

Отчёт по лабораторной работе №2

Управление версиями

Карагизов Анастас Сергеевич НБИбд-01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	11
4	Контрольные вопросы	12
	Список литературы	16

List of Figures

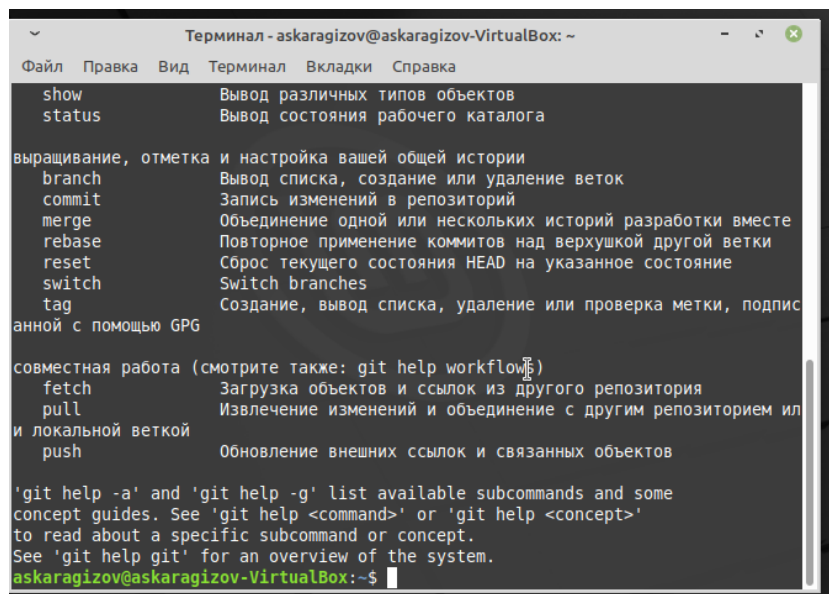
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	6
2.3	rsa-4096	6
2.4	ed25519	7
2.5	GPG ключ	7
2.6	GPG ключ	8
2.7	Параметры репозитория	8
2.8	Связь репозитория с аккаунтом	9
2.9	Загрузка шаблона	9
2.10	Первый коммит	10

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.



```
Терминал - askaragizov@askaragizov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

show      Вывод различных типов объектов
status    Вывод состояния рабочего каталога

выращивание, отметка и настройка 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 верхушкой другой ветки
reset      Сброс текущего состояния HEAD на указанное состояние
switch     Switch branches
tag        Создание, вывод списка, удаление или проверка метки, подпись
           анной с помощью GPG

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

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
askaragizov@askaragizov-VirtualBox:~$
```

Figure 2.1: Загрузка пакетов

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

```
Терминал - askaragizov@askaragizov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

reset      Сброс текущего состояния HEAD на указанное состояние
switch     Switch branches
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 репозиториум или локальной веткой
push       Обновление внешних ссылок и связанных объектов

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
askaragizov@askaragizov-VirtualBox:~$
askaragizov@askaragizov-VirtualBox:~$ git config --global user.name "AnastasKaragizov"
askaragizov@askaragizov-VirtualBox:~$ git config --global user.email "1032217041@pfur.ru"
askaragizov@askaragizov-VirtualBox:~$
askaragizov@askaragizov-VirtualBox:~$ git config --global core.quotepath false
askaragizov@askaragizov-VirtualBox:~$ git config --global init.defaultBranch master
askaragizov@askaragizov-VirtualBox:~$ git config --global core.autocrlf input
askaragizov@askaragizov-VirtualBox:~$ git config --global core.safecrlf warn
askaragizov@askaragizov-VirtualBox:~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
Терминал - askaragizov@askaragizov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

askaragizov@askaragizov-VirtualBox:~$ git config --global core.safecrlf warn
askaragizov@askaragizov-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/askaragizov/.ssh/id_rsa):
Created directory '/home/askaragizov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/askaragizov/.ssh/id_rsa
Your public key has been saved in /home/askaragizov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jotBRqeBVhcTN8ijcYF/0wJtZnc4EYmgUcyN2d6Q+7A askaragizov@askaragizov-VirtualBox
The key's randomart image is:
+---[RSA 4096]---+
|.++0*B+o|
|=B.@o.|
+ oB.= o
|.o+X *
+= *S+
o Eo+
. ....
o .
+---[SHA256]-----+
askaragizov@askaragizov-VirtualBox:~$
```

Figure 2.3: rsa-4096

```
Терминал - askaragizov@askaragizov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
+-----[SHA256]-----+
askaragizov@askaragizov-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/askaragizov/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/askaragizov/.ssh/id_ed25519
Your public key has been saved in /home/askaragizov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256: idRXy9qppZRimh4zhkRipvahUWlbTAK2PTkwQEkuxlQ askaragizov@askaragizov-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|o=BE0.o      +|
|+o=o0 .  * o |
|O*..+ + + = +|
|+ +o.o = = + |
| . . 0 S o   |
|    o +      |
|              |
+-----[SHA256]-----+
askaragizov@askaragizov-VirtualBox:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
Терминал - askaragizov@askaragizov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
"AnastasKaragizov <1032217041@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/askaragizov/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ 072F5459372F6150 помечен как абсолютно доверенный
gpg: создан каталог '/home/askaragizov/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/askaragizov/.gnupg/openpgp-revocs.d/835A4595B6DFA530A1E936D6072F5459372F6150.rev'
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-04-21 [SC]
      835A4595B6DFA530A1E936D6072F5459372F6150
uid     AnastasKaragizov <1032217041@pfur.ru>
sub   rsa4096 2022-04-21 [E]

askaragizov@askaragizov-VirtualBox:~$
```


Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

GPG keys

[New GPG key](#)

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



Email address: 1032217041@pfur.ru

Key ID: 072F5459372F6150

Subkeys: 9521E89E80E2F502

Added on 21 Apr 2022

Delete

[Learn how to generate a GPG key and add it to your account.](#)

Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
Терминал - askaragizov@askaragizov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

8a8VR9AnXzh3A1Y8r7oNRmUDXAazZ2eetu0tG4x0ajJdBnQD8Yao119ZwBXT4MCD
URoJABEBAGJAjYEGAeKACAWIQSDWkwVtt+LMKHpNtYHL1RZNy9hUAUCymFLXATb
DAACKRAHL1RZNy9hULmtEACbksIxXdynLBz09VpQaWhqRMT2sagQg36Jk/cV7K2x
P10n+v5fmmNNbc6uqS3pRY7F+K0nnDzsmRbMd80ay8LE62rw8SP0wKp6f0hpwbrL
dnSUEEjIDVj0iVzuq+JP25Sxa/U0071cGJfDxTLR8Mt8WV9JbXHLQYLByGApV+dm
AgVUVpumu1ta5CUTntuG3r0aLuKCzoavU0j3F7IohZvRiwUTqvUtuJhCjRHjswUo
rX++ZzXt8t3SivTW0fHI+rcmKUHCeMH4bzWrt8kkNmzYGwvjDNRaSzCZgEwvEvaC
usCRm8YTwaH8+eTcumS44Hh7FhicZLTejQJJKHEN7eSpXlfpLnVN2WgfDyGll0ev
l2f0nL3DPImKrrP9chrz7bWtH7fTEMBE0hj7+oNRkvKLrPiDgYyvTddVJNKdrl6J
+V1VMxGjHNzUX0uF9Pm/fwbxnoK+hXF6mim/CRIkhesRAN3TveFvBsk8vLBe4dt
YZvI36smVMoce/y3Gy0SFEIqgRPYf+Q5jQzrEUZMEVGkuFk3t0IP0HaNL9Pam3o/
Tc0NYHuN4HDi+kP6745whK4RegLoSK97+h6Fs0bpf4Jx9fKEtqs0AJRyf104pu
mF4mNsFk6rIGI9+Lq3I9mMxfoQ6J30ZtfKFD5YbaVrAniLFB/kZCfrwLh8bJ6ltj
2A==
-----END PGP PUBLIC KEY BLOCK-----
askaragizov@askaragizov-VirtualBox:~$
askaragizov@askaragizov-VirtualBox:~$
askaragizov@askaragizov-VirtualBox:~$
askaragizov@askaragizov-VirtualBox:~$
askaragizov@askaragizov-VirtualBox:~$ git config --global user.signingkey 072F5459372F6150
askaragizov@askaragizov-VirtualBox:~$ git config --global commit.gpgsign true
askaragizov@askaragizov-VirtualBox:~$ git config --global gpg.program $(which gpg2)
askaragizov@askaragizov-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

Настройка gh


```
askaragizov@askaragizov-VirtualBox:~$ git config --global user.signingkey 072F5459372F6150
askaragizov@askaragizov-VirtualBox:~$ git config --global commit.gpgsign true
askaragizov@askaragizov-VirtualBox:~$ git config --global gpg.program $(which gpg2)
askaragizov@askaragizov-VirtualBox:~$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/askaragizov/.ssh/id_rsa.pub
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 78EF-B7B5
Press Enter to open github.com 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/askaragizov/.ssh/id_rsa.pub
! Logged in as AnastasKaragizov
askaragizov@askaragizov-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
Терминал - askaragizov@askaragizov-VirtualBox: ~/work/study/2021-2022/Операционные сист
Файл Правка Вид Терминал Вкладки Справка
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 КиБ | 4.16 МиБ/с, готово.
Определение изменений: 100% (2/2), 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/askaragizov/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 42 (delta 9), reused 40 (delta 7), pack-reused 0
Клонирование в «/home/askaragizov/work/study/2021-2022/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Подмодуль по пути «template/presentation»: забрано состояние «3eae7b586f8a9aded2b506cd1018e625b228b93»
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»
askaragizov@askaragizov-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
Терминал - askaragizov@askaragizov-VirtualBox: ~/work/study/2021-2022/Операционные сист
Файл  Правка  Вид  Терминал  Вкладки  Справка
delete mode 100644 package.json
create mode 100644 project-personal/stage{1..6}/presentation/Makefile
create mode 100644 project-personal/stage{1..6}/presentation/presentation.md
create mode 100644 project-personal/stage{1..6}/report/Makefile
create mode 100644 project-personal/stage{1..6}/report/bib/cite.bib
create mode 100644 project-personal/stage{1..6}/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage{1..6}/report/pandoc/csl/gost-r-7-0-5-2008-numeric.c
sl
create mode 100644 project-personal/stage{1..6}/report/report.md
create mode 100644 structure
askaragizov@askaragizov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git p
ush
Warning: Permanently added the ECDSA host key for IP address '140.82.121.3' to the list of kno
wn hosts.
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (19/19), 266.47 Киб | 2.42 Миб/с, готово.
Всего 19 (изменения 2), повторно использовано 0 (изменения 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:AnastasKaragizov/study_2021-2022_os-intro.git
c2ecdfa..cb0d507 master -> master
askaragizov@askaragizov-VirtualBox:~/work/study/2021-2022/Операционные системы/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