

Отчёт по лабораторной работе №2

Управление версиями

Вероника Зиборова

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	17
4	Контрольные вопросы	18

Список иллюстраций

2.1	Загрузка пакетов	7
2.2	Параметры репозитория	8
2.3	rsa-4096	9
2.4	ed25519	10
2.5	GPG ключ	11
2.6	GPG ключ	12
2.7	Параметры репозитория	13
2.8	Связь репозитория с аккаунтом	14
2.9	Загрузка шаблона	15
2.10	Первый коммит	16

Список таблиц

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.

```

vnziborova@vnziborova:~$ git
использование: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
                  [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
                  [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
                  [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
                  [--work-tree=<path>] [--namespace=<name>] [--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

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

просмотр истории и текущего состояния (смотрите также: git help revisions)
bisect     Выполнение двоичного поиска коммита, который вносит ошибку
diff       Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
grep       Вывод строк, соответствующих шаблону
log        Вывод истории коммитов
show       Вывод различных типов объектов
status     Вывод состояния рабочего каталога

выращивание, маркировка и правка вашей общей истории
branch     Вывод списка, создание или удаление веток
commit     Запись изменений в репозиторий
merge      Объединение одной или нескольких историй разработки вместе
rebase     Повторное применение коммитов над вершущей другой ветки
reset      Сброс текущего состояния HEAD на указанное состояние
switch     Переключение веток
tag        Создание, вывод списка, удаление или проверка метки, подписанной с помощью GPG

совместная работа (смотрите также: git help workflows)
fetch      Загрузка объектов и ссылок из другого репозитория
pull       Извлечение изменений и объединение с другим репозиторием или локальной веткой
push       Обновление внешних ссылок и связанных объектов

```

Рис. 2.1: Загрузка пакетов

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

```
vnziborova@vnziborova:~$ git config --global user.name "1132246825"
vnziborova@vnziborova:~$ git config --global user.email "1132246825@rudn.university"
vnziborova@vnziborova:~$ git config --global core.quotepath false
vnziborova@vnziborova:~$ git config --global init.defaultBranch master
vnziborova@vnziborova:~$ git config --global core.autocrlf input
vnziborova@vnziborova:~$ git config --global core.safecrlf warn
vnziborova@vnziborova:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи


```

vnziborova@vnziborova:~$
vnziborova@vnziborova:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vnziborova/.ssh/id_rsa):
Created directory '/home/vnziborova/.ssh'.
Enter passphrase for "/home/vnziborova/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vnziborova/.ssh/id_rsa
Your public key has been saved in /home/vnziborova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:VDkcHJRFQvxTV5tDjzFFaSaTzF+/iWaTxKkKwG17rs vnziborova@vnziborova
The key's randomart image is:
+---[RSA 4096]-----+
|      . * @ 0 *  + + + |
|      . o B = * o B * |
|      . . 00.0 = + |
|      o  .0+ .0 |
|      . S  +.0 0 |
|      .  o * 0 |
|      . . 0 . |
|      .      |
|      E.      |
+---[SHA256]-----+
vnziborova@vnziborova:~$

```

Рис. 2.3: rsa-4096

```
vnziborova@vnziborova:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vnziborova/.ssh/id_ed25519):
Enter passphrase for "/home/vnziborova/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vnziborova/.ssh/id_ed25519
Your public key has been saved in /home/vnziborova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:hRuYE+HIfaBbzTNn/hc0vkF9WaUEQjdzkiSaJbMUNCU vnziborova@vnziborova
The key's randomart image is:
+--[ED25519 256]--+
|      +oE+*.Boo +|
|    . = 0 @ +.* oo|
|    + B % +  =.o|
|    o o X  + ..|
|    . S .  + |
|      .  + |
|      . o |
|      . |
|      |
+----[SHA256]-----+
vnziborova@vnziborova:~$
```

Рис. 2.4: ed25519

Создаем GPG ключ

```

vnziborova@vnziborova:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/vnziborova/.gnupg'
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) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: 1132246825
Адрес электронной почты: 1132246825@rudn.university
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "1132246825 <1132246825@rudn.university>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход?

```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```
vinziborova@vinziborova:~$  
vinziborova@vinziborova:~$  
vinziborova@vinziborova:~$ gpg --armor --export A45B44807844699B | xclip -sel clip  
vinziborova@vinziborova:~$ git config --global user.signingkey A45B44807844699B  
vinziborova@vinziborova:~$ git config --global commit.gpgsign true  
vinziborova@vinziborova:~$ git config --global gpg.program $(which gpg)  
vinziborova@vinziborova:~$
```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```

xu7WdWRjBVZrLMz90uVdEzR/j2F+D7mrjs0Cfykto/FQ2cZhrfcXQMA732z66dMj
04XQdrvcYQ3zoFLbbC7BaoGt6SHnWUngP5a0Cer0YxmvRCKeYgKw6oQI+MigDdjQ
CkHUF/2fRcwMc4HK79YTfbbi0vV2znIHDaIL26SOVV1xMDqACMTQ/9jrPNiQmpS9
N7UwLRQtXxxyvFFgsFZv4w+38QprJouUvwLXMYxiW0NlZRggTWf95aTDraes1ei
EX9m0TbEwWQMyIGBvVlqcnGPN50dtdzzgWI/hrs8qqZnsdgz4Reyl/OSD07NkPQq
e1ekIA0YLBULU4vNl3hysB/3b0xSa55hF0ho7JPJCM3LYRbAST7kPvTQHfUFG
ZIXrcEFLKhpyJIAgSRnGHgngcnHSf0AX0AMD7Vgm7rKfbEGVB/Fvgvm050BCOORV
GMJ4Mm4tpLjYyR4b4Fvv21RaZQ3rUFgZGey3/kRWZi8e5cqkn7VrdBZ0bXA9Cf+i
wpGvPI7eP8UAEQEAAyKCNgQYAQAIBYhBNUEJ9exi7i8FMVN7Arbqr14RAm7BQJn
ufc0AhsMAA0JEKRbqr14RAm7o3sP/3pYXgC1hi5/fuQ0JiXfrbjxfMpInLZevgy
vazhFLozSV20ycZ+UjtZvVWh6o6WX6Aq7eGN7HU/dcSm7OGALePTfhxXb+81f0IP
/nkT7MwgT2WQh2lkDVkWDuoyW9HKj6gxNSygKbH3snR9tamUmtx5N3m+5hpqZEc
Swc0d9kAjQydpSsHRL46sb3jsuyHw0u93EortvTf7kFEWbInTXs7KN9uAVr//cx8
qNnzLAV+T30T8Xma5gFKHyLMlnjX8Efl2frqh/lpsKawWCRzJRNzhPa/PtGAIEFQ
worT2fBd8EtBW9Zy53q+u0o3eph4JS0suV1VAWeCr8A6vRUt1KJSEw6uunVws0s
xCiWRkw6hKcVWgdLbKfDn1LP9GdF6E0pIcIUr5tFncHu36X1hw2Caccf6gOCG3hy
s1iN7JbXIXpVsXNyH3TtLJXsHmBJSXSAWcJE2BIHMrgpVXI2wS4QN5bBjdpMKov
zhAaKbV5QPajTuA80Lnf0Z6a19hV+MmJURuDnM5C2P7PvfgHoQAWd2ALZbynMtda
8zSTJdBnp4+qTKhsjXaqpeEwp2uqJ1QDpm3Wb1TeYp1k4ZZTQRnny00M3fFR6tbo
uyv0MVYveBT3oq1P92dDdUd9czGBJMKc3cQjQcNHbZguGbuoBseAKFXKQOR6dBme
CsDsw02N
=nmBQ
-----END PGP PUBLIC KEY BLOCK-----
vnziborova@vnziborova:~$ gpg --list-secret-keys --keyid-format LONG
[keyboxd]
-----
sec   rsa4096/A45BAABD784409BB 2025-02-22 [SC]
      D50427D7B18BB8BC14C54DEDA45BAABD784409BB
uid    [ абсолютно ] 1132246825 <1132246825@rudn.university>
ssb    rsa4096/3848EA24AF55DCD1 2025-02-22 [E]

vnziborova@vnziborova:~$

```

Рис. 2.7: Параметры репозитория

Настройка gh

```
vnziborova@vnziborova:~$  
vnziborova@vnziborova:~$ gh auth login  
? Where do you use GitHub? GitHub.com  
? What is your preferred protocol for Git operations on this host? SSH  
? Upload your SSH public key to your GitHub account? /home/vnziborova/.ssh/id_rsa.pub  
? Title for your SSH key: GitHub CLI  
? How would you like to authenticate GitHub CLI? Login with a web browser  
  
! First copy your one-time code: 2901-0474  
Press Enter to open https://github.com/login/device in your browser...  
✓ Authentication complete.  
- gh config set -h github.com git_protocol ssh  
✓ Configured git protocol  
✓ Uploaded the SSH key to your GitHub account: /home/vnziborova/.ssh/id_rsa.pub  
✓ Logged in as 1132246825  
vnziborova@vnziborova:~$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```

vznizborova@vznizborova: ~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
vznizborova@vznizborova: ~$ cd ~/work/study/2024-2025/"Операционные системы"
vznizborova@vznizborova: ~/work/study/2024-2025/"Операционные системы"$ gh repo create os-intro --template=panadharma/course-directory-student-template --public
/ Created repository 1132246825/os-intro on GitHub
https://github.com/1132246825/os-intro
vznizborova@vznizborova: ~/work/study/2024-2025/"Операционные системы"$ git clone --recursive git@github.com:1132246825/os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:0hY3mV16t0JhnpZisF/2LD002PMS-Hdkr4UvC0qI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?

```

Рис. 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```

create mode 100644 project-personal/stage6/presentation/presentation.md
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.jpg
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.py
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_tablenos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
vnziborova@vnziborova:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 38, готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (37/37), 342.27 КиБ | 2.36 МБ/с, готово.
Total 37 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:1132246825/os-intro.git
   fd83501..a448b63  master -> master
vnziborova@vnziborova:~/work/study/2024-2025/Операционные системы/os-intro$

```

Рис. 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: