

Отчет к лабораторной работе №3

Дисциплина: операционные системы

Крупенникова Виктория Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	10

List of Tables

List of Figures

1 Цель работы

Изучить идеологию и применение средств контроля версий

2 Задание

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. – В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (архив), поскольку он должен содержать скриншоты, Makefile ит.д.

3 Выполнение лабораторной работы

Создаю учетную запись на <https://github.com>. (рис. ??) Учетная запись

Настраиваю систему контроля версий git. Синхронизирую учётную запись github с компьютером: `git config --global user.name "Имя Фамилия" git config`

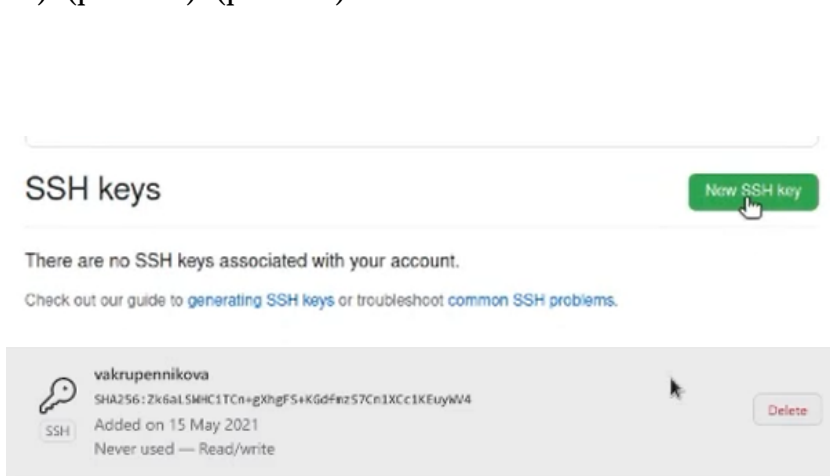
`--global user.email "work@mail"` (рис. ??)

```
vakrupennikova@dk4n58 ~$ git config --global user.name "vakrupennikova"
vakrupennikova@dk4n58 ~$ git config --global user.email "krupka2002@yandex.ru"
```

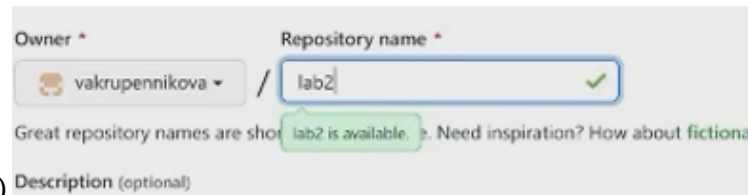
Создаю новый ключ на github (команда `ssh-keygen -C "vakrupennikova krupka2002@yandex.ru"`) и привязываю его к копыютеру через консоль. (рис. ??) (рис.

```
vakrupennikova@dk4n58 ~$ ssh-keygen -C "vakrupennikova <krupka2002@yandex.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Zk6aLSMHC1TCn+gXhgFS+KGdFmzS7Cn1XCc1KEuyWV4 vakrupennikova <krupka2002@yandex.ru>
The key's randomart image is:
+--[RSA 3072]-----+
|oo.
|=
|o*.o
|oo0 o +.E o
|+.0 B+S.
|.+.o%o
|.B.+.oo
| o o.
|
+-----[SHA256]-----+
```

??) (рис. ??) (рис. ??)



В github захожу в «repositories» и создаю новый репозиторий. Копируем в кон-



соль ссылку на репозиторий.(рис. ??)(рис. ??)

```
vakrupennikova@dk4n58 ~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
vakrupennikova@dk4n58 ~$ git clone https://github.com/vakrupennikova/lab2.git
Клонирование в «lab2»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (3/3), готово.
```

Работаю с каталогом и папками через консоль. Перед тем, как создавать файлы, захожу в репозиторий и создаю файлы: (рис. ??)(рис. ??)

```
vakrupennikova@dk4n58 ~$ cd lab2
vakrupennikova@dk4n58 ~/lab2$ mkdir 2020-2021
vakrupennikova@dk4n58 ~/lab2$ cd 2020-2021
vakrupennikova@dk4n58 ~/lab2/2020-2021$ mkdir os
vakrupennikova@dk4n58 ~/lab2/2020-2021$ cd os
vakrupennikova@dk4n58 ~/lab2/2020-2021/os$ mkdir laboratory
vakrupennikova@dk4n58 ~/lab2/2020-2021/os$ cd laboratory
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory$
```

Добавляю первый коммит и выкладываю на github. Для того, чтобы правильно разместить первый коммит, необходимо добавить команду `git add .`, далее с помощью команды `git commit -m "first commit"` выкладываем коммит:(рис. ??)

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory$ git add .
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory$ git commit -am "first commit"
[main 63e839b] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2020-2021/os/laboratory/h.txt
```

```
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory$
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (7/7), 6.59 KiB | 6.59 MiB/s, готово.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vakrupennikova/lab2.git
63e839b..4e66693 main -> main
```

Сохраняю первый коммит (`git push`):(рис. ??)

Добавляю файл лицензии (скринкаст оборвался, скриншоты тоже). Добавляю шаблон игнорируемых файлов. Получаю список имеющихся шаблонов. Скачиваю шаблон, например, для C. Также добавляю новые файлы и выполняю коммит. Отправляю на github (`git push`): (рис. ??)


```

vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (7/7), 6.59 KiB | 6.59 MiB/s, готово.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vakrupennikova/lab2.git
63e839b..4e66693 main -> main

```

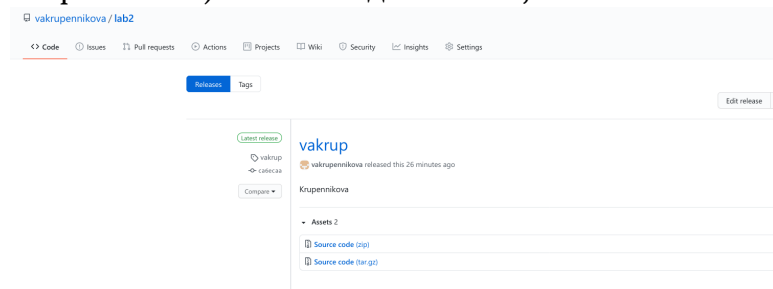
Инициализирую git-flow, используя команду `git flow init -f` (префикс для ярлыков установлен в v): Проверяю, что нахожусь на ветке `develop` (`git branch`): Создаю релиз с версией 1.0.0: Записываю версию и добавляю в индекс: `echo "1.0.0" > VERSION` `git add .` `git commit -am 'chore(main): add version'` Заливаю релизную ветку в основную ветку (команда `git flow release finish 1.0.0`): Отправляю данные на github: `git push - -all` `git push - -tags` (рис. ??)

```

vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push --all
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vakrupennikova/lab2.git
 * [new branch]      develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
vakrupennikova@dk4n58 ~/lab2/2020-2021/os/laboratory $ git push --tags
Username for 'https://github.com': vakrupennikova
Password for 'https://vakrupennikova@github.com':
Everything up-to-date

```

Создаю релиз на github. Заходим в «Releases», нажимаю «Создать новый релиз». Захожу в теги и заполняю все поля (теги для версии 1.0.0). После создания тега, ав-



томатически сформируется релиз. (рис. ??)

4 Выводы

В ходе выполнения лабораторной работы я изучила идеологию и применение средств контроля версий