

Лабораторная работа н.3

Markdown

ЛИ РОМАН ПАВЛОВИЧ

Содержание

Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий по предмету.

Теоретическое введение

Чтобы создать заголовок, используйте знак (#), например:

```
1 # This is heading 1
2 ## This is heading 2
3 ### This is heading 3
4 #### This is heading 4
```

Чтобы задать для текста полужирное начертание, заключите его в двойные звездочки:

```
1 This text is bold.
```

Чтобы задать для текста курсивное начертание, заключите его в одинарные звездочки:

```
This text is italic.
```

Чтобы задать для текста полужирное и курсивное начертание, заключите его в тройные звездочки:

This is text is both *****bold and italic*****

Блоки цитирования создаются с помощью символа >: > The drought had lasted now for ten million years, and the reign of the terrible lizards had long since ended. Here on the Equator, in the continent which would one day be known as Africa, the battle for existence had reached a new climax of ferocity, and the victor was not yet in sight. In this barren and desiccated land, only the small or the swift or the fierce could flourish, or even hope to survive.

Неупорядоченный (маркированный) список можно отформатировать с помощью звезд- дочек или тире:

- List item 1
- List item 2
- List item 3

Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка:

- List item 1
 - List item A
 - List item B
 - List item 2

Упорядоченный список можно отформатировать с помощью соответствующих цифр:

1. First instruction
1. Second instruction
1. Third instruction

Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка:

1. First instruction
 - 1. Sub-instruction
 - 1. Sub-instruction
1. Second instruction

Синтаксис Markdown для встроенной ссылки состоит из части [link text] , представляющей текст гиперссылки, и части (file-name.md) – URL-адреса или имени файла, на который дается ссылка:

[link text](file-name.md)

Markdown поддерживает как встраивание фрагментов кода в предложение, так и их размещение между предложениями в виде отдельных огражденных блоков. Огражденные блоки кода — это простой способ выделить синтаксис для фрагментов кода. Общий формат огражденных блоков кода:

your code goes in here

Верхние и нижние индексы: записывается как

$H_{20} \sim 0$

записывается как

2^{10^4}

Внутритекстовые формулы делаются аналогично формулам LaTeX. Например, формула $\sin^2(x) + \cos^2(x) = 1$ запишется как

$$\sin^2(x) + \cos^2(x) = 1$$

Выключные формулы: $\sin^2(x) + \cos^2(x) = 1$ {#eq:eq:sin2+cos2} со ссылкой в тексте «Смотри формулу ([@eq:eq:sin2+cos2]).» записывается как

```
1 $$
2 \sin^2 (x) + \cos^2 (x) = 1
3 $$ {#eq:eq:sin2+cos2}
4
5 Смотри формулу ([@eq:eq:sin2+cos2])
```

Для обработки файлов в формате Markdown будем использовать Pandoc

<https://pandoc.org/>. Конкретно, нам понадобится программа pandoc , pandoc-citeproc

<https://github.com/jgm/pandoc/releases>, pandoc-crossref

<https://github.com/lierdakil/pandoc-crossref/releases>.

Преобразовать файл README.md можно следующим образом:

```
1 pandoc README.md -o README.pdf
или так
1 pandoc README.md -o README.docx
Можно использовать следующий Makefile
1 FILES = $(patsubst %.md, %.docx, $(wildcard *.md))
2 FILES += $(patsubst %.md, %.pdf, $(wildcard *.md))
3
4 LATEX_FORMAT =
5
6 FILTER = --filter pandoc-crossref
7
8 %.docx: %.md
9 -pandoc "$<" $(FILTER) -o "$@"
10
11 %.pdf: %.md
12 -pandoc "$<" $(LATEX_FORMAT) $(FILTER) -o "$@"
13
14 all: $(FILES)
15 @echo $(FILES)
16
```

```
17 clean:
18 -rm $(FILES) *~
```

Выполнение лабораторной работы

Создание учетной записи на github.com

- Создание учетной записи в моем случае не требуется, поэтому перейдем к следующему заданию.

Установка Программного обеспечения

Установка git-flow

- Установим git-flow согласно указаниям (рис. [-@fig:001])

```
[aepetrov@fedora ~]$ cd /tmp
[aepetrov@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[aepetrov@fedora tmp]$ chmod +x gitflow-installer.sh
[aepetrov@fedora tmp]$ sudo ./gitflow-installer.sh install stable
```

1. Установка git-flow в терминале

Установка gh

- Установим gh (рис. [-@fig:002])

```
[aepetrov@fedora tutorial]$ sudo dnf install gh
[sudo] password for aepetrov:
```

2. Установка gh

Базовая настройка git

Зададим имя и email (рис. [-@fig:003]):

1. git config --global user.name "Artyem Petrov"
2. git config --global user.email "fittedorangeofficial@mail.ru"

```
[aepetrov@fedora tutorial]$ git config --global user.email "fittedorangeofficial@mail.ru"
>
> "
[aepetrov@fedora tutorial]$ git config --global user.name "Artyem Petrov"
```

3. Установление почты и имени для git

Настроим utf-8 в выводе сообщений git (рис. [-@fig:004]):

1. git config --global core.quotePath false

Настроим верификацию и подписание коммитов(1) и зададим имя начальной ветки(2) (рис. [-@fig:004]):

- Настроим верификацию и подписание коммитов:
1. git config --global init.defaultBranch master
- Параметры autocrlf и safecrlf:

1. `git config --global core.autocrlf input`
2. `git config --global core.safecrlf warn`

```
[aepetrov@fedora tutorial]$ git config --global core.quotepath false
[aepetrov@fedora tutorial]$ git config --global init.defaultBranch master
[aepetrov@fedora tutorial]$ git config --global core.autocrlf input
[aepetrov@fedora tutorial]$ git config --global core.safecrlf warn
```

4. Установление почты и имени для git

Создание ключей ssh по алгоритмам rsa & ed25519(рис. [-@fig:005])

1. `ssh-keygen -t rsa -b 4096`
2. `ssh-keygen -t ed25519`

```

[aepetrov@fedora tutorial]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aepetrov/.ssh/id_rsa):
/home/aepetrov/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aepetrov/.ssh/id_rsa
Your public key has been saved in /home/aepetrov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:cjvRfmenUT+oy6wp2lZGNVhH0Eki4TBclDzq00IQwBk aepetrov@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|  .E+  .o+==+=+. |
|  o .  .+=.000 |
|  .  ..o .  |
|  .  o .  |
|  o S o  . |
|  + = o  ... |
|  . o = ..+.o |
|  o.=+..o +. |
|  .+=o=. .  |
+-----[SHA256]-----+
[aepetrov@fedora tutorial]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/aepetrov/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aepetrov/.ssh/id_ed25519
Your public key has been saved in /home/aepetrov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:UGcR7TjuJ7ZRl8kFyn3PPyQkgVS2dfLbKBK3GcFRdwk aepetrov@fedora
The key's randomart image is:
+--[ED25519 256]--+
|  o.B0oE+.+ |
|  . +..*+++. |
|  .  .==...o |
|  . oo+= B+ |
|  S...=. *o+ |
|  .o oo . |
|  ..  .. |
|  +..  . |
|  ..+ |
+-----[SHA256]-----+

```

5. Генерация ключей по вышеописанным алгоритмам

Создание ключа pgp(рис. [-@fig:006])

- Генерируем ключ:
1. `gpg --full-generate-key`
- Выбираем следующим образом: RSA & RSA, 4096, 0, Artyem, fittedorangeofficial@mail.ru.

```
[aepetrov@fedora tutorial]$ gpg --full-generate-key
gpg (GnuPG) 2.3.4; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/aepetrov/.gnupg' created
gpg: keybox '/home/aepetrov/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Artyem
Email address: fittedorangeofficial@mail.ru
Comment:
You selected this USER-ID:
  "Artyem <fittedorangeofficial@mail.ru>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? 0
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/aepetrov/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/aepetrov/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/aepetrov/.gnupg/openpgp-revocs.d/D6470ED047E1D5851300B89BFCB2392B47851AE7.rev'
public and secret key created and signed.

pub   rsa4096 2022-04-29 [SC]
      D6470ED047E1D5851300B89BFCB2392B47851AE7
uid
      Artyem <fittedorangeofficial@mail.ru>
sub   rsa4096 2022-04-29 [E]
```

6. Создание ключа pgr

Добавление PGP ключа в GitHub

- Выводи список ключей и копируем отпечаток приватного ключа(рис. [-@fig:007]):
- ```
gpg --list-secret-keys --keyid-format LONG
```
- Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

- Формат строки: sec Алгоритм/Отпечаток\_ключа Дата\_создания [Флаги]  
[Годен\_до] ID\_ключа Ключен выделен на рис. [-@fig:007]:

```
[aepetrov@fedora tutorial]$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/home/aepetrov/.gnupg/pubring.kbx

sec rsa4096/FCB2392B47851AE7 2022-04-29 [SC]
 D6470ED047E1D5851300B89BFCB2392B47851AE7
uid [ultimate] Artyem <fittedorangeofficial@mail.ru>
ssb rsa4096/1A0071C2C730C4D9 2022-04-29 [E]
```

### 7.Нахождение ключа рдр

- Скопируйте ваш сгенерированный PGP ключ в буфер обмена(рис. [-@fig:008]):
1. `gpg --armor --export <PGP Fingerprint> | xclip -sel clip`

```
[aepetrov@fedora tutorial]$ gpg --armor --export FCB2392B47851AE7 | xclip -sel clip
```

### 8.Копирование ключа рдр

– Перейдем в настройки GitHub (<https://github.com/settings/keys>), нажмем на кнопку New GPG key и вставьте полученный ключ в поле ввода(рис. [-@fig:009])



The screenshot shows the GitHub profile page for a user named 'Artyem'. The left sidebar contains navigation links for Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, SSH and GPG keys (highlighted), Organizations, Moderation, Code, planning, and automation, Security, and Integrations. The main content area is titled 'GPG keys / Add new' and shows a GPG key in a text box. The key is a long string of characters, including 'r0316GGlv2W7Q/BFKfrZhlawfzCqUoh3bXUqFbrEJzldCjjTPGwlJtQtBkTkleIH'. Below the text box is a green 'Add GPG key' button.

## 9. Копирование ключа pgr

### Настройка автоматических подписей коммитов git

- Используя введённый email, укажем Git применять его при подписи коммитов (рис. [-@fig:010]):

- `git config --global user.signingkey <PGP Fingerprint>`
- `git config --global commit.gpgsign true`
- `git config --global gpg.program $(which gpg2)`

```
[aepetrov@fedora tutorial]$ git config --global user.signingkey FCB2392B47851AE7
[aepetrov@fedora tutorial]$ git config --global commit.gpgsign true
[aepetrov@fedora tutorial]$ git config --global gpg.program $aepetrov
```

## 10. Настройка автоматических подписей коммитов git

## Настройка gh.

- Для начала авторизуемся(рис. [-@fig:011]):  
gh auth login
- Утилита задаст несколько вопросов и попросит токен в конце аутентификации, который можно создать в настройках разработчика(<https://github.com/settings/tokens>)[link]. Если токен введен правильно, то вы авторизируетесь.

```
[aepetrov@fedora tutorial]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/aepetrov/.ssh/id_ed25519.pub
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
error validating token: HTTP 401: Bad credentials (https://api.github.com/)
Try authenticating with: gh auth login
[aepetrov@fedora tutorial]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/aepetrov/.ssh/id_rsa.pub
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/aepetrov/.ssh/id_rsa.pub
✓ Logged in as wlcmtunknwnth
```

## 11. Настройка автоматических подписей коммитов git

### Шаблон для рабочего пространства

- Шаблон находится по ссылке: (<https://github.com/yamadharma/course-directory-student-template>)

### Создание репозитория на основе шаблона

- Создадим необходимую директорию и перейдем в нее(рис. [-@fig:012]):
  1. `mkdir -p ~/work/study/2021-2022/"Операционные системы"`
  2. `cd ~/work/study/2021-2022/"Операционные системы"`
- Создадим репозиторий в нашей директории(рис. [-@fig:012]):
  1. `gh repo create study_2021-2022_os-intro`  
`--template=yamadharma/course-directory-student-template --public`
- Скопируем содержимое репозитория преподавателя к нам в репозиторий(рис. [-@fig:012]):
  1. `git clone --recursive`
  2. `git@github.com:<owner>/study_2021-2022_os-intro.git os-intro`

```

[aepetrov@fedora tutorial]$ mkdir -p ~/work/study/2021-2022/"Операционные системы"
[aepetrov@fedora tutorial]$ cd ~/work/study/2021-2022/"Операционные системы"
[aepetrov@fedora Операционные системы]$ gh repo create study_2021-2022_os-intro --template=yamadharm/course-directory
e/re-student-template --public
✓ Created repository wlcmtunknwdth/study_2021-2022_os-intro on GitHub
[aepetrov@fedora Операционные системы]$ git clone --recursive git@github.com:wlcmtunknwdth/study_2021-2022_os-intro.git
Cloning into 'os-intro'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Receiving objects: 100% (20/20), 12.49 KiB | 6.25 MiB/s, done.
Resolving deltas: 100% (2/2), done.
Submodule 'template/presentation' (https://github.com/yamadharm/academic-presentation-markdown-template.git) register
ed for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) registered for pat
h 'template/report'
Cloning into '/home/aepetrov/work/study/2021-2022/Операционные системы/os-intro/template/presentation'...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 42 (delta 9), reused 40 (delta 7), pack-reused 0
Receiving objects: 100% (42/42), 31.19 KiB | 725.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
Cloning into '/home/aepetrov/work/study/2021-2022/Операционные системы/os-intro/template/report'...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.

```

### Настройка каталога курса

- Перейдем в каталог курса(рис. [-@fig:013])

```
cd ~/work/study/2021-2022/"Операционные системы"/os-intro
```

- Удалим лишние файлы(рис. [-@fig:013])

```
rm package.json
```

- Создадим необходимые каталоги(рис. [-@fig:013]):

```
make COURSE=os-intro
```

- После отправим файлы на сервер(рис. [-@fig:013]):

1. git add .
2. git commit -am 'feat(main): make course structure'

```

[aepetrov@fedora Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[aepetrov@fedora os-intro]$ rm package.json
[aepetrov@fedora os-intro]$ make COURSE=os-intro
[aepetrov@fedora os-intro]$ git add .
[aepetrov@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master 91aca03] feat(main): make course structure
149 files changed, 16590 insertions(+), 14 deletions(-)
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg

```

### 13. Настройка каталога курса

- Окончательно “толкнем” их на сервер(рис. [-@fig:014]):

`git push`

```
[aepetrov@fedora os-intro]$ git push
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 6 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (19/19), 266.53 KiB | 2.34 MiB/s, done.
Total 19 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:wlcmtunknwndth/study_2021-2022_os-intro.git
 6eca554..91aca03 master -> master
```

*14.Отправка на сервер*

## Выводы

Благодаря данной лабораторной работе я научился пользоваться языком разметки markdown, что, как я считаю, невероятно важно. Ведь важно не просто пользоваться графическим интерфейсом, но и уметь пользоваться базисными методами оформления текста. Спасибо за внимание!