

Отчёт по лабораторной работе №2

Первоначальна настройка git.

Мокочунина Влада Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	15

Список иллюстраций

3.1	Создание учетной записи, заполнение данных.	7
-----	-----------------------------------------------------	---

Список таблиц

1 Цель работы

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

2 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.

3 Выполнение лабораторной работы

1. Я создала учётную запись на сайте <https://github.com/> и заполнила основные данные.(рис. [3.1]).

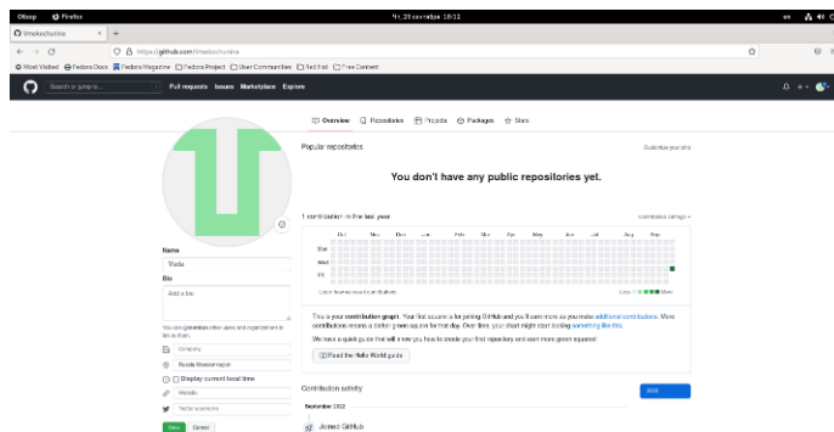
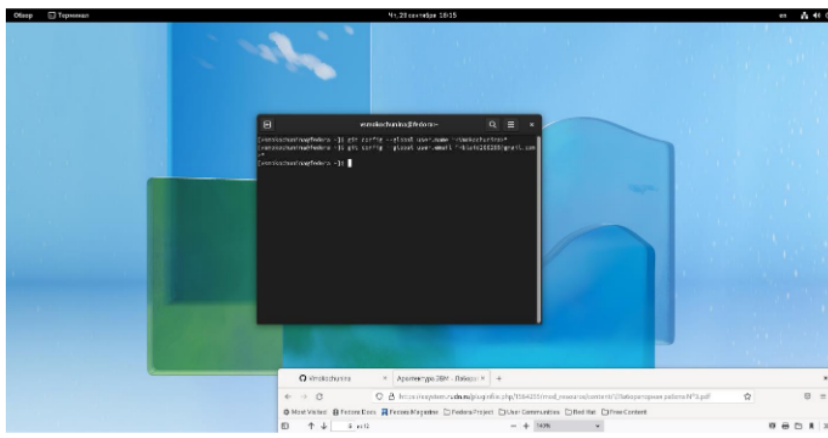
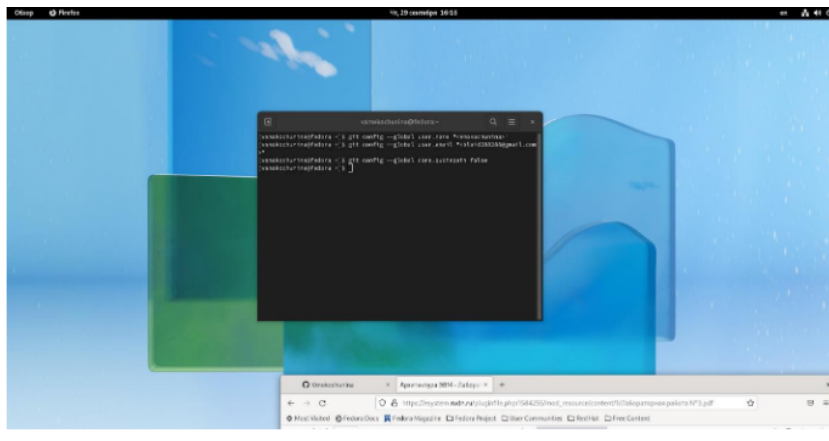


Рис. 3.1: Создание учетной записи, заполнение данных.

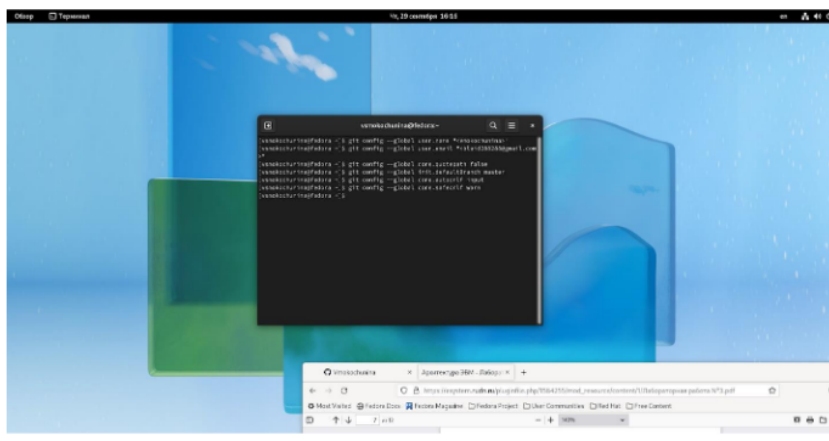
2. Я сделала предварительную конфигурацию git.



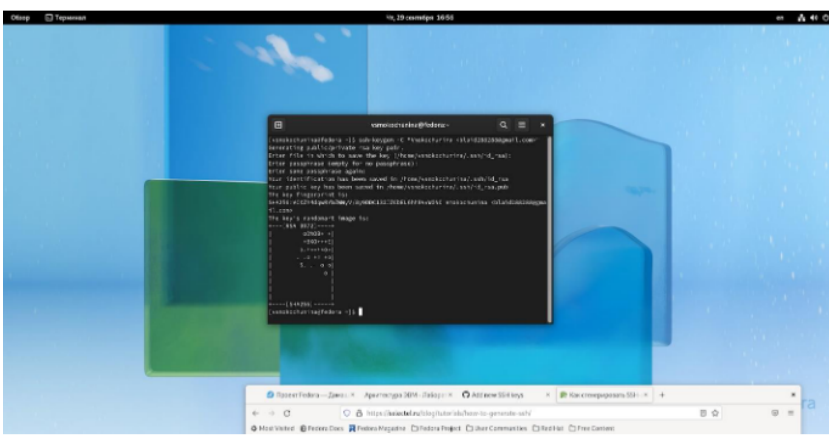
3. Я настроила utf-8 в выводе сообщений git.



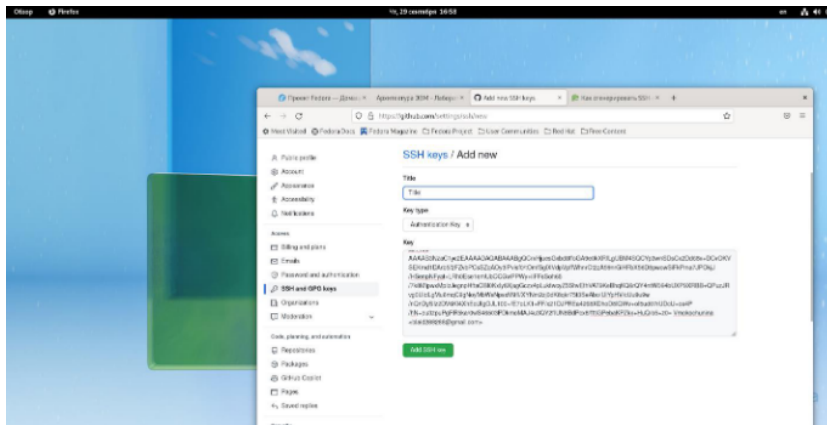
4. Я задала имя начальной ветки, параметр autocrlf и параметр safecrlf.



5. Я сгенерировала пару ключей (приватный и открытый).



6. Я загрузила сгенерённый открытый ключ.



7. Я сгенерировала ключ ргр.

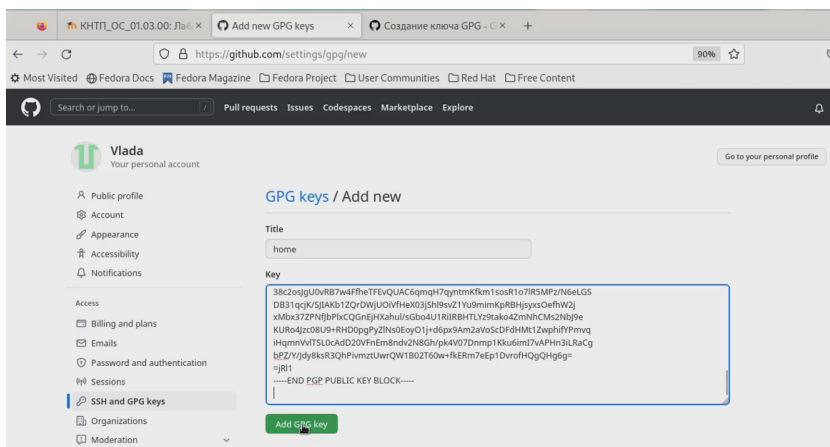
```
vsmokochunina@10:~ — gpg --full-generate-key
[vsmokochunina@10 ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.8; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Выберите тип ключа:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) 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
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0)
```

8. Я вывела сгенерированный ключ.

```
vsmokochunina@10:~  
DwqnpesjS/06fFs1kGerB46QQP8juGf64mFocJzPo/xT0XwQI18ucs7fa3bwSi2X  
vkt2Ch0jb3Yys1reAXztW8E79Zdq1IRcSVDFZ4q6ipD+ErKjHu7/2jsgppcQn4t6  
B5k0XvzDIV3LHXH4fpt+RcBYFuVWSjkAkpj5ATS0maI74ahueE5hxQF+LHYmHLZK  
epiknxG0U046ty9dILOuCF9B8i72xZgGyUwCp5UywdhZARZ9SzmXbtyvTld/8o  
/7UUD1+UTG1w030l8cPf7jSX02oA2/LbHU04140/Kddy3AvLdXJMMWd4JdRKUeSX  
gE1SyDfvxrHp/qaiwQfqc+NvvLp39PEaurTVpFevD0XZrGJVPMpMBwnV7LqkUpY8  
6e70nvfzSFeWL/DT/GkKUA5ApAmBMWaq7LJ2oqWV3tucFHIIm1B15fkaS/rpRapd  
i7K48kbwP6zS5RIM5KKY/3Mmp0fLuBR5M6pEnl6zS9vvC/+wvCF814QfFcCTGsD/  
BX38eXJnvntWYxgI134qgfbbT3AYnHChgq5ePdMyIorFHBcMhKR9vh3Zm5TEQAR  
AQABiQI2BBgBCAAgFiEEVRRbZHz3iF0BvYHVBM6dE+brpYkFamPsu7wCGwwACgkQ  
BM6dE+brpYkHaBAAkqFZ8ZTpa/tj24ULwumdKpLDFQ9YYvElb8XGF0y0Zc3unA8C  
sGJErzB8ypkRyX5xSVpoEydvkXxpIdGmofPMiXjesNeBI5ipk726K4oo48R0L9RC  
bXPymfTa4Q4Cadeoj/WKKXkZM9nRmY74mGr5BkxQ9det8XY1mbJMCBw/uv3Renc  
kxNtLfhRJAHA4p+3kRUzIrDfzvS0tTB0fbfRXcu9F3SyIyvabxysTFA164oSYqimy  
JhyvPpy8vaLQn/v64LC5qLwhc2XQKUm4RHRKD6L3MTYGLBt8dcM79RglegluZpc  
B8c2osJgU0vRB7w4FfheTFEUVQAC6qmQh7qyntmKfkm1sosR1o7lR5MPz/N6eLGS  
DB31qcjK/SJIAKb1ZQrDWjU0iVfHeX03jShl9svZ1Yu9mimKpRBHjsyxsoefhW2j  
xMb37ZPNfJbPlxQCGneJHXahul/sGbo4U1RiIRBHTLyz9tako4ZmNHCms2NbJ9e  
XUR04Jzc08U9+RHD0pgPyZlns0EoyO1j+d6px9Am2aVoScDFdHMT1ZwphiFYpvmvq  
iHqmnVvLTSLOcAdd20VFnm8ndv2N8Gh/pk4V07Dnmp1Kku6imI7vAPHn3iLRaCg  
bPZ/Y/Jdy8ksR3QhPivmztUwrQW1B02T60w+fKERM7eEp1DvrofHQgQHg6g=  
=jRl1  
-----END PGP PUBLIC KEY BLOCK-----  
vsmokochunina@10 ~]$
```

9. Добавление ключа pgr.



10. Я создала папку для будущей работы.

```
[vsmokochunina@10 ~]$ git config --global user.signingkey 1154516D91F7885381BD81  
D504CE9D13E6EBA589  
[vsmokochunina@10 ~]$ git config --global commit.gpgsign true  
[vsmokochunina@10 ~]$ git config --global gpg.program $(which gpg2)
```

11. Создание репозитория.

```
[vsmokochunina@10 ~]$ gh repo create study_2022-2023_os-intr  
--template=yamadharma/course-directory-student-template --public
```

Создание репозитория.]

12. Тут мы видим, что шаблон курса скопировался.

13. Удаление лишних файлов.

```
yasmokochunina@10 course-directory-student-template$ rm package.json
```

14. Создание необходимых каталогов.

```
[vsmokochunina@10 course-directory-student-template]$ echo os-intro > COURSE
[vsmokochunina@10 course-directory-student-template]$ make
```

15. Отправка файлов на сервер.

```

[+] vsmokochunina@10:~/work/study/2022-2023/Операционные ...
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech
pg
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008
numeric.csl
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_eqnos.
y
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos
py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_secnos
py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_tablen
s.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/_
nit__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/co
e.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/ma
n.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pa
docattributes.py
create mode 100644 project-personal/stage6/report/report.md
[vsmokochunina@10 course-directory-student-template]$ git push

```

#Контрольные

вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Контроль версий, также известный как управление исходным кодом, — это практика отслеживания изменений программного кода и управления ими. Системы контроля версий — это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Репозиторий - хранилище версий - в нем хранятся все документы вместе с историей их изменения и другой служебной информацией. Рабочая копия - копия проекта, связанная с репозиторием commit - сохранение изменений в репозитории
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion. Децентрализованные системы контроля версий — СКВ, которые позволяют клиенту полностью хранить у себя копию репозитория проекта. Примеры: Git, Mercurial, Bazaar или, например, Darcs.
4. Опишите действия с VCS при единоличной работе с хранилищем. Хранилище является разновидностью файл-сервера, однако не совсем обычного.
 - Хранилище запоминает каждое внесенное изменение: -любое изменение любого файла, -изменения в самом дереве каталогов, такие как добавление, удаление и реорганизация файлов и каталогов. •При чтении данных из хранилища клиент обычно видит только последнюю версию дерева файлов.
 - Клиент также имеет возможность просмотреть предыдущие состояния файловой системы. •Вопросы типа «Что содержал этот каталог в прошлую среду?», «Кто был последним, изменявшим этот файл, и какие вносились

изменения?»

5. Опишите порядок работы с общим хранилищем VCS.
6. Каковы основные задачи, решаемые инструментальным средством git? Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.
7. Назовите и дайте краткую характеристику командам git. Создание основного дерева репозитория: `git init` Получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull` Отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push` Просмотр списка изменённых файлов в текущей директории: `git status` Просмотр текущих изменений: `git diff` Сохранение текущих изменений: добавить все изменённые и/или созданные файлы и/или каталоги: `git add .` добавить конкретные изменённые и/или созданные файлы и/или каталоги: `git add имена_файлов` удалить файл и/или каталог из индекса репозитория

(при этом файл и/или каталог остаётся в локальной директории): `git rm` имена_файлов
Сохранение добавленных изменений: сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'` сохранить добавленные изменения с внесением комментария через встроенный редактор: `git commit` создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки` переключение на некоторую ветку: `git checkout имя_ветки`

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.
9. Что такое и зачем могут быть нужны ветви (branches)? Ветки нужны для того, чтобы программисты могли вести совместную работу над проектом и не мешать друг другу при этом. При создании проекта, Git создает базовую ветку. Она называется master веткой.
10. Как и зачем можно игнорировать некоторые файлы при commit? Чтобы игнорировать файл, для которого ранее был сделан коммит, необходимо удалить этот файл из репозитория, а затем добавить для него правило в .gitignore .

4 Выводы

Я научилась создавать базовую конфигурацию для работы с git, создавать ключи, настраивать подписи git, создавать локальный каталог для выполнения заданий по предмету.