

Содержание

Введение	4
1 Аналитический раздел	5
1.1 Структура почтовой системы	5
1.2 Типы МТА	7
1.3 Существующие подходы к созданию всячины	8
2 Конструкторский раздел	11
2.1 Архитектура всячины	11
2.2 Подсистема всякой ерунды	11
2.2.1 Блок-схема всякой ерунды	11
3 Технологический раздел	13
4 Экспериментальный раздел	15
Заключение	16
Список использованных источников	17
А Картинки	18
Б Еще картинки	19

Глоссарий

Распределённый — Слово, которое нельзя употреблять. Но надо протестировать длинные строки в глоссарии.

Обозначения и сокращения

— MUA — Mail User Agent

Введение

Почтовая программа (клиент электронной почты, почтовый клиент, мейл-клиент, мейлер) — программное обеспечение, устанавливаемое на компьютере пользователя и предназначенное для получения, написания, отправки и хранения сообщений электронной почты одного или нескольких пользователей (в случае, например, нескольких учётных записей на одном компьютере) или нескольких учётных записей одного пользователя. Их предназначением является предоставление удобного пользовательского интерфейса для работы с электронной почтой. В рамках целой почтовой системы почтовые клиенты являются лишь интерфейсом пользователя. Когда пользователь отправляет письмо через почтовый клиент, это не означает, что данное приложение непосредственно

Целью работы является создание всякой всячины. Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать существующую всячину;
- спроектировать свою, новую всячину;
- изготовить всякую всячину;
- проверить её работоспособность.

Вот так-то. А этот абзац вставлен для визуальной оценки отступа от перечня до следующего абзаца.

1 Аналитический раздел

В данном разделе анализируется и классифицируется существующая всячина и пути создания новой всячины. А вот отступ справа в 1 см. — это хоть и по ГОСТ, но ведь диагноз же...

1.1 Структура почтовой системы

Почтовая система в общем случае может быть представлена как совокупность следующих элементов:

- MUA — (англ. Mail User Agent) почтовый агент пользователя, почтовый клиент);
- MTA — (англ. Mail Transfer Agent) агент пересылки почты;
- MDA — (англ. Mail Delivery Agent) агент доставки почты.

MUA представляет собой программное обеспечение, устанавливаемое на компьютере пользователя и предназначенное для получения, написания, отправки и хранения сообщений электронной почты одного или нескольких пользователей (в случае, например, нескольких учётных записей на одном компьютере) или нескольких учётных записей одного пользователя. Предназначением является предоставление удобного пользовательского интерфейса для работы с электронной почтой.

MTA отвечает за отправку почты. MDA отвечает за доставку почты конечному пользователю. Почтовые сервера обычно выполняют роль MTA и MDA. Некоторые почтовые сервера (программы) выполняют роль как MTA, так и MDA, некоторые подразумевают разделение на два независимых сервера: сервер-MTA и сервер-MDA (при этом, если для доступа к почтовому ящику используются различные протоколы — например, POP3 и IMAP, — то MDA в свою очередь может быть реализован либо как единое приложение, либо как набор приложений, каждое из которых отвечает за отдельный протокол).

Только один MTA может функционировать на одной рабочей станции, так как исключительно одно приложение может быть назначено для получения входящих сообщений от других рабочих станций. Как правило, это привелигированная программа, которая прослушивает входящие TCP/IP соединения по SMTP-порту и имеет возможность сохранять данные в пользовательские почтовые ящики.

MTA способен принимать множество сообщений одновременно. Если в связи с непредвиденными обстоятельствами MTA не может доставить сообщение конечному пользователю, то посылается сообщение с причиной неудачной отправки. MTA хранит все сообщения, которые не могут быть своевременно отправлены конечному пользователю. Через определенные промежутки времени MTA инициирует повторную

отправку подобных сообщений. Чаще всего невозможность доставки почты связана с проблемами сетевого соединения и с отключением целевой рабочей станции.

С точки зрения МТА существуют два типа источников входящих сообщений: локальные процессы и другие рабочие станции. Также можно выделить три типа адресатов: локальные файлы, локальные процессы и другие рабочие станции.

Выделение агентов МТА и MUA означает, что они могут функционировать на различных рабочих станциях.

В верхней части рисунка ?? MUA, МТА и дисковое хранилище являются частью единой системы, которая выделена штриховой линией. Пользователи получают доступ к системе путем аутентификации и авторизации с помощью ввода логина и соответствующего пароля. MUA запускается с помощью пользовательской команды, как процесс операционной системы, и когда инициирует передачу сообщения МТА для последующей отправки, начинает взаимодействие с другим процессом операционной системы. MUA и МТА взаимодействуют с авторизованным пользователем, поэтому МТА обычно производит проверку того, что идентификационная информация пользователя включена в исходящее сообщение. Как описано в RFC 822, если в теле сообщения не указывается адресант From, то МТА обязан добавить запись Sender с указанием идентификационной информации отправителя.

МТА хранит сообщения в так называемой области спулинга. Под областью спулинга будем понимать дисковое хранилище, которое использует МТА для временного хранения очереди сообщений перед их отправкой.

Сообщения, которые предназначены другим удаленным рабочим станциям передаются по сети Интернет другим МТА с использованием протокола Simple Mail Transfer Protocol (SMTP). В том случае когда сервер-отправитель и сервер-получатель оба напрямую подключены к сети Интернет, сообщение может быть доставлено напрямую от отправителя к получателю. Однако иногда сообщению приходится преодолевать маршруты через промежуточные МТА. Большие организации часто организуют свои почтовые системы таким образом, что все входящие сообщения поступают на главный почтовый сервер. После этого сообщения пересылаются через другие серверы локальной сети. Когда сообщение доставляется получателю, МТА сохраняет его в почтовом ящике пользователя, который затем с помощью MUA получает доступ к своей почте.

Также промежуточные МТА используются в тех случаях, когда целевой почтовый сервер не доступен или сетевое соединение не может быть установлено. Преимущество такого подхода заключается в том, что сообщения аккумулируются на серверах максимально приближенных к целевому адресату и могут быть своевременно доставлены.

Нижняя часть рисунка иллюстрирует такую конфигурацию почтовой подсистемы получателя, когда MUA и МТА располагаются на различных рабочих станци-

ях. Такая конфигурация позволяет разделить процессы получения и отправки писем на отдельные операции. В процессе чтения почты пользователь напрямую взаимодействует с MUA, который использует протоколы POP3 (RFC 1939) или IMAP (RFC 2060) для получения доступа к почтовым ящикам пользователя и удаленным папкам на сервере. Для того чтобы осуществить подобные операции, пользователь должен быть авторизован в почтовой системе. Однако, протоколы POP3 и IMAP не содержат средств для пересылки сообщений. MUA подобного типа используют протокол SMTP для отправки сообщения MTA. Таким образом протокол SMTP, который изначально описывал передачу сообщений между MTA, в настоящее время используется также для транзита сообщений от MUA к MTA. Такое использование приводит к ряду проблем:

- MTA не может различить сообщения, поступающие от других MTA, от сообщений, предоставляемых MUA.
- Пользователь, отправляющий почту, не является установленным, то есть прошедшим авторизацию. В связи с этим MTA не предоставляется возможным определить является ли домен отправителя реально существующим.
- MUA может использовать различные сервера для отправки почты. Также специализированные MUA могут отправлять сообщения адресату напрямую через сеть интернет. Поэтому возможна ситуация, когда злоумышленники пытаются отправить поток нежелательной почты на произвольные сервера для ретрасляции.

На данный момент разработаны протоколы, которые способны решать описанные проблемы. Однако широкого применения они не получили.

1.2 Типы MTA

В простейшем случае отдельные рабочие станции или сервера в небольших офисах и домах, которые работают с несколькими почтовыми ящиками в одном домене, получают входящие сообщения от одного провайдера интернет услуг и передают сообщения провайдеру для последующей доставки адресату. Среди хостов, имеющих постоянное подключение к сети интернет, также имеются хосты, которые периодически подключаются к интернету для обмена входящей и исходящей почтой с сервером.

В [1] указано, что...

Кстати, про картинки. Во-первых, для фигур следует использовать `[ht]`. Если и после этого картинки вставляются «не по ГОСТ», т.е. слишком далеко от места ссылки, — значит у вас в РПЗ **слишком мало текста!** Хотя и ужасный параметр `!ht` у окружения `figure` тоже никто не отменял, только при его использовании документ получается страшный, как в ворде, поэтому просьба так не делать по возможности.

1.3 Существующие подходы к созданию всячины

Известны следующие подходы...

а) Перечисление с номерами.

б) Номера первого уровня. Да, ГОСТ требует именно так — сначала буквы, на втором уровне — цифры. Чуть ниже будет вариант «нормальной» нумерации и советы по её изменению. Да, мне так нравится: на первом уровне выравнивание элементов как у обычных абзацев. Проверим теперь вложенные списки.

1) Номера второго уровня.

2) Номера второго уровня. Проверяем на длиииииной-предлиииииииинной строке, что получается.... Сойдёт.

в) По мнению Лукьяненко, человеческий мозг старается подвести любую проблему к выбору из трех вариантов.

г) Четвёртый (и последний) элемент списка.

Теперь мы покажем, как изменить нумерацию на «нормальную», если вам этого захочется. Пара команд в начале документа поможет нам.

1) Изменим нумерацию на более привычную...

2) ... нарушим этим гост.

а) Но, пожалуй, так лучше.

В заключение покажем произвольные маркеры в списках. Для них нужен пакет **enumerate**.

1. Маркер с арабской цифрой и с точкой.

2. Маркер с арабской цифрой и с точкой.

I. Римская цифра с точкой.

II. Римская цифра с точкой.

В отчётах могут быть и таблицы — см. табл. 1.1 и 1.2. Небольшая таблица делается при помощи **tabular** внутри **table** (последний полностью аналогичен **figure**, но добавляет другую подпись).

Для больших таблиц следует использовать пакет **longtable**, позволяющий создавать таблицы на несколько страниц по ГОСТ.

Для того, чтобы длинный текст разбивался на много строк в пределах одной ячейки, надо в качестве ее формата задавать **p** и указывать явно ширину: в мм/дюймах (**110mm**), относительно ширины страницы (**0.22\textwidth**) и т.п.

Можно также использовать уменьшенный шрифт — но, пожалуйста, тогда уж во **всей** таблице сразу.

Таблица 1.2 — Пример длинной таблицы с длинным названием на много длинных-длинных строк

Вид шума	Громкость, дБ	Комментарий
Порог слышимости	0	
Шепот в тихой библиотеке	30	Конечно, это было до эпохи мобильных (внутри машины)
Обычный разговор	60-70	
Звонок телефона	80	
Уличный шум	85	
Гудок поезда	90	
Шум электрички	95	
Порог здоровой нормы	90-95	Длительное пребывание на более громком шуме может привести к ухудшению слуха
Мотоцикл	100	(модель бензокосилки) (Doom в целом вреден для здоровья)
Power Mower	107	
Бензопила	110	
Рок-концерт	115	
Порог боли	125	feel the pain
Клепальный молоток	125	(автор сам не знает, что это)
Порог опасности	140	Даже кратковременное пребывание на шуме большего уровня может привести к необратимым последствиям
Реактивный двигатель	140	Необратимое полное повреждение слуховых органов Интересно, почему?..
	180	
Самый громкий возможный звук	194	

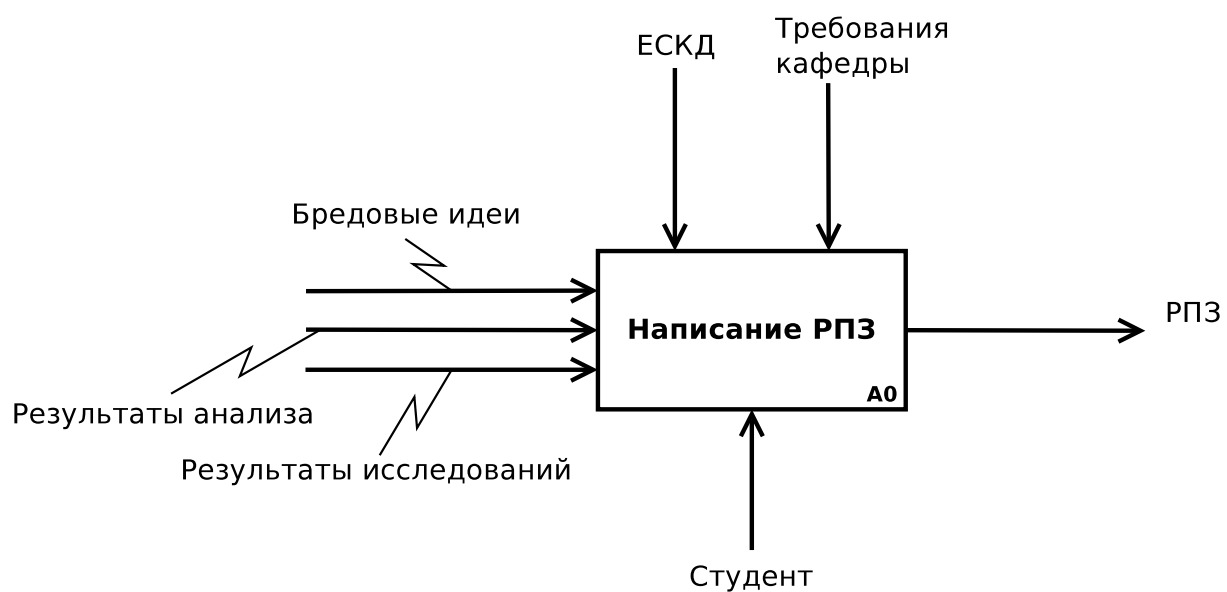


Рисунок 1.1 — Рисунок

Таблица 1.1 — Пример короткой таблицы с длинным названием на много длинных-длинных строк

Тело	F	V	E	$F + V - E - 2$
Тетраэдр	4	4	6	0
Куб	6	8	12	0
Октаэдр	8	6	12	0
Додекаэдр	20	12	30	0
Икосаэдр	12	20	30	0
Эйлер	666	9000	42	$+\infty$

2 Конструкторский раздел

В данном разделе проектируется новая всячина.

2.1 Архитектура всячины

Проверка параграфа. Вроде работает.

Вторая проверка параграфа. Опять работает.

Вот.

- Это список с «палочками».
- Хотя он и не по ГОСТ, кажется.

1) Поэтому для списка, начинающегося с заглавной буквы, лучше список с цифрами.

Формула 2.1 совершенно бессмысленна.

$$a = cb \tag{2.1}$$

Окружение `cases` опять работает (см. 2.2), спасибо И. Короткову за исправления..

$$a = \begin{cases} 3x + 5y + z, & \text{если хорошо} \\ 7x - 2y + 4z, & \text{если плохо} \\ -6x + 3y + 2z, & \text{если совсем плохо} \end{cases} \tag{2.2}$$

2.2 Подсистема всякой ерунды

Культурная вставка dot-файлов через утилиту `dot2tex` (рис. 2.1).

2.2.1 Блок-схема всякой ерунды

Кстати о заголовках

У нас есть и **subsubsection**. Только лучше её не нумеровать.

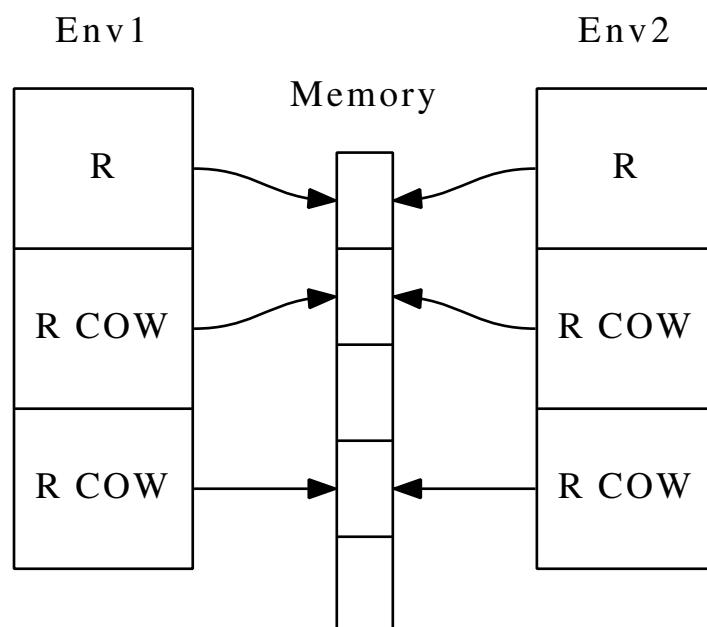


Рисунок 2.1 — Рисунок

3 Технологический раздел

В данном разделе описано изготовление и требование всячины. Благодаря пакет **underscore** эскейпить подчёркивание не нужно (**some_function**).

Для вставки кода есть пакет **listings**. К сожалению, пакет **listings** всё ещё работает криво при появлении в листинге русских букв и кодировке исходников utf-8. В данном примере он (увы) на лету конвертируется в koi-8 в ходе сборки pdf.

Есть альтернатива **listingsutf8**, однако она работает лишь с `\lstinputlisting`, но не с окружением **lstlisting**

Вот так можно вставлять псевдокод (питоноподобный язык определен в шаблоне):

Листинг 3.1 — Алгоритм оценки дипломных работ

```
1 def EvaluateDiplomas():
2     for each student in Masters:
3         student.Mark ← 5
4     for each student in Engineers:
5         if Good(student):
6             student.Mark ← 5
7         else:
8             student.Mark ← 4
```

Еще в шаблоне определен псевдоязык для BNF:

Листинг 3.2 — Грамматика

```
1 ifstmt → "if" "(" expression ")" stmt |
2         "if" "(" expression ")" stmt1 "else" stmt2
3 number → digit digit *
```

В листинге 3.3 работают русские буквы. Сильная магия. Однако, работает только во включаемых файлах, прямо в \TeX нельзя.

Листинг 3.3 — Пример (**test.c**)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Комментарий на русском с пробелами */
6     printf("Это строка с пробелами и русскими буквами");
7
8     return 0;
9 }
```

Можно также использовать окружение **verbatim**, если **listings** чем-то не устраивает. Только следует помнить, что табы в нём «съедаются». Существует также команда **verbatiminput** для вставки файла.

```
a_b = a + b; // русский комментарий
if (a_b > 0)
    a_b = 0;
```

4 Экспериментальный раздел

В данном разделе проводятся вычислительные эксперименты. А на рис. 4.1 показана схема мыслительного процесса автора...

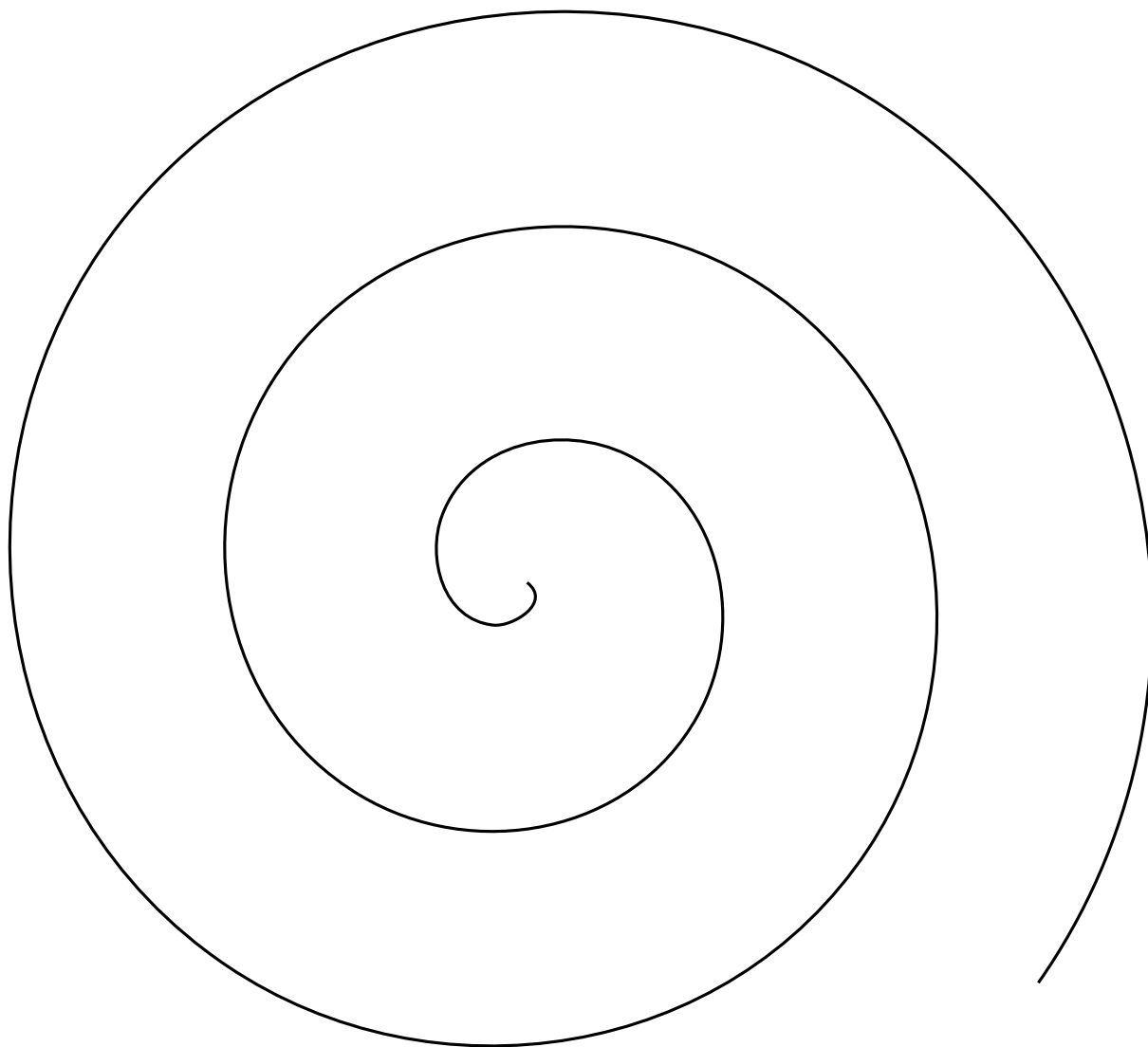


Рисунок 4.1 — Как страшно жить

Заключение

В результате проделанной работы стало ясно, что ничего не ясно...

Список использованных источников

1. *Пупкин, Василий*. L^AT_EX для «чайников» / Василий Пупкин, А. Эйнштейн.
— М., 2009.

Приложение А Картинки

Рисунок А.1 — Картинка в приложении. Страшная и ужасная.

Приложение Б Еще картинки

Рисунок Б.1 — Еще одна картинка, ничем не лучше предыдущей. Но надо же как-то заполнить место.