

Отчет по лабораторной работе №3

Дисциплина: операционные системы

Лобанова Полина Иннокентьевна

Содержание

1	Примечание	5
2	Цель работы	6
3	Задание	7
4	Выполнение лабораторной работы	8
5	Ответы на контрольные вопросы	15
6	Выводы	17

Список иллюстраций

4.1	Установка <i>git</i> .	8
4.2	Установка <i>gh</i> .	8
4.3	Имя пользователя и <i>email</i> .	9
4.4	Настройка <i>utf-8</i> .	9
4.5	Имя начальной ветки.	9
4.6	Настройка параметра <i>autocrlf</i> .	9
4.7	Настройка параметра <i>safecrlf</i> .	9
4.8	Создание <i>ssh</i> ключа.	10
4.9	Генерация <i>pgr</i> ключа.	10
4.10	Учетная запись на <i>github</i> .	11
4.11	Список ключей и отпечаток приватного ключа.	11
4.12	Копирование сгенерированного ключа в буфер обмена.	11
4.13	Настройка автоматических подписей коммитов <i>git</i> .	12
4.14	Авторизация.	12
4.15	Авторизация успешно выполнена.	12
4.16	Создание репозитория курса.	12
4.17	Создание репозитория курса.	13
4.18	Переход в каталог <i>~/work/study/2022-2023/“Операционные системы”/os-intro</i> .	13
4.19	Удаление файла <i>package.json</i> .	13
4.20	Создание каталогов.	13
4.21	Оправление файлов на сервер.	14
4.22	Оправление файлов на сервер.	14

Список таблиц

1 Примечание

В качестве отчета к лабораторной работе №3 нужно предоставить отчет по лабораторной работе №2.

2 Цель работы

Целью работы является изучение идеологии и применения средств контроля версий и освоение умения по работе с git.

3 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.

4 Выполнение лабораторной работы

1. Установим git.

```
[liveuser@localhost-live ~]$ sudo dnf install git
Последняя проверка окончания срока действия метаданных: 0:02:16 назад, Сб 17 сен 2022 12:19:22.
Пакет git-2.35.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[liveuser@localhost-live ~]$
```

Рис. 4.1: Установка git.

2. Установим gh.

```
[pilobanova@10 ~]$ sudo -i
[sudo] пароль для pilobanova:
Попробуйте ещё раз.
[sudo] пароль для pilobanova:
[root@10 ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 1:24:19 назад, Сб 18 фев 2023 00:23:24.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh         x86_64      2.22.1-1.fc36  updates      8.2 М
=====
Результат транзакции
=====
Установка  1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]:
```

Рис. 4.2: Установка gh.

3. Зададим имя и email владельца репозитория.


```
[pilobanova@i0 ~]$ git config --global user.name "pilobanova"  
[pilobanova@i0 ~]$ git config --global user.email "polla-2004@mail.ru"
```

Рис. 4.3: *Имя пользователя и email.*

4. Настроим utf-8 в выводе сообщений git.

```
[pilobanova@i0 ~]$ git config --global core.quotepath false
```

Рис. 4.4: *Настройка utf-8.*

5. Зададим имя начальной ветки.

```
[pilobanova@i0 ~]$ git config --global init.defaultBranch master
```

Рис. 4.5: *Имя начальной ветки.*

6. Зададим параметр autocrlf.

```
[pilobanova@i0 ~]$ git config --global core.autocrlf input
```

Рис. 4.6: *Настройка параметра autocrlf.*

7. Зададим параметр safecrlf.

```
[pilobanova@i0 ~]$ git config --global core.safecrlf warn
```

Рис. 4.7: *Настройка параметра safecrlf.*

8. Создадим ssh ключ.

```
[pilobanova@i0 ~]$ ssh-keygen -C "Polina Lobanova polla-2004@mail.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pilobanova/.ssh/id_rsa):
/home/pilobanova/.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/pilobanova/.ssh/id_rsa
Your public key has been saved in /home/pilobanova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:KkGlbzsrVuxAEgajFUDg4anfSaQW31KLLtRz++hiWs Polina Lobanova polla-2004@mail.ru
The key's randomart image is:
+---[RSA 3072]-----+
|B=0. .|
|+o= o .|
|.= .o . o|
|. ...o + o|
|. .o=.* S o|
|. B.Ooo .|
| + B+* o|
|. o+.oE. + .|
|. ...o.o .|
|+---[SHA256]-----+
```

Рис. 4.8: Создание ssh ключа.

9. Сгенерируем pgp ключ.

```
[root@i0 ~]# gpg --full-generate-key
gpg (GnuPG) 2.3.7; 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.

Выберите тип ключа:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(5) ECC (sign and encrypt) *default*
(10) ECC (только для подписи)
(14) Existing key from card
Ваш выбор? 1
Длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Ваш предложенный размер ключа - 4096 бит
Выберите срок действия ключа.
0 = не ограничен
<n> = срок действия ключа - n дней
<nw> = срок действия ключа - n недель
<nm> = срок действия ключа - n месяцев
<ny> = срок действия ключа - n лет
Срок действия ключа? (0)
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
Ваше полное имя: Lobanova Polina
Адрес электронной почты: polla-2004@mail.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"Polina Lobanova <polla-2004@mail.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
fjeffjwejeffjiefjiefjefogg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/78D40F1D3AE9162495A08960D4605BB6957765C4.rev'.
Открытый и секретный ключи созданы и подписаны.
```

Рис. 4.9: Генерация pgp ключа.

10. Настроим github.

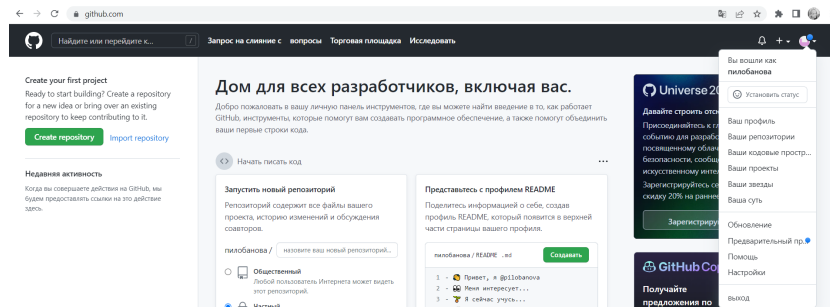


Рис. 4.10: Учетная запись на github.

11. Выведем список ключей и скопируем отпечаток приватного ключа.

```
[root@10 ~]# gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 3 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 3u
/root/.gnupg/pubring.kbx
-----
sec  rsa4096/B16F956ACEF46212 2023-02-17 [SC]
      9B7507B172175EDEA6E49EA9B16F956ACEF46212
uid          [ абсолютно ] Lobanova Polina <polla-2004@mail.ru>
ssb  rsa4096/AAC2D6C849539CBE 2023-02-17 [E]

sec  rsa4096/98B9E9CD6C841FE3 2023-02-17 [SC]
      058A2B4D7D25E8E1D76A02C99BB9E9CD6C841FE3
uid          [ абсолютно ] Lobanova Polina <polla-2004@mail.ru>
ssb  rsa4096/1863EFC40C3598A7 2023-02-17 [E]

sec  rsa4096/D4605BB6957765C4 2023-02-17 [SC]
      78D40F1D3AE9162495A00960D4605BB6957765C4
uid          [ абсолютно ] Lobanova Polina <polla-2004@mail.ru>
ssb  rsa4096/F6CC22C11C97B512 2023-02-17 [E]
```

Рис. 4.11: Список ключей и отпечаток приватного ключа.

12. Скопируем наш сгенерированный rgr ключ в буфер обмена.

```
[root@10 ~]# gpg --armor --export D4605BB6957765C4 | xclip -sel clip
```

Рис. 4.12: Копирование сгенерированного ключа в буфер обмена.

13. Используя введенный email, укажем Git применять его при подписи коммитов.

```
[root@10 ~]# git config --global user.singingkey D4605BB6957765C4
[root@10 ~]# git config --global commit.gpgsing true
[root@10 ~]# git config --global gpg.program $(which gpg2)
```

Рис. 4.13: Настройка автоматических подписей коммитов git.

14. Авторизируемся для настройки gh.

```
[root@10 ~]# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: C717-C198
```

Рис. 4.14: Авторизация.

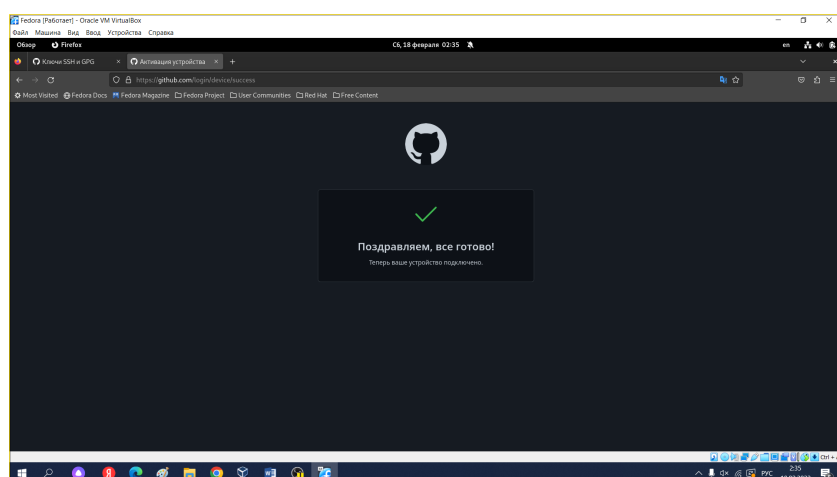


Рис. 4.15: Авторизация успешно выполнена.

15. Создадим репозиторий курса на основе шаблона.

```
[root@10 ~]# mkdir -p ~/work/study/2022-2023/"Операционные системы"
[root@10 ~]# cd ~/work/study/2022-2023/Операционные\ системы/
[root@10 Операционные системы]# gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository pilobanova/study_2022-2023_os-intro on GitHub
```

Рис. 4.16: Создание репозитория курса.

```
[pilobanova@10 ~]$ git clone --recursive git@github.com:pilobanova/study_2022-2023_os-intro.
git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 28 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (28/28), 17.44 КиБ | 279.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markd
own-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-templa
te.git) зарегистрирован по пути «template/report»
Клонирование в «/home/pilobanova/os-intro/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 158.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/pilobanova/os-intro/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
```

Рис. 4.17: Создание репозитория курса.

16. Перейдем в каталог курса.

```
[pilobanova@10 Операционные системы]$ cd os-intro
```

Рис. 4.18: Переход в каталог `~/work/study/2022-2023/“Операционные системы”/os-intro`.

17. Удалим лишний файл.

```
[pilobanova@10 os-intro]$ rm package.json
```

Рис. 4.19: Удаление файла `package.json`.

18. Создадим необходимые каталоги.

```
[pilobanova@10 os-intro]$ echo os-intro > COURSE
[pilobanova@10 os-intro]$ make
```

Рис. 4.20: Создание каталогов.

19. Отправим файлы на сервер.

```
[pilobanova@10 os-intro]$ git add .
[pilobanova@10 os-intro]$ git commit -am 'feat(main): make course structure'
[master 637f428] feat(main): make course structure
361 files changed, 100327 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
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 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
```

Рис. 4.21: Оправление файлов на сервер.

```
[pilobanova@10 os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.39 Киб | 1.53 Миб/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:pilobanova/study_2022-2023_os-intro.git
 180ad1b..637f428 master -> master
[pilobanova@10 os-intro]$
```

Рис. 4.22: Оправление файлов на сервер.

5 Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены? Система, позволяющая работать нескольким людям над одним проектом.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (репозиторий) - директория, хранящая конкретный проект; Коммит - текущее состояние рабочей копии; История - последовательность коммитов в порядке, в котором они добавлялись в репозиторий; Рабочая копия - текущее состояние репозитория, которое находится в состоянии изменения.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. В централизованных VCS (Mercurial) все пользователи подключены к единому серверу; в децентрализованных VCS пользователи подключены к нескольким владельцам.
4. Опишите действия с VCS при единоличной работе с хранилищем. При единоличной работе с хранилищем все изменения, созданные пользователем, не влияют на общий репозиторий.
5. Опишите порядок работы с общим хранилищем VCS. Из общего хранилища можно получать изменения проекта.
6. Каковы основные задачи, решаемые инструментальным средством git? git позволяет нескольким людям работать над одним проектом.

7. Назовите и дайте краткую характеристику командам git. add - добавить файлы в коммит, push - отправить коммит на удалённый репозиторий; pull - импортировать проект с удалённого репозитория.
8. Приведите примеры использования при работе с локальным и удалённым репозиториями.
 - 1) Работа с удаленным репозиторием: git remote – просмотр списка настроенных удаленных репозиториев.
 - 2) Работа с локальным репозиторием: git status - выводит информацию обо всех изменениях, внесенных в дерево директорий проекта по сравнению с последним коммитом рабочей ветки
9. Что такое и зачем могут быть нужны ветви (branches)? Создав новую ветвь, можно, не вредя проекту, работать над конкретной частью проекта.
10. Как и зачем можно игнорировать некоторые файлы при commit? some files may well be user specific

6 Выводы

Я изучила идеологию и применение средств контроля версий и освоила работу с git.