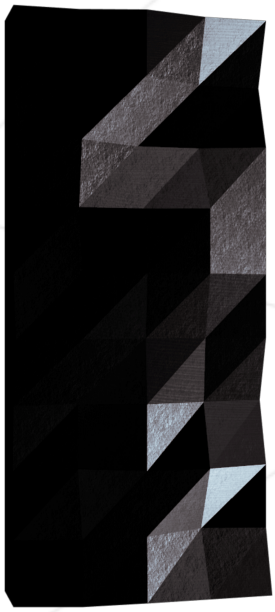

ft_irc

Интернет-чат-реле

Краткое содержание:

Этот проект посвящен созданию собственного IRC-сервера.

Для подключения к серверу и его тестирования вам понадобится настоящий IRC-клиент.

Интернет управляется надежными стандартными протоколами, которые позволяют подключенным компьютерам взаимодействовать

друг с другом.

Это всегда полезно знать.

Версия: 8

Содержание

Глава I

Введение

Интернет-чат-релеили IRC — это текстовый протокол общения в Интернете. Он предлагает обмен сообщениями в реальном времени, который может быть как публичным, так и приватным. Пользователи могут обмениваться прямыми сообщениями и присоединяться к групповым каналам.

Клиенты IRC подключаются к серверам IRC для присоединения к каналам. Серверы IRC соединены вместе, образуя сеть.

Глава 2

Общие правила

- Ваша программа не должна аварийно завершать работу ни при каких обстоятельствах (даже если заканчивается память) и не должна неожиданно завершаться.
  Если это произойдет, ваш проект будет считаться нефункциональным, а ваша оценка будет снижена. 0.

- Вам нужно сдать вMakefileкоторый скомпилирует ваши исходные файлы. Он не должен перелинковываться.
- ТвойMakefileдолжен как минимум содержать
  правила: $(ИМЯ), все, очистить, fcleanире.
- Скомпилируйте свой код с помощьюс++и флаги -Стена -Векстра -Веррор
- Ваш код должен соответствовать**Стандарт C++ 98**. Тогда он все равно должен скомпилироваться, если вы добавите флаг -стандарт=c++98.
- Старайтесь всегда разрабатывать, используя самыеC++функции, которые вы можете (например, выбрать <cstring>более <строка.h>).Вам разрешено использоватьCфункции, но всегда предпочитают ихC++версии, если возможно.
- Любая внешняя библиотека иСпособствовать ростуБиблиотеки запрещены.

Глава 3

Обязательная часть

Название программы	ircserv
Сдайте файлы	Makefile, *.{h, hpp}, *.cpp, *.tpp, *.ipp, необязательный файл конфигурации
Makefile	ИМЯ, все, чистый, fclean, re
Аргументы	порт: Порт прослушивания пароль: Пароль подключения
Внешние функции.	Все на C++ 98. сокет, закрыть, setsockopt, getsockname, getprotobyname, gethostbyname, getaddrinfo, freeaddrinfo, привязать, подключить, прослушать, принять, htons, htonl, ntohs, ntohl, inet_addr, inet_ntoa, отправить, получить, сигнал, sigaction, lseek, fstat, fcntl, опрос (или эквивалент)
Libft авторизован	н/д
Описание	IRC-сервер на C++ 98

Вам необходимо разработать IRC-сервер на C++ 98.

Ты**не должен**развивать клиента.
Ты**не должен**управлять взаимодействием сервер-сервер.

Ваш исполняемый файл будет запущен следующим образом:

. /ircserv <порт> <пароль>

- порт:Номер порта, на котором ваш IRC-сервер будет прослушивать входящие IRC-соединения.
- пароль:Пароль подключения. Он понадобится любому IRC-клиенту, который попытается подключиться к вашему серверу.

Даже если poll() упоминается в теме и шкале оценки, вы можете использовать любой эквивалент, например select(), kqueue() или epoll().

III.1 Требования

- Сервер должен иметь возможность обрабатывать несколько клиентов одновременно и никогда не зависать.
- Форкинг не допускается. Все операции ввода-вывода должны быть**неблокирующий**.
- Только**1**опрос() (или эквивалент) может использоваться для обработки всех этих операций (чтение, запись, а также прослушивание и т. д.).

Поскольку вам приходится использовать неблокирующие файловые дескрипторы, можно использовать функции чтения/приема или записи/отправки без poll() (или эквивалента), и ваш сервер не будет блокироваться.
Но это потребляло бы больше системных ресурсов.
Таким образом, если вы попытаетесь прочитать/получить или записать/отправить любой файловый дескриптор без использования poll() (или эквивалента), ваша оценка будет равна 0.

- Существует несколько клиентов IRC. Вам нужно выбрать один из них в качестве**ссылка**. Ваш референс-клиент будет использован в процессе оценки.
- Ваш клиент-образец должен иметь возможность подключаться к вашему серверу без возникновения каких-либо ошибок.
- Связь между клиентом и сервером должна осуществляться черезTCP/IP (v4 или v6).
- Использование вашего референсного клиента с вашим сервером должно быть похоже на использование его с любым официальным сервером IRC. Однако вам нужно реализовать только следующие функции:
  - Вы должны иметь возможность проходить аутентификацию, устанавливать псевдоним, имя пользователя, присоединяться к каналу, отправлять и получать личные сообщения с помощью вашего реферального клиента.
  - Все сообщения, отправленные одним клиентом на канал, должны быть пересланы каждому другому клиенту, присоединившемуся к каналу.
  - У вас должно бытьоператорыи постоянные пользователи.
  - Затем вам необходимо реализовать команды, специфичные дляоператоры каналов:

    * KICK - Исключить клиента из канала * INVITE -

    Пригласить клиента на канал * TOPIC -

    Изменить или просмотреть тему канала

    * MODE - Изменить режим канала:

    · i: Установить/удалить канал только по приглашению

    · t: Установить/снять ограничения команды TOPIC для операторов канала

    · k: Установить/удалить ключ канала (пароль)

    · o: Дать/забрать привилегию оператора канала

· l: Установить/снять ограничение пользователей на канал

- Конечно, от вас ждут написания чистого кода.

III.2 Только для MacOS

Поскольку MacOS не реализует write() так же, как другие ОС Unix, вам разрешено использовать fcntl().
Чтобы получить поведение, аналогичное поведению других ОС Unix, необходимо использовать файловые дескрипторы в неблокируемом режиме.

Однако вам разрешено использовать fcntl() только следующим образом: fcntl(fd, F_SETFL, O_NONBLOCK);
Любой другой флаг запрещён.

III.3 Тестовый пример

Проверьте абсолютно все возможные ошибки и проблемы (получение частичных данных, низкая пропускная способность и т. д.).

Чтобы убедиться, что ваш сервер правильно обрабатывает все, что вы ему отправляете, выполните следующий простой тест с использованиемнкможно сделать:

```
\$> нк -C 127.0.0.1 6667
com^Dman^Dd
\$>
```

Использоватьctrl+Dотправить команду несколькими частями: 'ком',затем 'мужчина',затем 'д\н'.

Чтобы обработать команду, необходимо сначала объединить полученные пакеты, чтобы восстановить ее.

Глава IV

Бонусная часть

Вот дополнительные функции, которые вы можете добавить к своему IRC-серверу, чтобы он стал еще более похожим на настоящий IRC-сервер:

- Управление передачей файлов.
- Бот.

Бонусная часть будет оцениваться только в том случае, если обязательная часть выполнена ИДЕАЛЬНО. Идеально означает, что обязательная часть выполнена полностью и работает без сбоев. Если вы не выполнили ВСЕ обязательные требования, ваша бонусная часть вообще не будет оцениваться.

Глава V

Представление и экспертная оценка

Сдайте задание в вашемГитрепозиторий как обычно. Только работа внутри вашего репозитория будет оцениваться во время защиты. Не стесняйтесь дважды проверять имена ваших файлов, чтобы убедиться, что они верны.

Вам рекомендуется создавать тестовые программы для вашего проекта, даже если они**не будет отправлен и не будет оценен**. Эти тесты могут быть особенно полезны для проверки вашего сервера во время защиты, а также вашего сверстника, если вам нужно оценить другогоft_irc один день. Действительно, вы вольны использовать любые тесты, которые вам нужны в процессе оценки.

Ваш референтный клиент будет использован в процессе оценки.

16D85ACC441674FBA2DF65190663F432222F81AA0248081A7C1C1823F7A96F0B74495
15056E97427E5B22F07132659EC8D88B574BD62C94BB654D5835AAD889B014E078705
709F6E02