

Отчёт по курсовому проекту

(Пучнина Анастасия Ивановна, Козлова Юлия Алексеевна)

19 января 2019 г.

Оглавление

Введение	1
0.1 Серверная часть SMTP агента	2
0.1.1 Задание. Вариант 10	2
0.1.2 Цель и задачи	2
0.2 Клиентская часть SMTP агента	2
0.2.1 Задание. Вариант 31	2
1 Аналитический раздел	3
2 Конструкторский раздел	6
2.1 Конечный автомат состояний сервера	6
2.2 Синтаксис команд протокола	6
2.3 Синтаксис команд протокола	8
3 Технологический раздел	11
3.1 Сборка программы	11
3.1.1 Сборка серверной части SMTP агента	11
Выводы	11
3.2 Серверная часть SMTP агента	12

Введение

0.1 Серверная часть SMTP агента

0.1.1 Задание. Вариант 10

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

0.1.2 Цель и задачи

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

- проанализировать архитектурное решение;
- разработать подход для обработки входящих соединений;
- разработать подход хранения входящих писем в `maildir`;
- проанализировать **SMTP**-протокол и разработать конечный автомат обработки SMTP-сообщений;
- реализовать программу для получения и сохранения писем по протоколу **SMTP**.

0.2 Клиентская часть SMTP агента

0.2.1 Задание. Вариант 31

Используется вызов `select` и единственный рабочий поток (или процесс). Журналирование в отдельном процессе. Пытаться отправлять все сообщения для одного MX за одну сессию.

....

Глава 1

Аналитический раздел

Предметная область

ER-диаграмма предметной области

В результате проведенного исследования были выявлены следующие сущности предметной области:

1. Отправитель.
2. Получатель.
3. Пользователь.
4. Электронное письмо.
5. Файл письма.
6. Каталог Maildir.

Зависимость между сущностями предметной области может быть описана ER-диаграммой (1.1). Диаграмма выполнена в соответствии с нотацией UML.

Достоинства и недостатки реализуемой архитектуры

Серверная часть SMTP агента

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

Достоинства данного 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;

../images/er_diagram.png

Рис. 1.1: ER-диаграмма предметной области

- 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.

Клиентская часть SMTP агента

....

Глава 2

Конструкторский раздел

2.1 Конечный автомат состояний сервера

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

Данный рисунок не отображает переходы между состояниями при использовании именнованных обработчиков, поэтому, для корректного отображения конечного автомата, был создан специальный файл конфигурации *serverfsm_report.def autogentype.. ??*.

2.2 Синтаксис команд протокола

*Ниже приведен формат команд сообщений протокола в виде регулярных выражений:
Общие регулярные выражения:*

Перевод строки `\r\textbackslashn`

Пробелы `\s*`

Доменное имя `<?<domain>.+>`

Электронный адрес или пустые скобки `<?<address>.+@.+>/<>`

Электронный адрес `<?<address>.+@.+>`

Регулярные выражения SMTP команд:

NOOP `^NOOP`

HELO `^HELO:`

EHLO `^EHLO:`

MAIL `^MAIL FROM:`

RCPT `^RCPT TO:`

VERFY `^VERFY:`

DATA `^DATA`

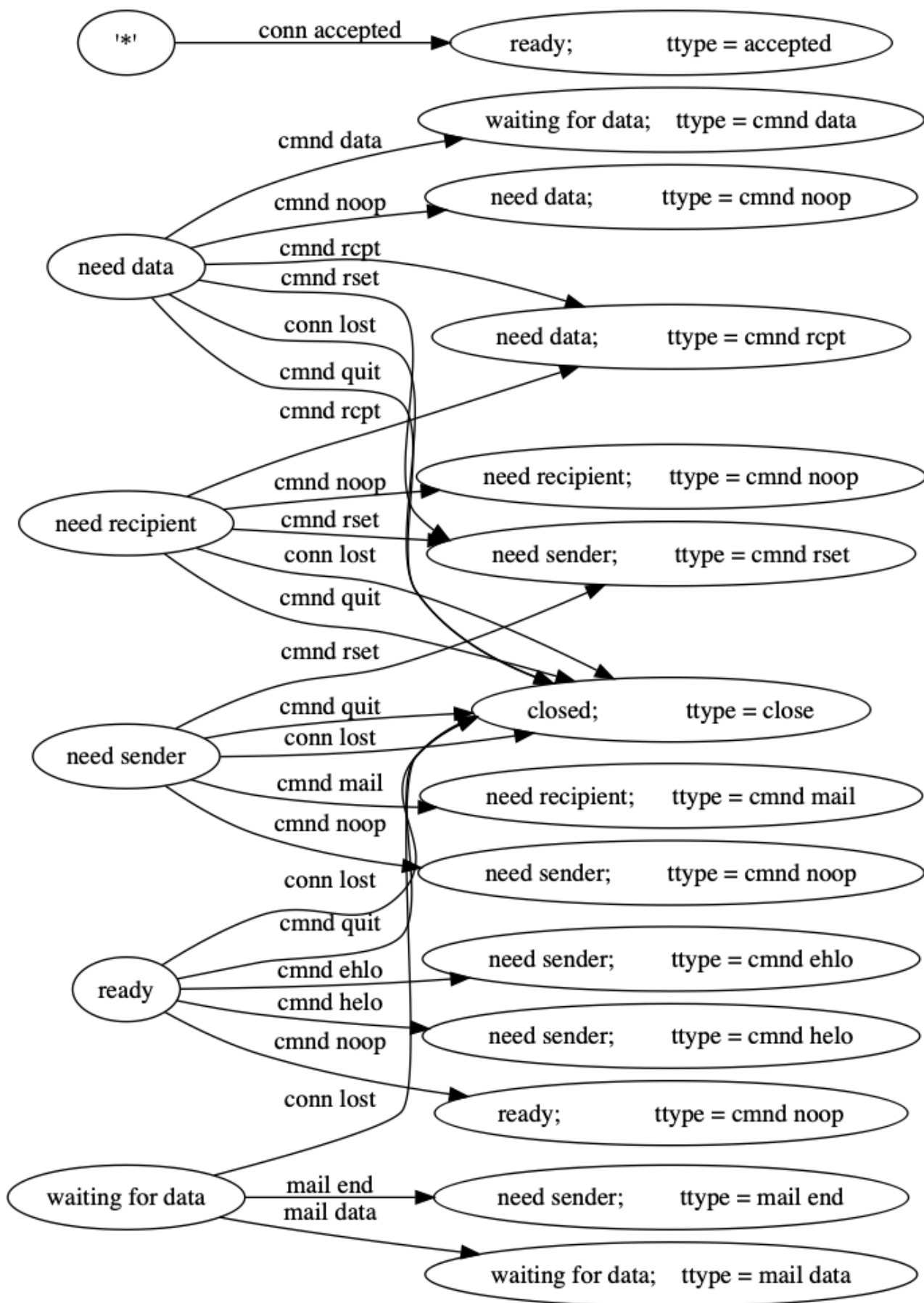


Рис. 2.1: Построенный граф конечного автомата SMTP сервера с использованием именованных обработчиков

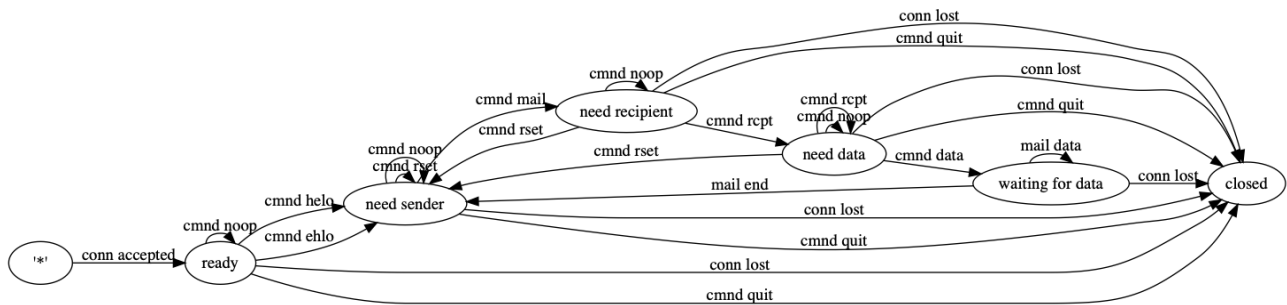


Рис. 2.2: Граф состояний конечного автомата SMTP сервера

RSET $\sim RSET$

QUIT $\sim QUIT$

Регулярные выражения содержимого письма:

Данные письма $[\backslash x00-\backslash x7F]^+$

Окончание данных письма $\sim \backslash$.

2.3 Синтаксис команд протокола

На рис. 2.3 и на рис. 2.4 представлены физическая и логическая диаграммы представления данных в системе соответственно.

Конфигурация `serveropts.def`, используемая для автоматической генерации исходного кода обработки входных флагов приложения с помощью `autogen`:

`language=C ../../source/server/autogen/serveropts.def`

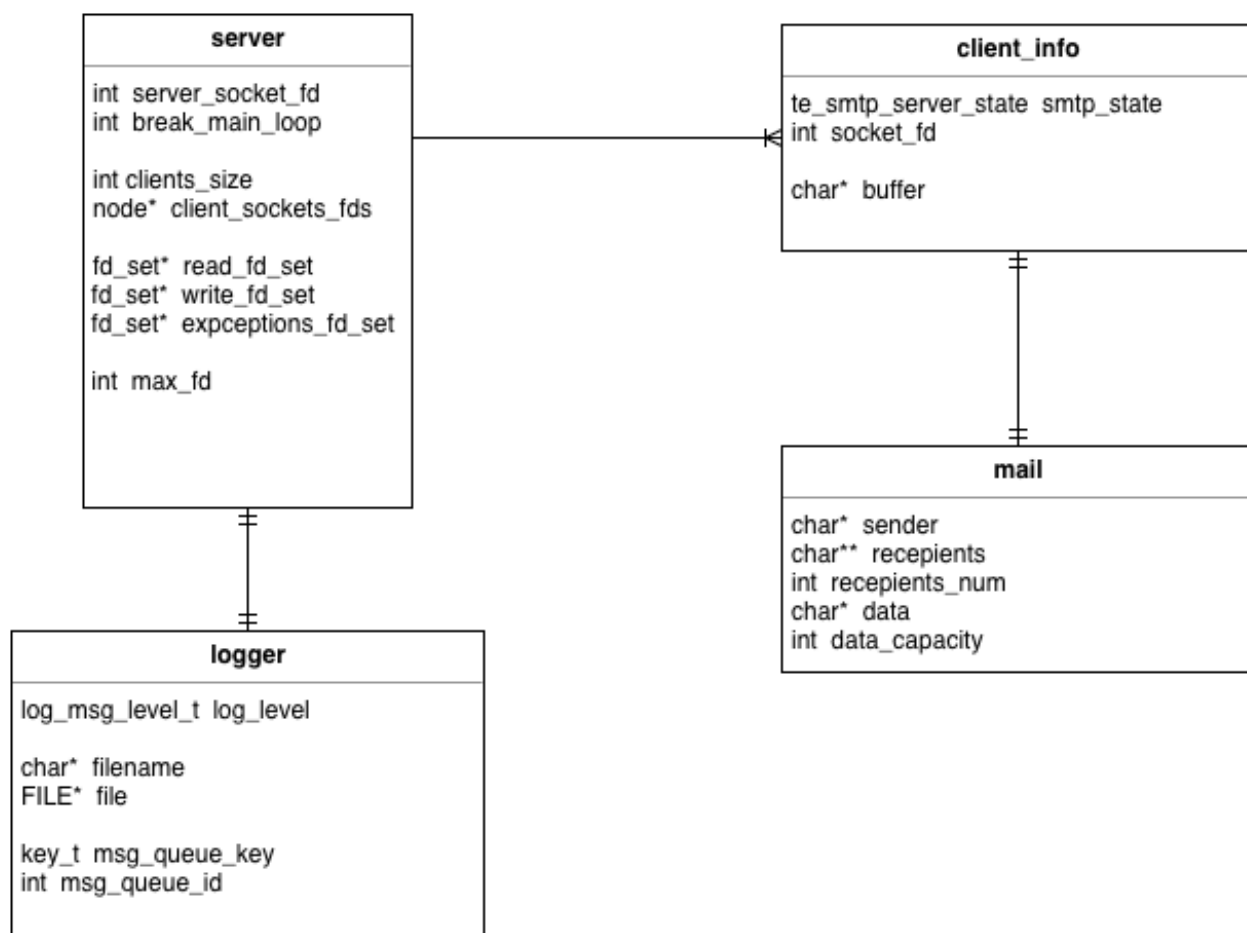


Рис. 2.3: Физическая диаграмма представления данных в серверной части системы

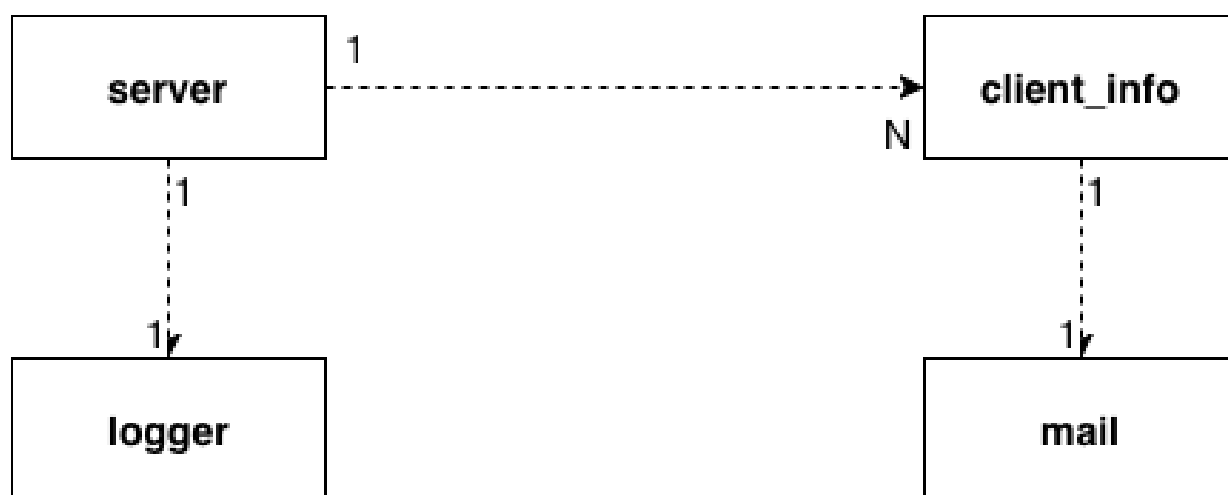


Рис. 2.4: Логическая диаграмма представления данных в серверной части системы

Глава 3

Технологический раздел

3.1 Сборка программы

Сборка SMTP агента состоит из трех Makefile системы сборки take:

- 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 достигнута 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 `C++`;
- 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 функции с помощью утилит;
- использование внешних конфигурационных файлов;
- модульное тестирование;