

Отчет по лабораторной работе L^AT_EX, Git, GPG

Абрамов Антон

2 июня 2015 г.

Содержание

1	Система верстки T_EX и расширение L^AT_EX	2
1.1	Цель работы	2
1.2	Ход работы	2
1.3	Выводы	5
2	Система контроля версий Git	5
2.1	Цель работы	5
2.2	Ход работы	5
2.3	Выводы	7
3	Программа для шифрования и подписи GPG, пакет Gpg4win	7
3.1	Цель работы	7
3.2	Ход работы	7
3.3	Выводы	10

1 Система верстки \TeX и расширение \LaTeX

1.1 Цель работы

Изучение принципов верстки \TeX , создание первого отчета.

1.2 Ход работы

Изучение

1. Создание минимального файла `.tex` в простом текстовом редакторе - преамбула, тело документа.

Любой файл должен начинаться с преамбулы. При создании преамбулы были задействованы следующие декларации:

- `documentclass` - декларация, которая задает класс документа, содержащий определения команд, специфических для выбранного типа документа. В нашем случае используется класс `article` (статья). Также в декларации можно изменить значение ряда параметров и некоторые правила формирования, принятые по умолчанию для этого класса. Мы задаем формат листа (`a4paper`) и размер шрифта (`10pt`).
- `usepackage` - декларация используется для расширения базовой версии \LaTeX . В фигурных скобках задается имя файла с расширением `.sty`. Этот файл содержит переопределение уже имеющихся команд и определение новых команд. Количество деклараций `usepackage` не ограничено. Одной декларацией можно загрузить сразу несколько пакетов. Также можно указать опции пакетов, они задаются, как и в декларации `documentclass`.

Текст документа размещается за преамбулой в командных скобках `\begin{document} . . . \end{document}`. Всё, что идет после `\end{document}` \LaTeX игнорирует.

2. Компиляция в командной строке - `latex`, `xdvi`, `pdflatex`

Что бы скомпилировать `.tex` файл из командной строки необходимо воспользоваться командой `latex имя_файла.tex`. В случае успеха будет создан файл с расширением `.dvi`, пригодный к распечатке и используемый для предпросмотра документа. Для просмотра `dvi`-файла можно воспользоваться утилитой `Yap`, распространяемой вместе с дистрибутивом `MikTeX`. Для создания `pdf`-документа из `dvi`-файла можно воспользоваться командой `dvipdfm имя_файла.dvi`.

Чтобы напрямую создать pdf-документ из tex-файла воспользуемся командой `pdflatex имя_файла.tex`.

3. Оболочка TexMaker, Быстрый старт, Быстрая сборка

Для удобства создания документов воспользуемся оболочкой TexMaker. TexMaker является свободным, современным и кросс-платформенным редактором L^AT_EX для Linux, MacOSX и Windows, который объединяет множество инструментов, необходимых для разработки документов с L^AT_EX, в одном приложении.

Для задания преамбулы документа можно воспользоваться помощником "Быстрый старт" (Меню "Помощник"). С помощью данного диалога можно задать особенности документа (класс, размер бумаги, кодировки и т.д.). Также можно самому добавить другие возможности кликнув на кнопку "+".

Самый простой способ скомпилировать документ — это использовать команду "Быстрая сборка". Задать последовательность команд используемых командой "Быстрый старт" можно в диалоге "Настроить Texmaker". Для запуска команды из панели инструментов сначала выберем ее, а затем нажмем кнопку "Run".

4. Создание титульного листа, нескольких разделов, списка, несложной формулы

Для создания титульного листа воспользуемся командой `maketitle`, который выводит заголовок, автора и дату. Ей должны предшествовать две команды `title{...}` и `author{...}`, содержащие название документа и автора соответственно. Аргументы обеих команд могут быть пустыми.

Перед `maketitle` можно с помощью команды `date{...}` указать дату создания документа. Если команда `date` отсутствует, то печатается текущая дата.

Чтобы создать несколько разделов мы воспользовались командой `section`, для создания подраздела `subsection`, а под-подраздела `subsubsection`.

Для создания списка используется следующая конструкция: `\beginenumerate ... \endenumerate`. Каждый пункт списка нумеруется с помощью команды `item`.

Чтобы текст воспринимался как формула, необходимо обособить его знаком `$` в начале и в конце или `\[` в начале и `\]` в конце. Например: $x + y = 5$ или

$$x + y = 5$$

Сразу заметно отличие первого способа от второго. В первом случае формула является включенной в текст и занимает меньше места, так как должна поместиться в строку. Во втором случае формула, выключенная и располагается по центру страницы, при этом не ограничивая себя по высоте.

5. Понятие классов документов, подключение пактов

Как уже было сказано выше, в преамбуле используется декларация `documentclass`, которая задает класс документа. Класс определяет тип создаваемого документа. Основные классы документов:

- `article` - для статей в научных журналах, презентаций, коротких отчетов, программной документации, приглашений и т.д.;
- `report` - для более длинных отчетов, содержащих несколько глав, небольших книжек, диссертаций и т.д.;
- `book` - для настоящих книг;
- `slides` - для слайдов. Использует большие буквы без засечек.

6. Верстка более сложных формул

Мы уже попробовали создать простую включенную и выключенную формулы. Теперь создадим что-то более сложное. Для ясного задания стиля оформления формул используются четыре команды:

- Выключенная формула - команда `\displaystyle: \frac{1}{2} \int dF`
- Текстовая формула - команда `\textstyle: \frac{1}{2} \int dF`
- Индексная формула - команда `\scriptstyle: \frac{1}{2} \int dF`
- Подиндексная формула - команда `\scriptscriptstyle: \frac{1}{2} \int dF`

Для написания тех или иных формул существуют соответствующие команды. Их достаточно много, поэтому не имеет смысла все перечислять. Достаточно воспользоваться справочным пособием при написании формул и найти там необходимые команды. Пример нескольких сложных формул ниже:

- Использование радикалов

$$\sqrt{x} + \sqrt[3]{x+y}$$

- Команда `\frac` создаёт дробь

$$\frac{x}{2} + \frac{1}{1+x} + \frac{1+x}{2} + \frac{1+x}{1-x}$$

- Команда `\substack` создаёт многоэтажные индексы

$$\sum_{\substack{n_1, n_2, \dots, n_r \\ n_1 + n_2 + \dots + n_r = n \\ n_1, n_2, \dots, n_r > 0}} \frac{n!}{n_1! n_2! \dots n_r!}$$

Выполнение практического задания Создание отчетов по лабораторным работам

1.3 Выводы

На первый взгляд может показаться, что верстать документы в \TeX сложно и занимает много времени. Однако после создания первого документа, и ознакомления с основными командами, работа с \TeX становится куда быстрее и приятней.

2 Система контроля версий Git

2.1 Цель работы

Изучить систему контроля версий Git, освоить основные примеры работы с ней.

2.2 Ход работы

1. Изучить справку для основных команд

По адресу <http://git-scm.com/book/ru/v1> расположен хороший учебник, описывающий работу с git и пример использования команд.

2. Получить содержимое репозитория

Для клонирования существующего репозитория воспользуемся командой **git clone [url]**. После чего содержимое центрального репозитория окажется в локальном репозитории.

3. Добавить новую папку и первого файла под контроль версий

Добавим новую папку в рабочем каталоге, а в папку положим файл. Для того, чтобы добавления их под контроль версий (проиндексировать) воспользуемся командой **git add ***.

4. Зафиксируем изменение в локальном репозитории

Фиксируем изменения в локальном репозитории с помощью команды **git commit -m 'имя_коммита'**.

5. Внести изменений в файл и посмотреть различия

Сделав коммит внесем изменения в файл. Напишем команду **git status** и увидим, что изменения произведены, но не зафиксированы.

6. Отменить локальные изменения

На любой стадии может возникнуть необходимость что-либо отменить. Если необходимо отменить индексацию файла, то используем **git reset HEAD <файл>**.

Если необходимо отменить изменения файла, тогда используем **git checkout -- <файл>**.

7. Внести изменения в файл и посмотреть различия

После того как мы отменили все старые изменения и ввели новые, в файле будут только последние изменения. А изменения, которые были сделаны после коммита и до отмены изменений учтены не будут.

8. Зафиксировать изменения в локальном репозитории, зафиксировать изменения в центральном репозитории

Сначала проиндексируем все новые файлы **git add ***. Затем зафиксируем изменения в локальном репозитории **git commit -m 'home reading'**. После этого зафиксируем изменения в центральном репозитории **git push origin master**. Тут origin - сервер, а master - ветка. После этого изменения появятся в центральном репозитории на сервере.

9. Получить изменения из центрального репозитория.

Получить изменения из центрального репозитория можно так: **git pull origin master**. Тут надо учитывать, если изменения на центральном и локальном репозиториях затрагивают одни и те же файлы, то необходимо производить слияние.

10. Поэкспериментировать с ветками

Для создания ветки с именем new: **git branch new**. Для переход к ветке new: **git checkout new**. Чтобы создать ветку и сразу же

перейти на неё, можно выполнить команду `git checkout` с ключом `-b`: **`git checkout -b new`**. После того, как все изменения в новой ветке произведены и протестированы, мы можем слить эту ветку с веткой `master`. Для этого переходим в ветку `master`: **`git checkout master`** и производим слияние с помощью команды `git merge`: **`git merge new`**.

2.3 Выводы

Распределенные системы контроля версий значительно облегчают разработку как индивидуальных, так и командных проектов. Позволяют фиксировать изменения в проекте, производить разработку проекта независимо (в разных ветках), откатываться на предыдущие версии проекта и многое другое.

3 Программа для шифрования и подписи GPG, пакет Gpg4win

3.1 Цель работы

Научиться создавать сертификаты, шифровать файлы и ставить ЭЦП.

3.2 Ход работы

1. Изучить документацию, запустить графическую оболочку Kleopatra

В ходе выполнения лабораторной работы была установлена и освоена графическая оболочка Kleopatra.

2. Создать ключевую пару OpenPGP (File->New Certificate)

Была создана ключевая пара OpenPGP (PGP/MIME). Для этого в главном меню выбираем File->New Certificate. Далее из предложенных вариантов указываем, что необходимо создать ключевую пару OpenPGP. После этого вводим имя, электронную почту и passphrase. Далее ждем пока будет создана пара ключей Open PGP и нажимаем Finish.

3. Экспортировать сертификат (File->Export Certificate)

Экспортируем сертификат выбрав в главном меню File->Export Certificate и указав папку, куда необходимо произвести экспорт. В этой папке

будет создан файл с расширением .asc, который содержит открытый ключ. Теперь этот файл можно передать партнеру, для возможности передавать сообщения и файлы в зашифрованном виде.

4. Поставить ЭЦП на файл (File->Sing/Encrypt Files)

Что бы подписать файл выберем в главном меню пункт File->Sing/Encrypt Files. Теперь укажем файл, который необходимо подписать и выберем пункт Sing. Указываем сертификат, которым производится подпись и файла. Теперь подтверждаем выбор и нажимаем Sing. Вводим passphrase в диалоговом окне PIN-кода. После того, как подпись будет создана, появится файл подписи с расширением .sig.

5. Получить чужой сертификат из репозитория, файл с данными и файл с сигнатурой(подписью)

Из репозитория возьмем файл с данными - myfirst.pdf, файл подписи - myfirst.pdf.sig и файл сертификата - karina.asc.

6. Импортировать сертификат, подписать его

Для импортирования сертификата в главном меню выбираем File->Import Certificates и выберем в качестве сертификата файл karina.asc. После чего сертификат будет добавлен в список импортированных сертификатов.

7. Проверить подпись

Для проверки подлинности и целостности файла myfirst.pdf нажмем на нем правой кнопкой, выберем пункт контекстного меню: Другие параметры GpgEX->Проверить. Для проверки целостности и подлинности необходимо, чтобы файл подписи лежал в одной папке с исходным файлом. Это позволит автоматически найти соответствие между ними. Подтвердим операцию нажатием Decrypt/Check. В случае успеха, будет выдано соответствующее сообщение. Если же в файле было произведено малейшее изменение, после совершение подписи, то подтверждение целостности пройдет неудачно.

8. Взять сертификат кого-либо из коллег, зашифровать и подписать для него какой-либо текст, предоставить свой сертификат, убедиться, что ему удалось получить открытый текст, проверить подпись.

Для пары ключей OpenPGP возможно одновременно подписать и зашифровать файл. Причем подпись происходит до шифрования, что позволяет проверить подлинность и целостность файла только

после расшифровки. Для этого в главном меню выбираем пункт File->Sing/Encrypt Files, указываем путь до файла, а далее указываем пункт Sing end Encrypt. После этого указываем все сертификаты, которыми будет шифровать файл и производим подпись и шифрование. Будет создан файл с расширением .gpg.

Эксперимент был проведен в паре с моей коллегой, Агафоновой Оксаной. Мною был зашифрован и подписан документ Privet.docx, приложенный к отчету. В итоге получился файл Privet.docx.gpg, он и был отправлен Оксане. Как сообщила моя коллега, эксперимент прошел успешно и файл был ею расшифрован и прочитан.

9. Предыдущий пункт наоборот

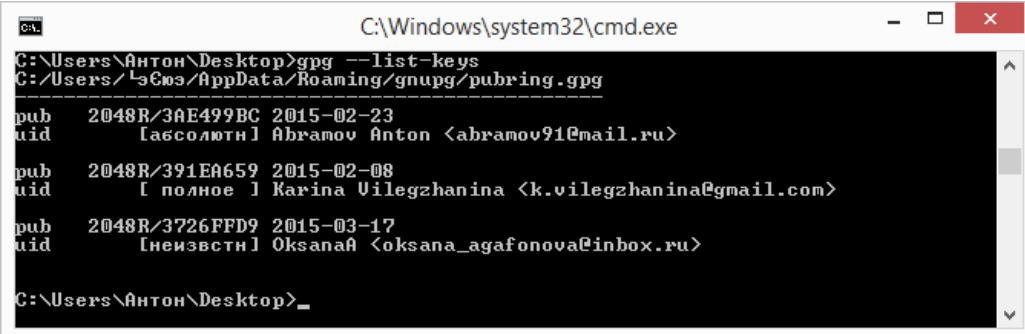
Мной был взят файл с открытым ключом и импортирован в систему Kleopatra. Далее получив файл 111.txt.gpg от Оксаны, расшифровал и проверил подпись. Эксперимент прошел успешно. Содержимое файла: **Привет**). Файл приложен к отчету.

10. Изучить GNU Privacy handbook, протестировать в использовании gpg через интерфейс командной строки, без использования графических оболочек.

Опция **-gen-key** используется для создания новой пары ключей.

Для создания сертификата используется опция **-gen-revoke**.

Получить набор ключей можно с помощью опции **-list-keys**.



```
C:\Windows\system32\cmd.exe
C:\Users\Антон\Desktop>gpg --list-keys
C:\Users\Антон\AppData\Roaming\gnupg\pubring.gpg
-----
pub 2048R/3AE499BC 2015-02-23
uid [абсолютн] Abramov Anton <abramov91@mail.ru>
pub 2048R/391EA659 2015-02-08
uid [ полное ] Karina Vilegzhanina <k.vilegzhanina@gmail.com>
pub 2048R/3726FFD9 2015-03-17
uid [неизвестн] OksanaA <oksana_agafonova@inbox.ru>
C:\Users\Антон\Desktop>_
```

Рис. 1: Использование опции -list-keys

Для отправки открытого ключа необходимо сначала его экспортировать, а для этого у нас есть опция **-export**. Опция принимает дополнительный аргумент, идентифицирующий открытый ключ на

экспорт. Ключи можно не только импортировать, но и экспортировать. Для этого существует опция **-import**, а в качестве аргумента указывается имя импортируемого ключа.

Вот мы и подошли к основным опциям gpg, первой из них рассмотрим шифрование - это опция **-encrypt**. На рисунке 2 приведен пример ее использования. Результирующий файл указан через **-output**. **-recipient** опция используется один раз для каждого получателя и принимает дополнительный аргумент, указывающий ключ, которым будет зашифрован документ.

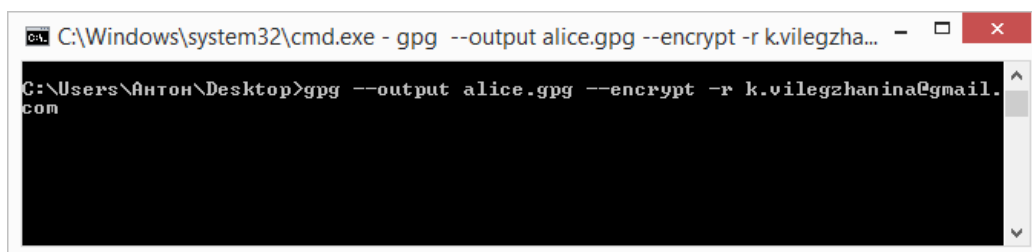


Рис. 2: Шифрование документа

При получении зашифрованного документа необходимо расшифровать его, в этом нам поможет опция **-decrypt**. После ввода команды будет предложено ввести ключевую фразу.

Подписать файл можно указав опцию **clearsign**.

Есть также множество других опций, для разных целей. Но самой главной является **-help** - ключ к остальным опциям (Рис. 3).

3.3 Выводы

В ходе выполнения третьего пункта лабораторной работы была освоена программа Kleopatra, входящая в пакет Gpg4win и используемая для шифрования и подписи GPG. Была создана пара ключей OpenPGP, в которой закрытый ключ защищен при помощи passphrase, а открытый ключ распространяется как сертификат (файл с расширением .asc). Произведен импорт нескольких сертификатов, что позволило зашифровать файлы, которые могут быть расшифрованы только владельцами этих сертификатов. Также была произведена подпись документов и соответственно проверка це-

```

C:\Windows\system32\cmd.exe

Команды:
-s, --sign                создать подпись
--clearsign              создать прозрачную подпись
-b, --detach-sign        создать отделенную подпись
-e, --encrypt            зашифровать данные
-c, --symmetric           зашифровать только симметричным шифром
-d, --decrypt            расшифровать данные (по умолчанию)
--verify                проверить подпись
-k, --list-keys           вывести список ключей
--list-sigs              вывести список ключей и подписи
--check-sigs             вывести и проверить подписи
--fingerprint            вывести список ключей с отпечатками
-K, --list-secret-keys   вывести список секретных ключей
--gen-key                создать новую пару ключей
--gen-revoke             создать сертификат отзыва
--delete-keys            удалить ключи из таблицы открытых ключей
--delete-secret-keys     удалить ключи из таблицы закрытых ключей
--sign-key               подписать ключ
--lsign-key              подписать ключ локально
--edit-key               подписать или редактировать ключ
--passwd                сменить фразу-пароль
--export                 экспортировать ключи
--send-keys              экспортировать ключи на сервер ключей
--recv-keys              импортировать ключи с сервера ключей
--search-keys            искать ключи на сервере ключей
--refresh-keys           обновить все ключи с сервера ключей
--import                импортировать/объединить ключи
--card-status            показать состояние карты
--card-edit              изменить данные на карте
--change-pin             сменить PIN карты
--update-trustdb          обновить таблицу доверий
--print-md               вывести хэши файлов
--server                 запуск в режиме сервера

Параметры:
-a, --armor              вывод в ASCII формате
-r, --recipient USER-ID зашифровать для USER-ID
-u, --local-user USER-ID использовать USER-ID для подписывания и расшифровки
-z N                     установить уровень сжатия N (0 без сжатия)
--textmode               использовать канонический текстовый режим
-o, --output FILE        взять параметры из FILE
-v, --verbose            подробно
-n, --dry-run            не делать никаких изменений
-i, --interactive        спросить перед перезаписью
--openpgp                строго следовать стандарту OpenPGP

(См. документацию для более полного ознакомления с командами и параметрами)

Примеры:
-se -r Bob [файл]        подписать и зашифровать для получателя Bob
--clearsign [файл]       создать прозрачную подпись
--detach-sign [файл]     создать отделенную подпись
--list-keys [имена]      показать ключи
--fingerprint [имена]    показать отпечатки

0 найденных ошибок сообщайте <http://bugs.gnupg.org>.

```

Рис. 3: А как насчет справки?

лостности и подлинности этих и других документов. Для проверки целостности и подлинности файла необходим сам файл, а также файл подписи.