

Отчёт по лабораторной работе №2

Управление версиями

Сиссе Мохамед Ламин

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	10
4	Контрольные вопросы	11
	Список литературы	15

List of Figures

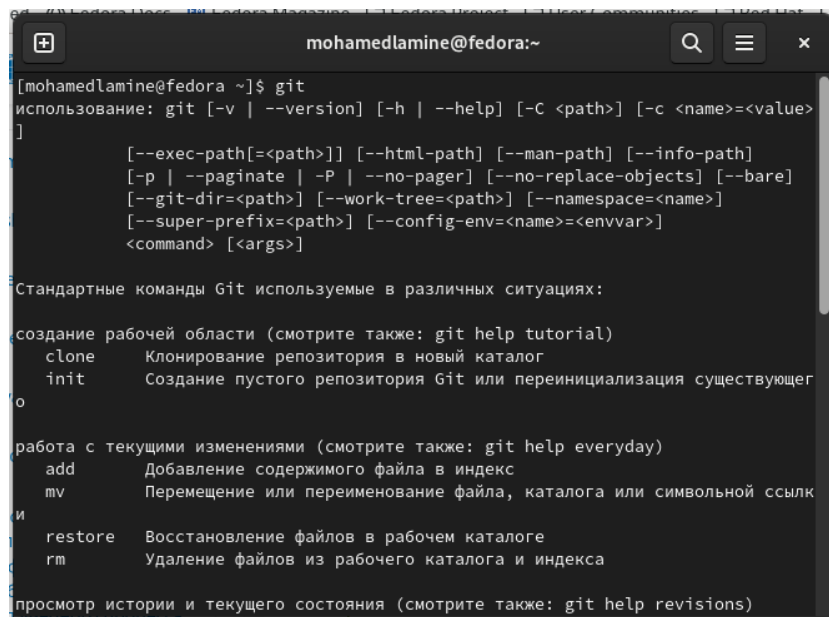
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	5
2.3	rsa-4096	6
2.4	ed25519	6
2.5	GPG ключ	7
2.6	GPG ключ	7
2.7	Параметры репозитория	7
2.8	Связь репозитория с аккаунтом	8
2.9	Загрузка шаблона	8
2.10	Первый коммит	9

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.



```
[mohamedlamine@fedora ~]$ git
использование: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
[<command>] [<args>]

--exec-path=<path>] [--html-path] [--man-path] [--info-path]
[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
[--super-prefix=<path>] [--config-env=<name>=<envvar>]
<command> [<args>]

Стандартные команды Git используемые в различных ситуациях:

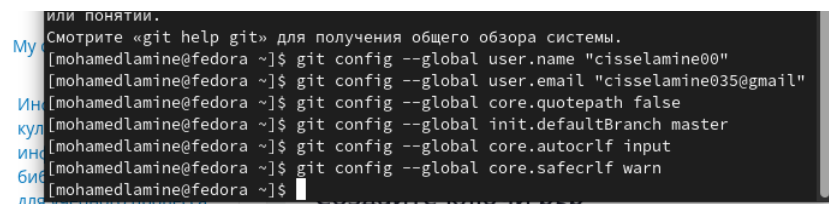
создание рабочей области (смотрите также: git help tutorial)
clone    Клонирование репозитория в новый каталог
init     Создание пустого репозитория Git или переинициализация существующего
о

работа с текущими изменениями (смотрите также: git help everyday)
add      Добавление содержимого файла в индекс
mv       Перемещение или переименование файла, каталога или символической ссылки
и
restore  Восстановление файлов в рабочем каталоге
rm       Удаление файлов из рабочего каталога и индекса

просмотр истории и текущего состояния (смотрите также: git help revisions)
```

Figure 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.



```
или понятии.
Можете посмотреть «git help git» для получения общего обзора системы.
[mohamedlamine@fedora ~]$ git config --global user.name "cisselamine00"
[mohamedlamine@fedora ~]$ git config --global user.email "cisselamine035@gmail"
Информация: [mohamedlamine@fedora ~]$ git config --global core.quotepath false
культуры: [mohamedlamine@fedora ~]$ git config --global init.defaultBranch master
информация: [mohamedlamine@fedora ~]$ git config --global core.autocrlf input
библиотека: [mohamedlamine@fedora ~]$ git config --global core.safecrlf warn
для: [mohamedlamine@fedora ~]$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```

[mohamedlamine@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mohamedlamine/.ssh/id_rsa):
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mohamedlamine/.ssh/id_rsa
Your public key has been saved in /home/mohamedlamine/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:GCyb2QIMtavVF4gH126B1Ef4onCCvsfRzCiv/dWG2kc mohamedlamine@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|..o.o+ o.|
|o =.o= .|
|.= +.++|
|.o=.+++|
|.o+@o+.S|
|= o.* oE|
|. = . o.o|
|..+ + ..|
|.o..o ..|
+---[SHA256]-----+
[mohamedlamine@fedora ~]$

```

Figure 2.3: rsa-4096

```

[mohamedlamine@fedora ~]$
[mohamedlamine@fedora ~]$
[mohamedlamine@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/mohamedlamine/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mohamedlamine/.ssh/id_ed25519
Your public key has been saved in /home/mohamedlamine/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:xdWrwecAAAPzAG9iTYGtqvm95FFLHc5oDCAe6DfAmoE mohamedlamine@fedora
The key's randomart image is:
+--[ED25519 256]--+
|..*oo... ..|
|.+.+. ... .|
|o.o.+ . oo .|
|E++o. . + o|
|o=.o . S *|
|+..o . . . .|
|... o = * .|
|o + . * +|
|.. +o .|
+---[SHA256]-----+
[mohamedlamine@fedora ~]$

```

Figure 2.4: ed25519

Создаем GPG ключ

```
"cisselamine00 <cisselamine035@gmail>"
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
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 (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/mohamedlamine/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/mohamedlamine/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/mohamedlamine/.gnupg/openpgp-revocs.d/8F
C7288751AFDEC18B62C2DE7B2D8A769FB8B46B.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2023-02-25 [SC]
      8FC7288751AFDEC18B62C2DE7B2D8A769FB8B46B
uid                               cisselamine00 <cisselamine035@gmail>
sub  rsa4096 2023-02-25 [E]

[mohamedlamine@fedora ~]$
```

Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

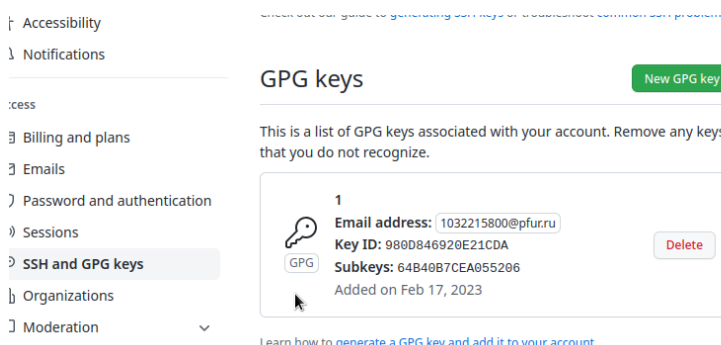


Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
[mohamedlamine@fedora ~]$
[mohamedlamine@fedora ~]$
[mohamedlamine@fedora ~]$ gpg --list-secret-keys --keyid-format LONG
/home/mohamedlamine/.gnupg/pubring.kbx
-----
sec  rsa4096/7B2D8A769FB8B46B 2023-02-25 [SC]
      8FC7288751AFDEC18B62C2DE7B2D8A769FB8B46B
uid                               [ абсолютно ] cisselamine00 <cisselamine035@gmail>
ssb  rsa4096/70CED9C5DFFDFA8C 2023-02-25 [E]

[mohamedlamine@fedora ~]$ git config --global user.signingkey 7B2D8A769FB8B46B
[mohamedlamine@fedora ~]$ git config --global commit.gpgsign true
[mohamedlamine@fedora ~]$ git config --global gpg.program $(which gpg2)
[mohamedlamine@fedora ~]$
```

Figure 2.7: Параметры репозитория

Настройка gh



Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

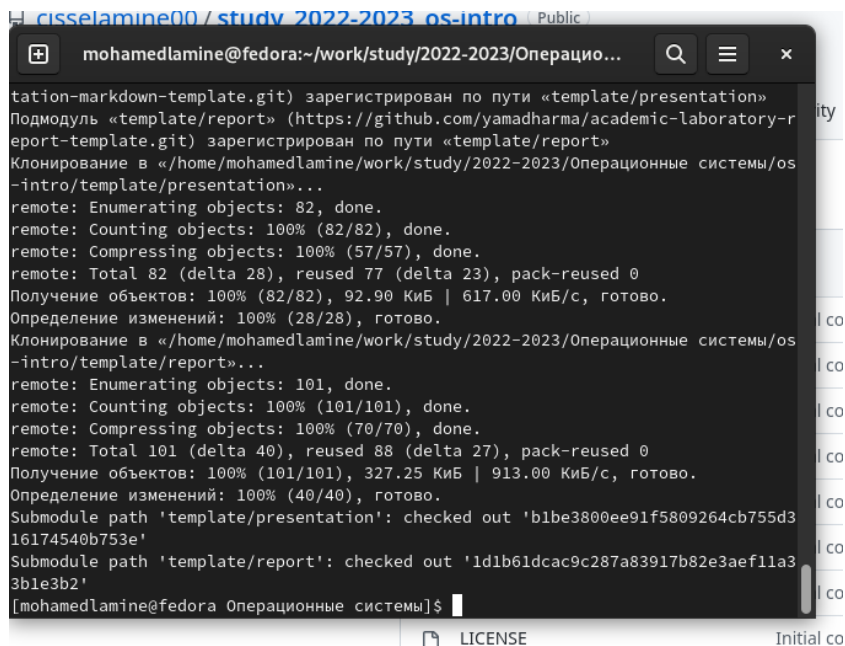
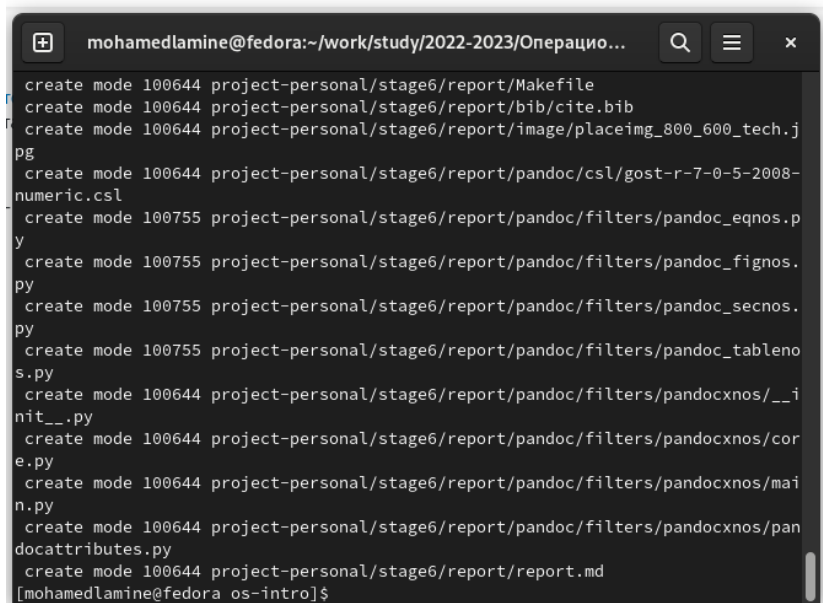


Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

A terminal window with a dark background and light text. The title bar shows the user 'mohamedlamine@fedora' and the current directory path. The terminal displays the output of a 'git commit' command, listing 15 files that were created. Each line shows the file mode (all are 'mode 100644'), the file path, and the repository path. The files are: Makefile, cite.bib, placeimg_800_600_tech.jpg, gost-r-7-0-5-2008-numeric.csl, pandoc_eqnos.py, pandoc_fignos.py, pandoc_secnos.py, pandoc_tablenos.py, pandocxnos__init__.py, pandocxnos_core.py, pandocxnos_main.py, pandocxnos_pandocattributes.py, and report.md. The prompt at the bottom is '[mohamedlamine@fedora os-intro]\$'.

```
mohamedlamine@fedora:~/work/study/2022-2023/Операцио...
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech.j
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.p
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_tableno
s.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__i
nit__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/cor
e.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/mai
n.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pan
docattributes.py
create mode 100644 project-personal/stage6/report/report.md
[mohamedlamine@fedora os-intro]$
```

Figure 2.10: Первый коммит

3 Вывод

Мы приобрели практические навыки работы с сервисом github.

4 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
 - хранилище - пространство на накопителе где расположен репозиторий
 - commit - сохранение состояния хранилища
 - история - список изменений хранилища (коммитов)
 - рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)
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.

Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

4. Опишите действия с VCS при единоличной работе с хранилищем.

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- git config - установка параметров
- git status - полный список изменений файлов, ожидающих коммита
- git add . - сделать все измененные файлы готовыми для коммита.
- git commit -m "[descriptive message]" - записать изменения с заданным сообщением.
- git branch - список всех локальных веток в текущей директории.
- git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.
- git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.
- git push - запустить текущую ветку в удаленную ветку.
- git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

- git remote add [имя] [url] — добавляет удалённый репозиторий с заданным именем;
- git remote remove [имя] — удаляет удалённый репозиторий с заданным именем;
- git remote rename [старое имя] [новое имя] — переименовывает удалённый репозиторий;
- git remote set-url [имя] [url] — присваивает репозиторию с именем новый адрес;

- `git remote show [имя]` — показывает информацию о репозитории.

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветвление — это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

10. Как и зачем можно игнорировать некоторые файлы при `commit`?

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить:

Список литературы

1. Лекция Системы контроля версий
2. GitHub для начинающих