

Отчёт по лабораторной работе №1

(Ф. И. О)

22 января 2016 г.

Оглавление

Введение	1
1 Конструкторский раздел	3
1.1 Конечный автомат состояний сервера	3
1.2 Синтаксис команд протокола	3
2 Технологический раздел	5
2.1 Сборка программы	5
2.2 Основные функции программы	5
2.3 Графы вызова функций	5
Выводы	5

Введение

Два предложения о содержании отчёта. Для нового абзаца в исходном тексте должна быть пустая строка.

Это – шаблон отчёта (вот как оформляется длинное тире, перед котрым идёт неразрывный пробел).

Здесь должно быть вербальное задание.

А вот так оформляются списки:

- элемент списка;
- последний элемент списка.

Нумерованный список выглядит следующим образом.

1. Первый элемент.
2. Второй элемент.

Глава 1

Конструкторский раздел

1.1 Конечный автомат состояний сервера

Рис. 1.1 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.

1.2 Синтаксис команд протокола

Команда выхода из сеанса `^\s*QUIT\s*\n$`

Команда передачи имени пользователя `^\s*USER:\s*\w+\s*\n$`

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

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

```
flag = {  
  name      = port; /* Порт, который слушает сервер */  
  value     = p;      /* Краткий флаг (-p) */  
  arg-type  = number;
```

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

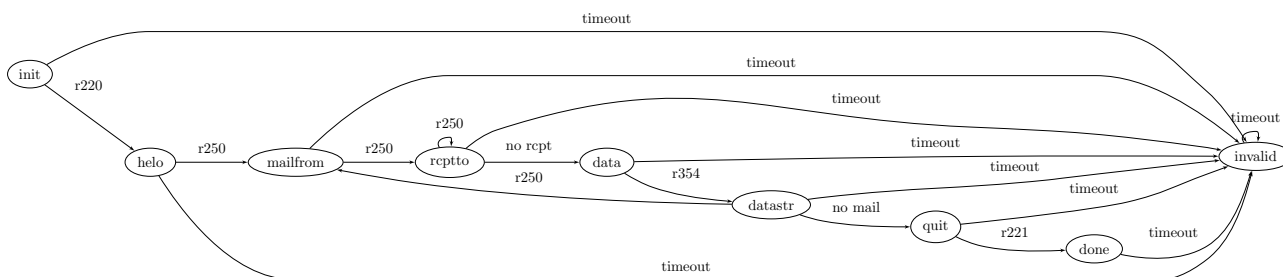


Рис. 1.1: Состояния сервера

```

    arg-range = 110;
    arg-range = "1024->65000";
    max      = 1;          /* Не более одного раза */
    min      = 1;          /* Обязательный параметр */
};
flag = {
    name      = file;      /* Файл с командами */
    value     = f;         /* Краткий флаг (-f) */
    arg-type  = string;
    max      = 1;          /* Не более одного раза */
    min      = 1;          /* Обязательный параметр */
};

```

Глава 2

Технологический раздел

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

2.1 Сборка программы

Сборка программы описана в файле *Makefile* системы сборки *make*. Рис. 2.1 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*.

2.2 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*.

2.3 Графы вызова функций

Поскольку функций много, графы вызовов разбиты на два рисунка. На рис. 2.2 показаны основные функции, на рис. 2.3 – функции обработки команд. Файл **cflow.ignore** содержит список функций (точнее, шаблонов поиска), используемых программой *grep* для удаления малоинтересных стандартных функций¹.

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

¹Функции по работе с сокетами, *irc* и привилегиями к малоинтересным ни в коем случае не относятся.

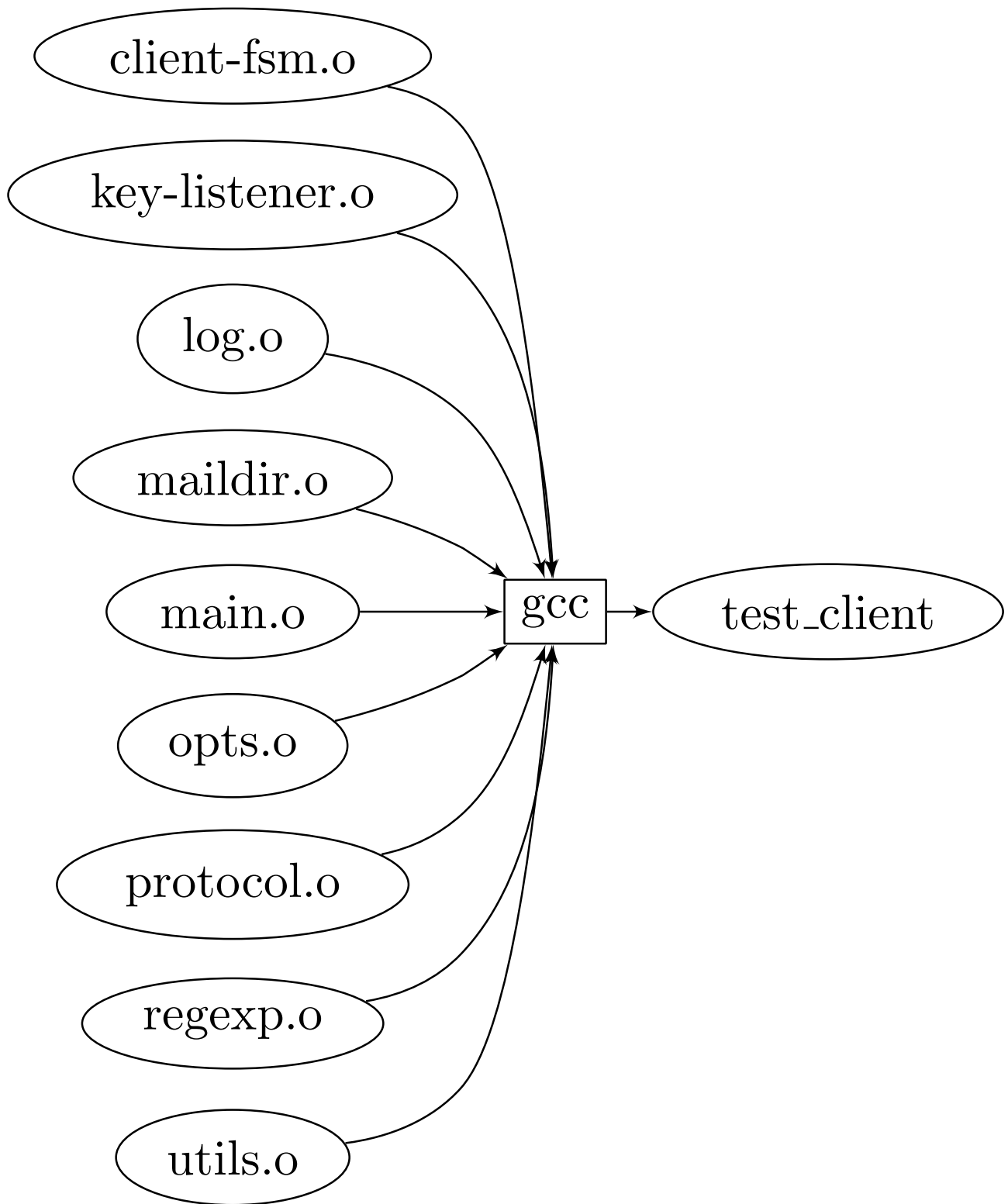


Рис. 2.1: Сборка программы

Рис. 2.2: Граф вызовов, основные функции

Рис. 2.3: Граф вызовов, функции обработки команд

Выводы

Что вы сделали и поняли.