

# **Отчёт по лабораторной работе №2**

**Управление версиями**

Бегенджов Гурбанмырат

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Вывод</b>	<b>17</b>
<b>4</b>	<b>Контрольные вопросы</b>	<b>18</b>

# 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.

```
gbeginzhov@gbeginzhov:~  
gbeginzhov@gbeginzhov:~$ 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
```

Рис. 2.1: Загрузка пакетов

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

```
gbegendzhov@gbegendzhov:~$  
gbegendzhov@gbegendzhov:~$ git config --global user.name "BegendzhovGurbanmirat"  
gbegendzhov@gbegendzhov:~$ git config --global user.email "1032244381@rudn.university"  
gbegendzhov@gbegendzhov:~$ git config --global core.quotepath false  
gbegendzhov@gbegendzhov:~$ git config --global init.defaultBranch master  
gbegendzhov@gbegendzhov:~$ git config --global core.autocrlf input  
gbegendzhov@gbegendzhov:~$ git config --global core.safecrlf warn  
gbegendzhov@gbegendzhov:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи



```

gbegendzhov@gbegendzhov:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gbegendzhov/.ssh/id_rsa):
Created directory '/home/gbegendzhov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gbegendzhov/.ssh/id_rsa
Your public key has been saved in /home/gbegendzhov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vRzX38e1Q5JjRTIdi/H38i34PFPKyEmXEIcEcqNSfJc gbegendzhov@gbegendzhov
The key's randomart image is:
+---[RSA 4096]-----+
|      .0 +0.*.0. |
|      ..+..E 0.. |
|      . . . + +. |
|      . . .0 0 |
|      S 0 .*000 |
|      . +0.*=* |
|      00.=.=B |
|      +0=.0 |
|      00 |
+----[SHA256]-----+
gbegendzhov@gbegendzhov:~$

```

Рис. 2.3: rsa-4096

```

gbegendzhov@gbegendzhov:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gbegendzhov/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gbegendzhov/.ssh/id_ed25519
Your public key has been saved in /home/gbegendzhov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:wNnHHnqirfoycdL4ZjXrK6w2P8xE4vM41V9YurNHJZ0 gbegendzhov@gbegendzhov
The key's randomart image is:
+--[ED25519 256]--+
|
|      . 0 .
|      + . = .
|      . . * E
|.00 . S =
|000000. = =
| +00 +0 =
| =*== = .
|0+++*=*0.+
+----[SHA256]-----+
gbegendzhov@gbegendzhov:~$

```

Рис. 2.4: ed25519

Создаем GPG ключ

```

    <p>w = срок действия ключа - n недель
    <p>m = срок действия ключа - n месяцев
    <p>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: BegendzhovGurbanmirat
Адрес электронной почты: 1032239631@rudn.university
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "BegendzhovGurbanmirat <1032239631@rudn.university>"

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 (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/gbegendzhov/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/gbegendzhov/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/gbegendzhov/.gnupg/openpgp-revocs.d/10A29EF828E344F7BD3834CA651A0B10A70CCF3A.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2025-02-18 [SC]
    10A29EF828E344F7BD3834CA651A0B10A70CCF3A
uid                               BegendzhovGurbanmirat <1032239631@rudn.university>
sub  rsa4096 2025-02-18 [E]

gbegendzhov@gbegendzhov:~$

```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```

gbegendzhov@gbegendzhov:~$ 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: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/651A0B10A70CCF3A 2025-02-18 [SC]
      10A29EF828E344F7BD3834CA651A0B10A70CCF3A
uid           [ абсолютно ] BegendzhovGurbanmirat <1032239631@rudn.university>
ssb   rsa4096/FF27BE9E834AC46C 2025-02-18 [E]

gbegendzhov@gbegendzhov:~$ gpg --armor --export 651A0B10A70CCF3A
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGe0T84BEADCsL5bMgLnmdpvEFK/rtDV6zep4Qdy2kowda+ij7rIz7RppnLY
MvUUM1fQoBqVOKQDeaOnxvFBTF61gb1ggQy5YBCVZXlysWcIlcP017vJbFrDKaP
eNo7Kg6ASZHe2JnzHKG2SiHmLuRRe3ZmhgKcbOB1cS3QED0DkFVQwGA5WwGbuea
e1yTcNYON1D8Fz5twTuKATA/bTWjttXhMTCAASl1LdDseDLvdTm70f1hT+q+miXh
HyC9LAF+DcdPauonP18DrkuTRO2Xc9hKHTPHPaLdBbEzw4+KzZU21GbdlT6vM
QdFxy+zyNoFifnxb0/T8NRJmPyHyR9N506JVE+LESkLioLkGB7s5mMQ7XLQpd2JT
dIgsVJCtv9hEAT+NCZ05eTgyi8D16uk7a5ckPLFbvcZdU4fUHblyzorC8faIv3KF
278uZ+NhkG5CAxbw2iStnJYQDHdr+P+kt8tTITtPxCMSh7Af26Qes6U5FVaHXP4u
+LKTPRoIfFIW7ptLa0ivJXo6rkxShWddJGAJ8Snp0VMY14jdc+YqI+IAXEHsuk2I
poA6FcuVcDMQW2DNaTKZQDTK1u8Bgq0jvQ1jTUB8j5Ic6aGB3CR5dLDrPIk/H8Tz
G7E/kbtlyIGr3QjRbJIVRZoYLZwJdmeBgb+xiXmkk6cCpPxzhT9j1wEAQARAQAB
tDJCZWdlbmR6aG92R3VyYmFubWlyYXQgPDEwMzIyMzY2MzY2MzY2MzY2MzY2MzY2
aXR5PokCUQOTAQgAOxYhBBcinvgo40T3vTg0vmlJaCxCnDM86BQJnte/OAhsDBQsJ

```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```

Ta3AkpbSQsbwWZCy1e7Ufz+BnkP687V+v3y/jhW38m7LHl3ijKoBJDe3oWYOTEw
s4W5FFl3ksnQpmtBRTUsPWHMCeDfwag2VBN01RA1q38kMm3Y67pTqh906301jEQJ
xc1C4JxH17U3palTIPTmu1rWB3iUuFJPDuw1Fm0oXz/cog2Kg/T/72Q2uEZA0AJA
taJ2Q+0wrHMTZss3X/fx6kbbcc0vs1Akc1eB12KgLaoOuNjYKXMiTj5ooVvXmTAON
bsQUa4iq9s9sX5n1CzzhqTIp2gpM7HKA+uGLcSi4/3fk4DMNpjhnSDduETcw9hDL
/5H2uhpQW9r1UTNTCAr3lnMSLbjta/wrcy/4rudwFR1/u1PipkffzsSk2HJrQ4w/
jgWpxVA3qoLISihUqVLas=
=I/oV
-----END PGP PUBLIC KEY BLOCK-----

gbegendzhov@gbegendzhov:~$
gbegendzhov@gbegendzhov:~$
gbegendzhov@gbegendzhov:~$ git config --global user.signingkey 651A0B10A70CCF3A
gbegendzhov@gbegendzhov:~$ git config --global commit.gpgsign true
gbegendzhov@gbegendzhov:~$ git config --global gpg.program $(which gpg2)
gbegendzhov@gbegendzhov:~$

```

Рис. 2.7: Параметры репозитория

Настройка gh

```
gbegendzhov@gbegendzhov:~$  
gbegendzhov@gbegendzhov:~$ 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/gbegendzhov/.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: 7AC4-C374  
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/gbegendzhov/.ssh/id\_rsa.pub  
✓ Logged in as BegendzhovGurbanmirat
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```

gbegendzhov@gbegendzhov:~$
gbegendzhov@gbegendzhov:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
gbegendzhov@gbegendzhov:~$ cd ~/work/study/2024-2025/"Операционные системы"
gbegendzhov@gbegendzhov:~/work/study/2024-2025/Операционные системы$ gh repo create os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository BegendzhovGurbanmirat/os-intro on GitHub
https://github.com/BegendzhovGurbanmirat/os-intro
gbegendzhov@gbegendzhov:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:
BegendzhovGurbanmirat/os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (36/36), 19.38 Киб | 9.69 Миб/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/gbegendzhov/work/study/2024-2025/Операционные системы/os-intro/template/presentation»...

```

Рис. 2.9: Загрузка шаблона

## Подготовка репозитория и коммит изменений

```
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
gbegendzhov@gbegendzhov:~/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 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (37/37), 342.28 КиБ | 2.38 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:BegendzhovGurbanmirat/os-intro.git
   95ff54d..07fc243  master -> master
gbegendzhov@gbegendzhov:~/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: