

# Отчёт по курсовому проекту

(Федоров Павел Вячеславович)

11 января 2019 г.

# Оглавление

<b>Введение</b>	<b>1</b>
<b>1 Аналитический раздел</b>	<b>3</b>
<b>2 Конструкторский раздел</b>	<b>4</b>
2.1 Конечный автомат состояний сервера . . . . .	4
2.2 Синтаксис команд протокола . . . . .	4
<b>3 Технологический раздел</b>	<b>6</b>
3.1 Сборка программы . . . . .	6
3.2 Основные функции программы . . . . .	6
3.3 Графы вызова функций . . . . .	6
<b>Выводы</b>	<b>6</b>

# Введение

## Сервер

## Клиент

### Задание. 26

Используется вызов `pselect` и рабочие потоки. Журналирование в отдельном потоки. Не обязательно пытаться отправлять все сообщения для одного `MX` за одну сессию.

### Цели и задачи

Цель: Разработать **SMTP-клиент** с использованием рабочих потоков и `pselect`.

Задачи:

- Проанализировать архитектурное решение `maildir`
- Разработать подход для обработки писем в `maildir`
- Рассмотреть **SMTP**-протокол
- Проанализировать способы получения и обработки **MX**-записей
- Реализовать программу для отправки писем по протоколу **SMTP**

# Глава 1

## Аналитический раздел

# Глава 2

## Конструкторский раздел

### 2.1 Конечный автомат состояний сервера

Рис. 2.1 нагенерил самодельный *fsm2dot* из *autogen* и *dot2tex* на пару *dot*. Никто не мешает изменить параметры типа *rankdir* прямо в *fsm2dot*, если он будет лучше смотреться, например, сверху-вниз.

### 2.2 Синтаксис команд протокола

**Команда выхода из сеанса**

**Команда передачи имени пользователя**

Для грамматики можно использовать вставку из файла и оформление `\begin{verbatim}` и `\end{verbatim}` или пакет *listings*<sup>1</sup>.

Для примера воспользуемся автоматической вставкой файла описания параметров программы (не забудьте перенести это в технологический раздел) через утилиту *src2tex*.

---

<sup>1</sup>На дворе XXI век, но пакет *listings* всё ещё не пашет с русскими комментариями без бубна, и лично я его пока не победил.

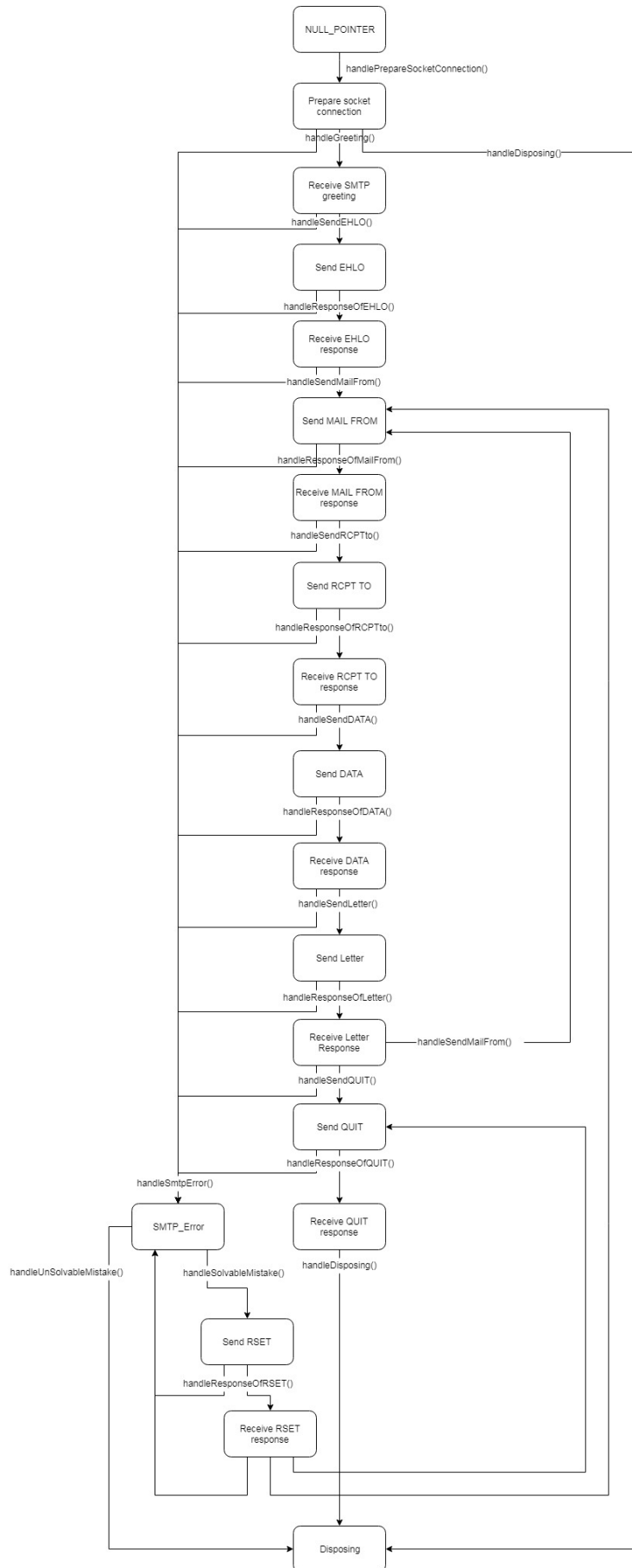


Рис. 2.1: Состояния сервера

## Глава 3

# Технологический раздел

Нужно отметить, что символ «`_`» необходимо оформлять как «`\_`».

### 3.1 Сборка программы

Сборка программы описана в файле *Makefile* системы сборки *make*. Рис. 3.1 нагенерили самодельные *makesimple* и *makefile2dot*, а также *dot2tex* и *dot*.

Отмечу, что за исключения целей типа *all*, *install*, *clean*, *tests*, все имена целей в файле систем сборки *make* обычно совпадают с именами файлов (такой вот низкоуровневый инструмент). То есть вместо цели *lexer* следует использовать цель *src/lexer.c*.

### 3.2 Основные функции программы

Весь этот раздел сгенерировал *doxygen* из части комментированных исходников программы. В файле конфигурации **doxyggen.cfg** был отключён параметр **HAVE\_DOT**, поскольку для рисования графов вызовов используется *cflow*.

### 3.3 Графы вызова функций

Поскольку функций много, графы вызовов разбиты на два рисунка. На рис. 3.2 показаны основные функции, на рис. 3.5 – функции обработки команд. Файл **cflow.ignore** содержит список функций (точнее, шаблонов поиска), используемых программой *grep* для удаления малоинтересных стандартных функций<sup>1</sup>.

Графы созданы с помощью *cflow*, *cflow2dot*, *dot*.

---

<sup>1</sup>Функции по работе с сокетами, *ipcs* и привилегиями к малоинтересным ни в коем случае не относятся.

Рис. 3.1: Сборка программы

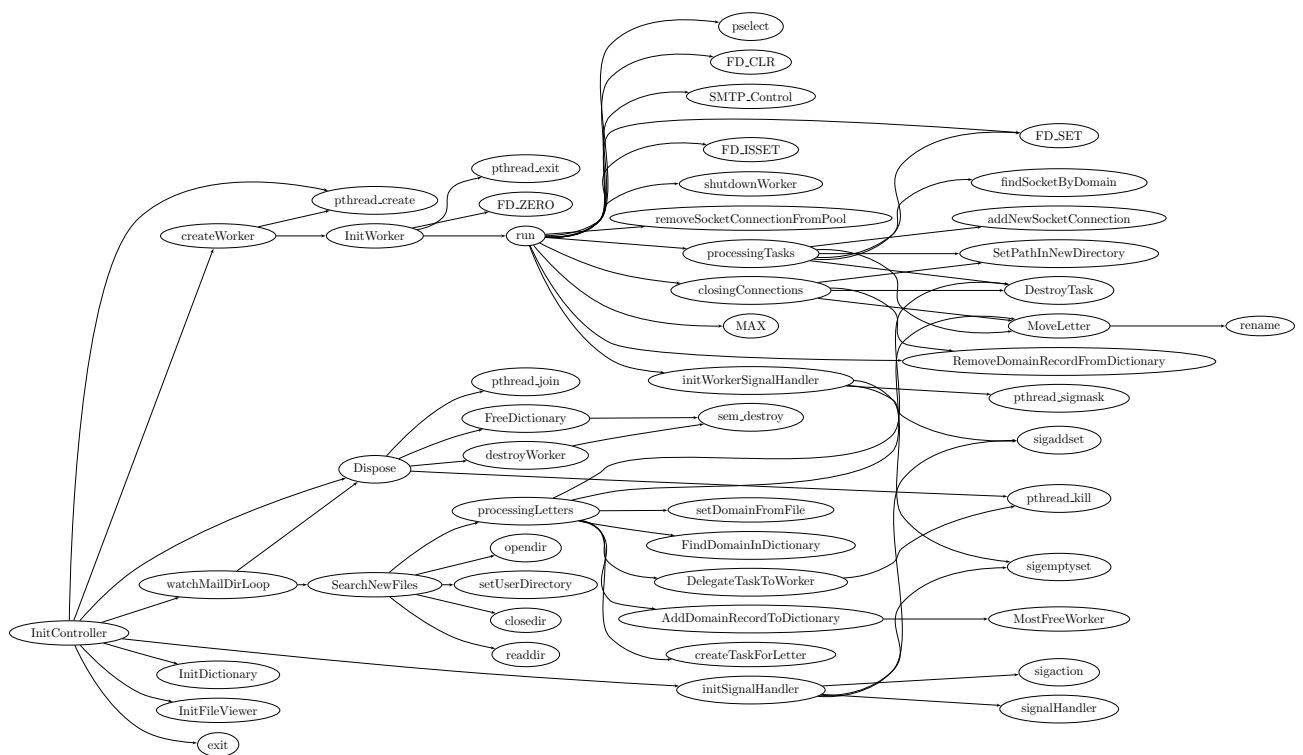


Рис. 3.2: Граф вызовов, основные функции



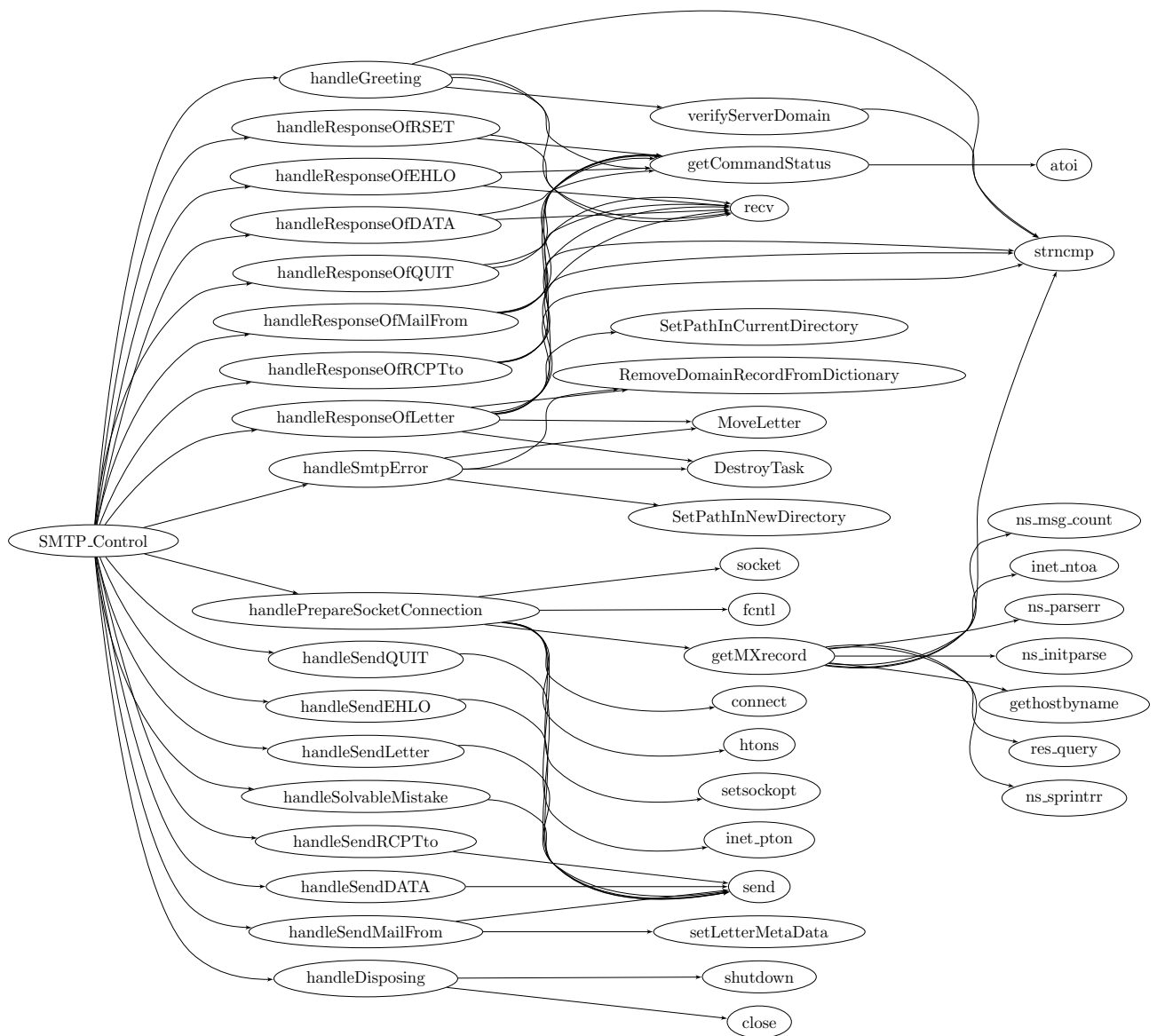


Рис. 3.3: Граф вызовов, основные функции

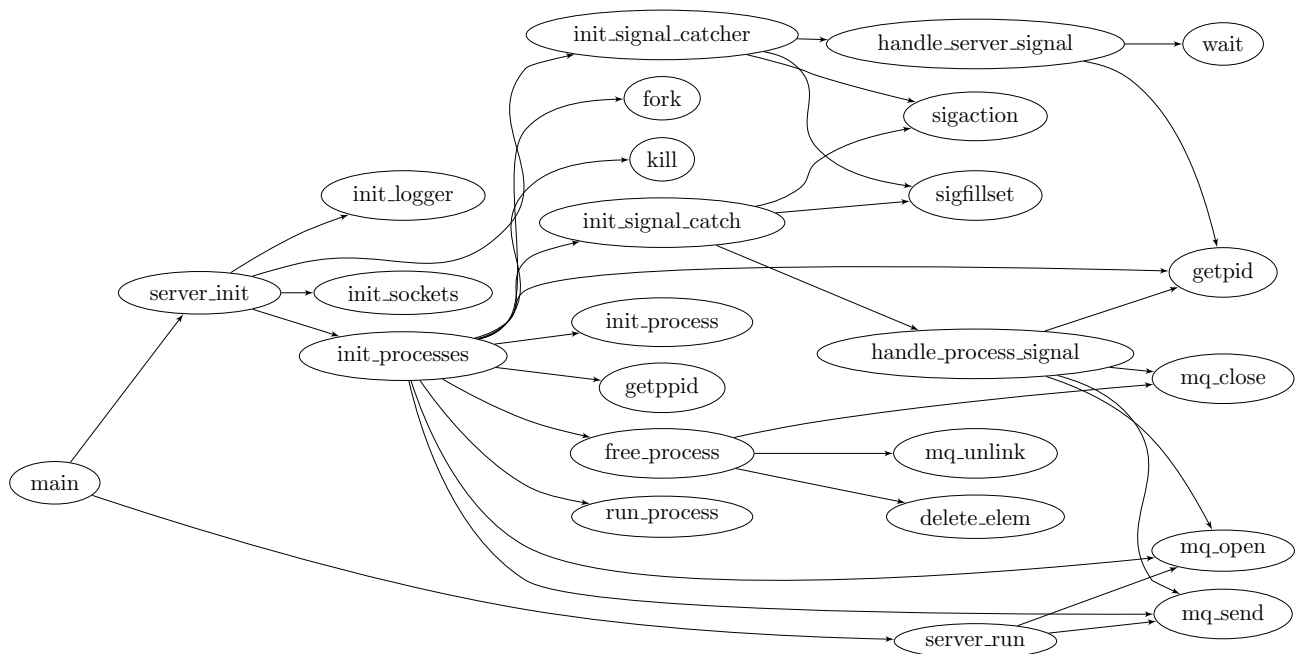


Рис. 3.4: Граф вызовов, функции обработки команд



Рис. 3.5: Граф вызовов, функции обработки команд

# Выводы

Что вы сделали и поняли.