

Отчёт по лабораторной работе №2

Управление версиями

Скрипник Виктория Дмитриевна

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.

```

vdscriprik@vdscriprik:~$
vdscriprik@vdscriprik:~$ 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`)

<code>clone</code>	Клонирование репозитория в новый каталог
<code>init</code>	Создание пустого репозитория Git или переинициализация существующего

работа с текущими изменениями (смотрите также: `git help everyday`)

<code>add</code>	Добавление содержимого файла в индекс
<code>mv</code>	Перемещение или переименование файла, каталога или символической ссылки
<code>restore</code>	Восстановление файлов в рабочем каталоге
<code>rm</code>	Удаление файлов из рабочего каталога и индекса

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

<code>bisect</code>	Выполнение двоичного поиска коммита, который вносит ошибку
<code>diff</code>	Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
<code>grep</code>	Вывод строк, соответствующих шаблону
<code>log</code>	Вывод истории коммитов
<code>show</code>	Вывод различных типов объектов
<code>status</code>	Вывод состояния рабочего каталога

Рис. 2.1: Загрузка пакетов

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

```
vdskripnik@vdskripnik:~$ git config --global user.name "vdskripnik"
vdskripnik@vdskripnik:~$ git config --global user.email "1132242463@pfur.ru"
vdskripnik@vdskripnik:~$ git config --global core.quotepath false
vdskripnik@vdskripnik:~$ git config --global init.defaultBranch master
vdskripnik@vdskripnik:~$ git config --global core.autocrlf input
vdskripnik@vdskripnik:~$ git config --global core.safecrlf warn
vdskripnik@vdskripnik:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи


```

vdscripnik@vdscripnik:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vdscripnik/.ssh/id_rsa):
Created directory '/home/vdscripnik/.ssh'.
Enter passphrase for "/home/vdscripnik/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vdscripnik/.ssh/id_rsa
Your public key has been saved in /home/vdscripnik/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:q+abc27xGBolEulrT5Ye0NtfnFKnZUw9MTuQdo7il70 vdscripnik@vdscripnik
The key's randomart image is:
+---[RSA 4096]-----+
|          ..0  |
|          00.+ |
| .          ..+= |
| 0.        .0. .0 |
| ....  S...=0  |
| 0..+. 0+.*0 .  |
| 0*..+0==. .  |
| 0+ .,++0.  E  |
| . .00=*0      |
+---[SHA256]-----+
vdscripnik@vdscripnik:~$

```

Рис. 2.3: rsa-4096

```

vdscripnik@vdscripnik:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vdscripnik/.ssh/id_ed25519):
Enter passphrase for "/home/vdscripnik/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vdscripnik/.ssh/id_ed25519
Your public key has been saved in /home/vdscripnik/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:dFXmhJaTKGH3yrBvWw28U+wIXBm9sweejicLr0jZAMA vdscripnik@vdscripnik
The key's randomart image is:
+--[ED25519 256]--+
|0      0....=+  |
| E      ....0*+  |
| .      0.....0 |
| .      . = . 0 + |
| .      S 0 . Boo|
| +      . + + *0 |
| 0 . . = 0 *.0 |
| . . . + . + 0. |
| . 000   . .   |
+-----[SHA256]-----+
vdscripnik@vdscripnik:~$

```

Рис. 2.4: ed25519

Создаем GPG ключ

```

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: vdskrpnik
Адрес электронной почты: 1132242463@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "vdskrpnik <1132242463@pfur.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 (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/vdskrpnik/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/vdskrpnik/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/vdskrpnik/.gnupg/openpgp-revocs.d/F868C737F9F93634E3D22609A84C12227D6422E2
.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2025-06-11 [SC]
      F868C737F9F93634E3D22609A84C12227D6422E2
uid                               vdskrpnik <1132242463@pfur.ru>
sub   rsa4096 2025-06-11 [E]

vdskrpnik@vdskrpnik:~$

```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```

vdscripnik@vdscripnik:~$ 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/A84C12227D6422E2 2025-06-11 [SC]
      F868C737F9F93634E3D22609A84C12227D6422E2
uid           [ абсолютно ] vdscripnik <1132242463@pfur.ru>
ssb   rsa4096/E11881805D290977 2025-06-11 [E]

vdscripnik@vdscripnik:~$ gpg --armor --export A84C12227D6422E2
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGhJNZgBEACrKdrIFMUb+PjMmmv9rqJlRAjlhQ36DTQaIfnu1+DLynce+er
ntuDtUCBwGnk6p6sE/LkjSwytxdD7UuQwsaGn3jdt3dQE9vDQv+25aKfa8yDZqJN
7M/+5RJyM5ZHM7hmXfCUip/v9B2ZXmFTbchdecguHbh95vYKkVq41nU2JwtCXgn
X1DvaWzulMsTweDv09wyXz1Q5DAhVX2kUQarHX4URIvtFEaGtS8WxTR0ntd0fG9Z
ltwQDDzqlUtvXl/XeSqhPH0Qbja0HWET/w4NkfTiQhvCe03H/EjvJLE2xqf0CRbAt
/2Li47SUKhli8GnzwTMPL3veT9K8KH8jz97tIne1PxvrmV004i+sQuAsoILTwpPJ
6ZdwPLUP4epZRipvwfcDvmkUFLMuGLkf4fzYrAiPFk367s+ijKeAYSWzEs/eYQo
UerRu+kQG90CkL/vdp78Jfv85Rsd+vd38llobjBYWj0+IxyBjfKIxvGDQHy6aYvd
CGC5Xr/YhGcSb/1Hz9tOKrb2k8zgUKUrzpwJBd3DXYGcyEXsLCmz3pt34WZ3GH7f
R4chiBeZdHVeQLByTze37WuUXBwUuKxAlVW/mqDLuaBWDpsYI+wHRQNm/71WAJ4/
+b/yRLcD4Aka/MUGlUT2o/Vwrxn8yHU7fDf+lVYM5JJGb0fKK48LS2lwZwARAQAB
tB92ZHNrcmlwbmllrIDwxMTMyMjQyNDYzQHBmdXIucnU+iQJRBBMBCAA7FiEE+GjH
N/n5NjTj0iYJqEwSIn1kIuIFAmhJNZgCGwMFCwkIBwICIGFQoJCAscBBYCAwEC
HgCCF4AACgkQqEwSIn1kIuIvsQ//fu8AKEnG7p+3bXlBemj+Ug/6G944cqpII057

```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
A/Jtmuqm9IWLxAB0vAQ6oD1p0Mugd1cH52BWOX0XpQBfgDvWFr l0We2vTWMSaDSd
gJhMjcMBFV/QuBmPjYRNkLaewYQL/U04vMn8WCpDaGP/EeHsGaJHUbX00U+8yAiq
4paWm5zFIUcCCU7g0LHkE3vrZy/MfU2ZvWIVCQZb8JoUC5Zf2PSZG+bJMj94dVlr
CHHK7qW5mX9pECMUCsz1WZEu5rablG3W0oIZ8+/IJauIjbYfk15CYncezgwBqzj0
ayDtph/9RKPYiLzEiCnrI2BsXoBjD7LwtEdkiRRpF5iutkeDSlsarSPaSthxTc0P
aXQiVMUM1vHa9Vmm5/xQ3HEL4xYjgKQ8teBydB1fCE3x9FzU7GuAT2Kv3PWHaQ==
=HU0a
-----END PGP PUBLIC KEY BLOCK-----
vdscripnik@vdscripnik:~$
vdscripnik@vdscripnik:~$
vdscripnik@vdscripnik:~$ git config --global user.signingkey A84C12227D6422E2
vdscripnik@vdscripnik:~$ git config --global commit.gpgsign true
vdscripnik@vdscripnik:~$ git config --global gpg.program $(which gpg2)
vdscripnik@vdscripnik:~$
```

Рис. 2.7: Параметры репозитория

Настройка gh

```
vdskripnik@vdskripnik:~$ 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/vdskripnik/.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: DF4B-6ECF
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/vdskripnik/.ssh/id_rsa.pub
✓ Logged in as vdskripnik
vdskripnik@vdskripnik:~$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
Получение объектов: 100% (36/36), 19.43 КиБ | 19.43 МБ/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/vdskripnik/work/study/2024-2025/Операционные системы/os-intro/template/presentation»...  
remote: Enumerating objects: 147, done.  
remote: Counting objects: 100% (147/147), done.  
remote: Compressing objects: 100% (100/100), done.  
remote: Total 147 (delta 55), reused 131 (delta 39), pack-reused 0 (from 0)  
Получение объектов: 100% (147/147), 2.64 МиБ | 8.32 МБ/с, готово.  
Определение изменений: 100% (55/55), готово.  
Клонирование в «/home/vdskripnik/work/study/2024-2025/Операционные системы/os-intro/template/report»...  
remote: Enumerating objects: 195, done.  
remote: Counting objects: 100% (195/195), done.  
remote: Compressing objects: 100% (133/133), done.  
remote: Total 195 (delta 86), reused 161 (delta 52), pack-reused 0 (from 0)  
Получение объектов: 100% (195/195), 756.76 КиБ | 2.78 МБ/с, готово.  
Определение изменений: 100% (86/86), готово.  
Submodule path 'template/presentation': checked out '645759e4b104e93753637dedf8109adf24d071b7'  
Submodule path 'template/report': checked out 'b5a97ed1ef3e6f462109b7d03d82339ccaf27ea6'  
vdskripnik@vdskripnik:~/work/study/2024-2025/Операционные системы$
```

Рис. 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/_quarto.yml
create mode 100644 project-personal/stage6/presentation/_resources/image/logo_rudn.png
create mode 100644 project-personal/stage6/presentation/image/kulyabov.jpg
create mode 100644 project-personal/stage6/presentation/presentation.qmd
create mode 100644 project-personal/stage6/report/.gitignore
create mode 100644 project-personal/stage6/report/.projectile
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/_quarto.yml
create mode 100644 project-personal/stage6/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/solvay.jpg
create mode 100644 project-personal/stage6/report/report.qmd
vdscripnik@vdscripnik:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 35, готово.
Подсчет объектов: 100% (35/35), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (21/21), готово.
Запись объектов: 100% (34/34), 699.13 Киб | 4.79 Миб/с, готово.
Total 34 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:vdscripnik/os-intro.git
   cb7cfd4..be3b671  master -> master
vdscripnik@vdscripnik:~/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: