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、独自の IRC サーバーを作成することに関するものです。
実際の IRC クライアントを使用してサーバーに接続し、テストします。
インターネットは、接続されたコンピューターが相互にやり取りできるようにする堅牢な標準プロトコルによって制御されています。

知っておくのは常に良いことです。

バージョン: 8

コンテンツ

第1章

導入

インターネット リレー チャット(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します。ユーザーはダイレクト メッセージを交換したり、グループ
チャンネルに参加したりできます。

IRCクライアントはチャンネルに参加するためにIRCサーバーに接続します。IRCサーバーは接続されています
一緒にネットワークを形成します。

第2章

一般的なルール

- プログラムはいかなる状況でもクラッシュしてはいけません（たとえ
  メモリ) を消費し、予期せず終了することはありません。
  そのような事態が発生した場合、プロジェクトは機能していないとみなされ、成績は 0 になります。

- ソースファイルをコンパイルするためのMakefileを提出する必要があります。
  再リンクします。

- Makefile には少なくとも以下のルールが含まれている必要があります。
  $(NAME)、all、clean、fclean、re。

- コードをC++でコンパイルし、フラグ -Wall -Wextra -Werror を付ける

- コードはC++ 98標準に準拠している必要があります。そうすれば、コンパイルは問題なく実行できます。
  フラグ -std=c++98 を追加した場合。

- 常にできる限り多くの C++ 機能を使用して開発するようにしてください (たとえば、<string.h> ではなく <cstring>
  を選択します)。C 関数を使用することもできますが、可能な場合は常に C++ バージョンを優先してください。

- 外部ライブラリおよび Boost ライブラリは禁止されています。

第3章

必須部分

プログラム名	ircserv
ファイルを提出する	Makefile、*.{h, hpp}、*.cpp、*.tpp、*.ipp、オプションの設定ファイル NAME、all、clean、 fclean、re port: リスニングポート password: 接続パスワード
メイクファイル	C++ 98 のすべて。 socket、close、setsockopt、
引数	getsockname、getprotobyname、gethostbyname、 getaddrinfo、freeaddrinfo、bind、connect、listen、accept、htons、htonl、
外部関数。	ntohs、ntohl、inet_addr、inet_ntoa、send、 recv、signal、sigaction、lseek、fstat、fcntl、poll (または同等のもの) n/a C++ 98 の IRC サーバー
Libft 認定	
説明	

C++ 98 で IRC サーバーを開発する必要があります。

クライアントを開発してはいけません。
サーバー間通信を処理してはいけません。

実行ファイルは次のように実行されます。

./ircserv <ポート> <パスワード>

- ポート: IRCサーバーが受信を待機するポート番号
  IRC 接続。
- パスワード: 接続パスワード。サーバーに接続しようとするすべての IRC クライアントで必要になります。

件名と評価スケールに poll() が記載されている場合でも、select()、kqueue()、epoll() などの同等の関数を使用できます。

III.1 要件

- サーバーは複数のクライアントを同時に処理でき、
  下がる。
- フォークは許可されません。すべての I/O 操作は非ブロッキングである必要があります。
- これらすべての操作 (読み取り、書き込み、リッスンなど) を処理するために使用できるのは、 1 つのpoll() (または同等のもの) のみです。

非ブロッキング ファイル記述子を使用する必要があるため、poll() (または同等のもの) なしで read/recv または write/send 関数を使用することが可能であり、サーバーはブロックされません。

ただし、システムリソースの消費量が増えます。
したがって、poll() (または同等のもの) を使用せずに任意のファイル記述子で読み取り/受信または書き込み/送信を行おうとすると、グレードは 0 になります。

- IRCクライアントは複数存在します。そのうちの1つを参照として選択する必要があります。
  評価プロセスでは参照クライアントが使用されます。
- 参照クライアントは、サーバーに接続する際に問題に遭遇することなく、
  エラー。
- クライアントとサーバー間の通信は TCP/IP (v4 または v6) 経由で行う必要があります。
- リファレンス クライアントをサーバーで使用する方法は、公式 IRC サーバーで使用する方法と同様である必要があります。ただし、実装する必要があるのは、次の機能だけです。
  - 認証、ニックネームやユーザー名の設定、チャンネルへの参加、
    参照クライアントを使用してプライベート メッセージを送受信します。
  - 1つのクライアントからチャネルに送信されたすべてのメッセージは、
    チャネルに参加した他のすべてのクライアント。
  - オペレーターと通常のユーザーが必要です。
  - 次に、チャネル固有のコマンドを実装する必要があります
    演算子:

    KICK - チャネルからクライアントを排除する

    INVITE - クライアントをチャネルに招待する

    トピック - チャンネルのトピックを変更または表示する

    MODE - チャンネルのモードを変更する: i: 招待の
    ・
    みのチャンネルを設定/削除 ・ t: チャンネルへ

    のTOPICコマンドの制限を設定/削除
    オペレーター

    ・ k: チャンネルキー（パスワード）の設定/削除 ・ o: チャンネ

    ルオペレータ権限の付与/取得

- l: チャンネルのユーザー制限を設定/削除する

- もちろん、きれいなコードを書くことが求められます。

III.2 MacOSのみ

MacOS は他の Unix OS と同じように write() を実装していないため、fcntl() を使用できます。

非ブロッキングモードでファイル記述子を使用する必要があります。
他の Unix OS と同様の動作です。

ただし、fcntl() は次のようにのみ使用できます: fcntl(fd, F_SETFL, O_NONBLOCK); その他のフラグは
禁止されています。

III.3 テスト例

考えられるすべてのエラーと問題（部分的なデータの受信、低帯域幅など）を徹底的に検証します。

サーバーが送信したすべての情報を正しく処理できるようにするには、
nc を使用した次の簡単なテストを実行できます。

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

Ctrl+D を使用して、コマンドを複数の部分に分けて送信します (「com」、「man」、「d\n」の順)。

コマンドを処理するには、まず受信したパケットを集約する必要があります。
再建するために。

第4章

ボーナス部分

IRC サーバーに追加できる追加機能は次のとおりです。これにより、IRC サーバーが実際の IRC サーバーにさらに似たものになります。

- ファイル転送を処理します。
- ボット。

ボーナス部分は、必須部分が完璧である場合にのみ評価されます。完璧とは、必須部分が完全に実行され、誤動作することなく機能することを意味します。必須要件のすべてに合格していない場合、ボーナス部分はまったく評価されません。

第5章

提出とピア評価

いつものように Git リポジトリに課題を提出してください。審査中はリポジトリ内の作業のみが評価されます。ファイル名が正しいことを確認するために、躊躇せずに再確認してください。

提出されず、採点もされないとしても、プロジェクトのテスト プログラムを作成することは推奨されます。これらのテストは、防衛中に自分のサーバーをテストするのに特に役立ちますが、いつか別の ft_irc を評価する必要があるときに同僚のサーバーをテストするのにも役立ちます。実際、評価プロセス中に必要なテストは自由に使用できます。

評価プロセスでは参照クライアントが使用されます。

16D85ACC441674FBA2DF65190663F432222F81AA0248081A7C1C1823F7A96F0B74495
15056E97427E5B22F07132659EC8D88B574BD62C94BB654D5835AAD889B014E078705
709F6E02