Отчёт по курсовому проекту

(Спасенов Иван Владимирович, Доктор Артем Алексеевич)

7 февраля 2022 г.

Оглавление

Введение

Серверная часть SMTP агента

Задание. Вариант 10

Используется вызов pselect и единственный рабочий поток. Журналирование в отдельном процессе. Нужно проверять обратную зону днс.

Цель и задачи

Цель: Разработать **SMTP-сервер** с использованием одного потока и метода pselect(). Задачи:

– проанализировать **SMTP**-протокол и разработать конечный автомат обработки SMTP-сообщений;

– реализовать программу для получения и сохранения писем по протоколу **SMTP** на языке программирования **C**;

– оформить расчетно-пояснительную записку.

Клиентская часть SMTP агента

Задание. Вариант

Глава 1

Аналитический раздел

1.1 Предметная область

1.1.1 ER-диаграмма предметной области

В результате проведенного исследования были выявлены следующие сущности предметной области:

1. Клиент.
2. Сервер.
3. Логгер.
4. Письмо.
5. Отправитель.
6. Получатель.
7. Данные письма.

Зависимость между сущностями предметной области может быть описана ER-диаграммой (1.1).

Рис. 1.1: ER-диаграмма предметной области

1.2 Достоинства и недостатки реализуемой архитектуры

1.2.1 Серверная часть SMTP агента

Согласно условию задачи, в работе сервера предлагается использовать один поток выполнения и один отдельный поток журналирования.

Достоинства варианта реализации:

- простота реализации, отсутствует необходимость реализации разделяемой памяти и взаимодействия между процессами или потоками;
- отсутствие времени на переключение контекстов;
- благодаря неблокирующему вводу/выводу, сервер может обслуживать множество клиентов с достаточно высокой производительностью, при условии, что обработка занимает мало времени;
- логирование в отдельном процессе позволяет не блокироваться на операциях ввода/вывода при записи в файл или в терминал;

Недостатки данной архитектуры:

- низкая производительность при длительной обработке клиентских команд;
- низкая отказоустойчивость (использование одного потока является менее надежным при возникновении фатальных ошибок в приложении, чем при наличии нескольких взаимозаменяемых потоков,);
- сложность масштабирования и использования всех аппаратных ресурсов системы.

Недостатки программной реализации с одним потоком выполнения и мультиплексированием можно уменьшить с помощью создания нескольких (пула) потоков с неблокирующим вводом/выводом и распределения нагрузки между ними.

1.2.2 Клиентская часть SMTP агента

Глава 2

Конструкторский раздел

2.1 Конечный автомат состояний сервера

На рис. 2.1 представлен сгенерированный с использованием *fsm2dot* скрипта из *autogen* файла конфигурации конечного автомата *serverfsm.def* и *dot*.

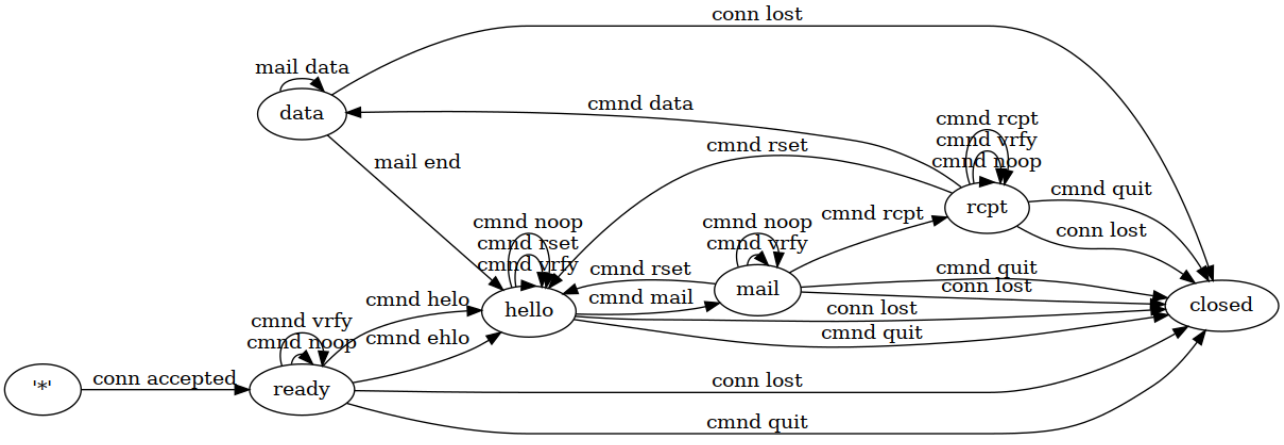

Рис. 2.1: Построенный граф конечного автомата SMTP сервера

2.2 Синтаксис команд протокола

Ниже приведен формат команд сообщений протокола в виде регулярных выражений: Регулярные выражения SMTP команд:

NOOP `[Nn][Oo][Oo][Pp]\\r\\n`

HELO `[Hh][Ee][Ll][Oo]\\s*(?<domain>.+)\\r\\n`

EHLO `[Ee][Hh][Ll][Oo]\\s*(?<domain>.+)\\r\\n`

MAIL `[Mm][Aa][Ii][Ll] [Ff][Rr][Oo][Mm]:\\s*<(?<address>.+@.+)?>\\r\\n`

RCPT `[Rr][Cc][Pp][Tt] [Tt][Oo]:\\s*<(?<address>.+@.+)>\\r\\n`

VRFY `[Vv][Rr][Ff][Yy]\\s*(?<domain>.+)\\r\\n`

DATA `[Dd][Aa][Tt][Aa]\\r\\n`

RSET `[Rr][Ss][Ee][Tt]\\r\\n`

QUIT `[Qq][Uu][Ii][Tt]\\r\\n`

Окончание данных письма `^\\.\\r\\n`

2.3 Синтаксис команд протокола

На рис. 2.2 и на рис. 2.3 представлены физическая и логическая диаграммы представления данных в системе соответственно.

Конфигурация *serveropts.def*, используемая для автоматической генерации исходного кода обработки входных флагов приложенияс помощью *autogen.*:

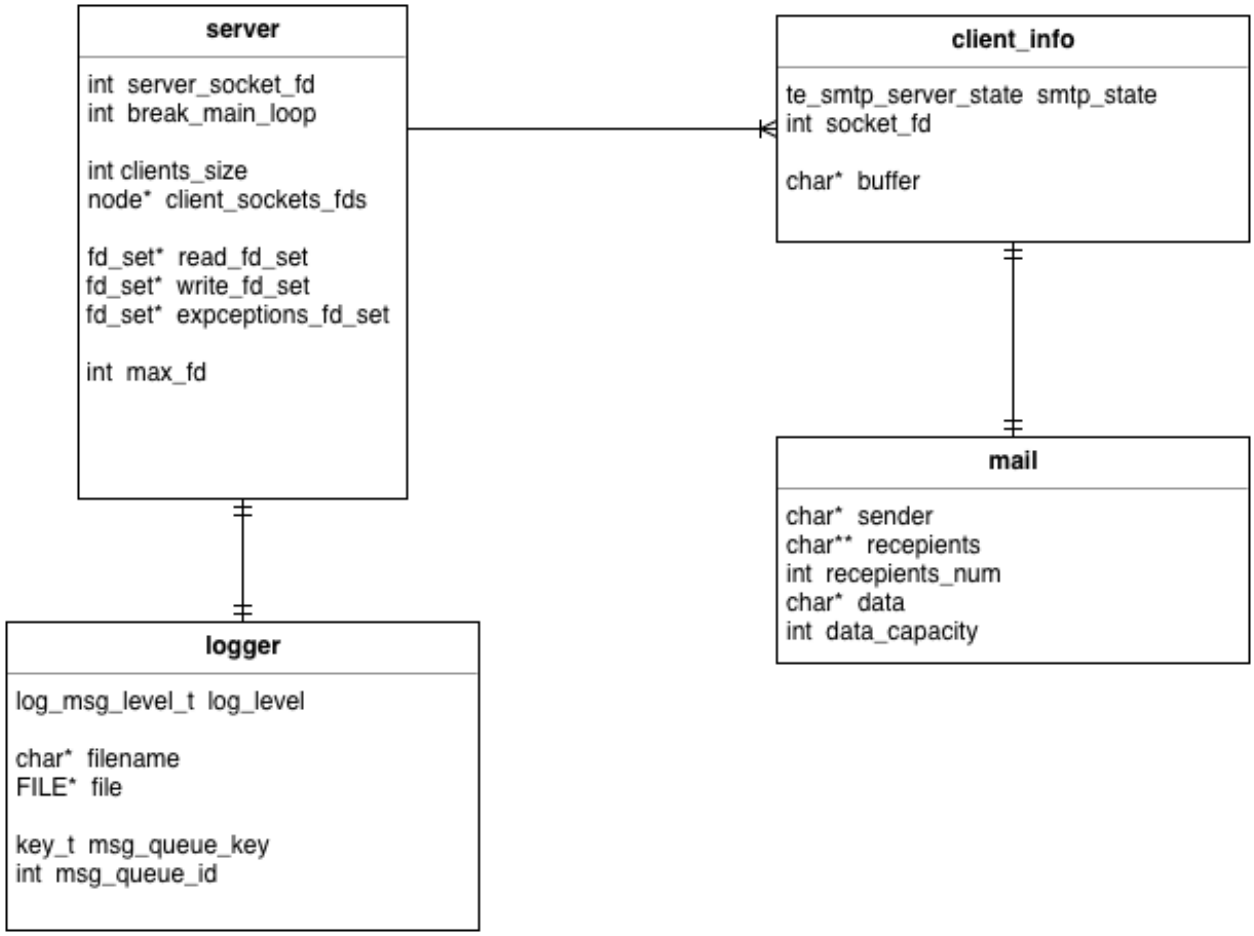

Рис. 2.2: Физическая диаграмма представления данных в серверной части системы

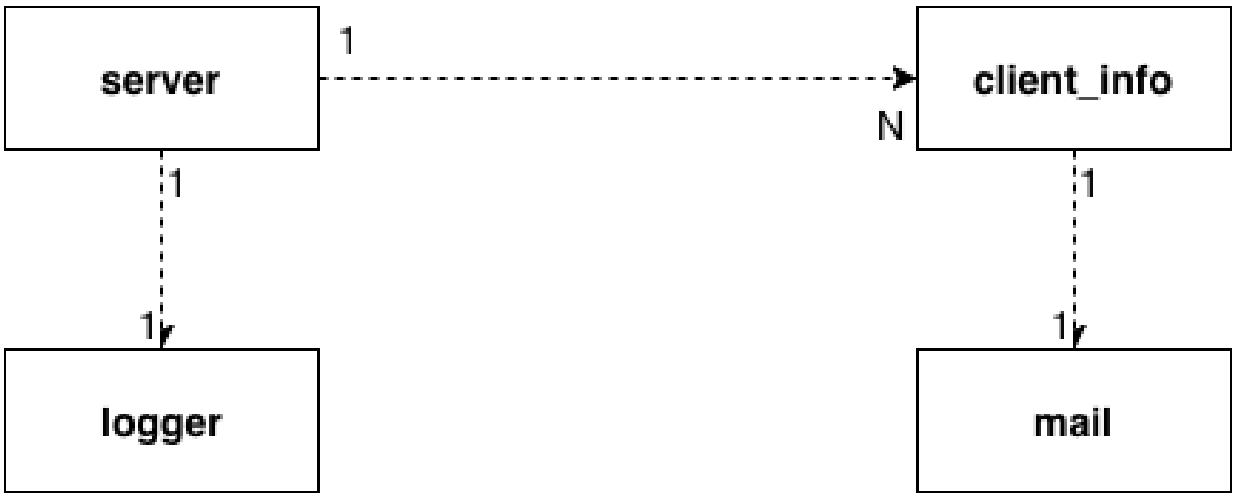

Рис. 2.3: Логическая диаграмма представления данных в серверной части системы

Глава 3

Технологический раздел

3.1 Сборка программы

Сборка SMTP агента состоит из трех *Makefile* системы сборки *make*:

– сборка клиента,

– сборка сервера,

– сборка отчета.

3.1.1 Сборка серверной части SMTP агента

Сборка SMTP сервера состоит из следующих целей:

1. сборка сервера;
2. сборка тестового клиента;
3. генерация исходных кодов конечного автомата и опций с помощью *autogen*;
4. сборка сервера и тестового клиента;
5. запуск системного тестирования.

Сборка программы осуществляется с помощью следующей команды:

```
make autogen_all && make all
```

Выводы

3.2 Серверная часть SMTP агента

В результате выполнения курсового проекта была достугнута поставленная цель, а именно разработан **SMTP-сервер** с использованием одного потока и метода pselect(), осуществляющее прием и сохранение писем для дальнейшей поставки их пользователям.

Во время выполнения работы были выполнены следующие задачи:

- проанализировано архитектурное решение, данное по условиям задачи, определены его преимущества и недостатки;
- разработан и реализован подход для обработки входящих соединений на основе метода pselect();
- разработан и реализовано хранение входящих писем в каталоге maildir;
- проанализирован протокол **SMTP** и реализован конечный автомат обработки входящих SMTP-сообщений;

А также получены и закреплены следующие навыки:

- проектирование и реализация сетевого протокола SMTP;
- реализация серверного приложения с несколькими процессами на языке программирования Си;
- создание сценариев сборки программного обеспечения;
- использование latex и сценариев сборки для автогенерации расчетно-пояснительной записки.

В ходе работы не были реализованы следующие пункты, планируемые к разработке в дальнейшем:

- проверка обратной зоны DNS;
- генерация документации и графов функци с помощью утилит;
- использование внешних конфигурационных файлов;
- модульное тестирование;