったく評価されません。

第五章

提出と相互評価

通常通り、課題を Git リポジトリに提出してください。あなたのリポジトリ内の作業のみが、ディフェンス中に評価されます。ファイル名が正しいかどうか、ためらわずにダブルチェックしてください。

提出されず採点もされませんが、プロジェクトのテストプログラムを作成することが推奨されます。これらのテストは、特に防衛中に自分のサーバをテストするのに役立つだけでなく、いつか他のft_ircを評価しなければならなくなったときに、仲間のサーバをテストするのにも役立ちます。実際、評価の過程で必要なテストは何でも自由に使って構いません。



あなたのリファレンスクライアントは、評価の過程で使用されます。



16D85ACC441674FBA2DF65190663F432222F81AA0248081A7C1C1823F7A96F0B74495
15056E97427E5B22F07132659EC8D88B574BD62C94BB654D5835AAD889B014E078705
709F6E02

