

# **Отчёт по лабораторной работе №2**

**Управление версиями**

Полвонов Нуриддин Абдуджалилович НБИбд-01-21

# Содержание

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

# List of Figures

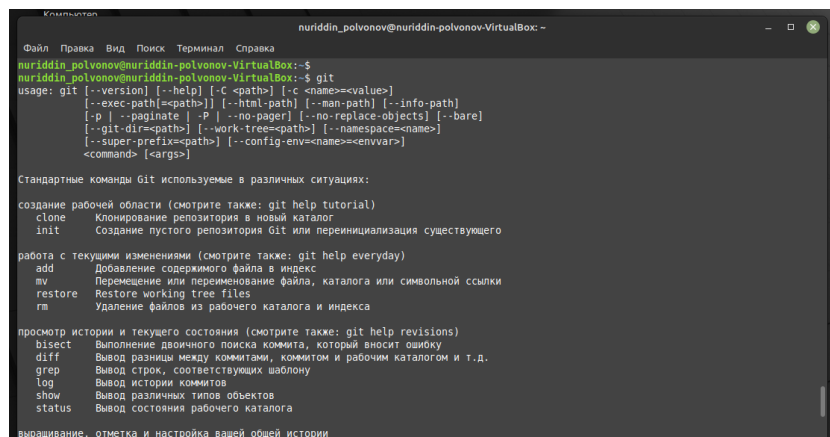
2.1	Загрузка пакетов . . . . .	5
2.2	Параметры репозитория . . . . .	5
2.3	rsa-4096 . . . . .	6
2.4	ed25519 . . . . .	6
2.5	GPG ключ . . . . .	7
2.6	GPG ключ . . . . .	7
2.7	Параметры репозитория . . . . .	8
2.8	Связь репозитория с аккаунтом . . . . .	8
2.9	Загрузка шаблона . . . . .	8
2.10	Первый коммит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



```
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git --help
usage: git [-version] [--help] [-C <path>] [-c <name>=<value>]
          [--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

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 Restore working tree files
  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

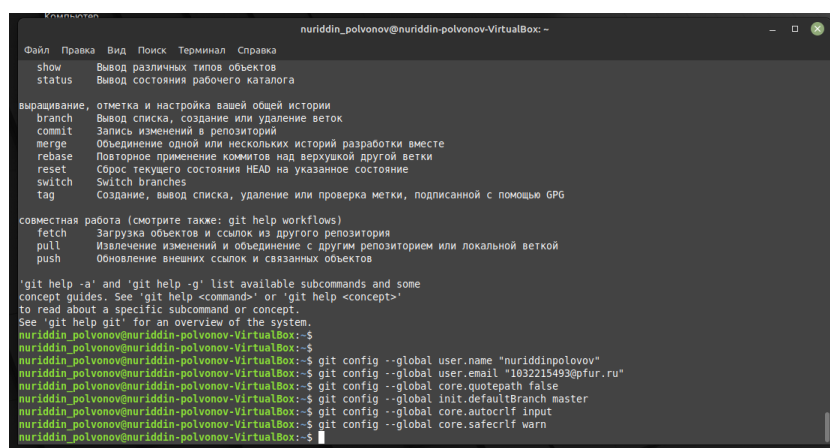
выращивание, отметка и настройка вашей общей истории
  branch Вывод списка, создание или удаление веток
  commit Запись изменений в репозиторий
  merge  Объединение одной или нескольких историй разработки вместе
  rebase Повторное применение коммитов над вершней другой ветки
  reset  Сброс текущего состояния 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 репозиторием или локальной веткой
  push   Обновление внешних ссылок и связанных объектов

'git help -a' и '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.
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$
```

Figure 2.1: Загрузка пакетов

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



```
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global user.name "nuriddinpolovov"
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global user.email "1032215493@pfur.ru"
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global core.quotepath false
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global init.defaultBranch master
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global core.autocrlf input
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global core.safecrlf warn
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
nuriddin_polvonov@nuriddin-polvonov-VirtualBox: ~  
File Правка Вид Поиск Терминал Справка  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global user.email "1692215493@pfur.ru"  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global core.quotepath false  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global init.defaultBranch master  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global core.autocrlf input  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ git config --global core.safecrlf warn  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ ssh-keygen -t rsa -b 4096  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/nuriddin_polvonov/.ssh/id_rsa):  
Created directory /home/nuriddin_polvonov/.ssh.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/nuriddin_polvonov/.ssh/id_rsa  
Your public key has been saved in /home/nuriddin_polvonov/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:UWfV46ACKUN2qzElFqgsw/8WtBaAPf+U1TyIlrgpRY nuriddin_polvonov@nuriddin-polvonov-VirtualBox  
The key's randomart image is:  
+--[RSA 4096]-----+  
|  o...E..+..+o..o |  
|==..+..+ o . o . |  
|++..+..+ . . . |  
|o=..+..+ |  
|+o..+ S |  
|o=..o |  
| = . |  
| + o |  
| + |  
+----[SHA256]-----+  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$
```

Figure 2.3: rsa-4096

```
nuriddin_polvonov@nuriddin-polvonov-VirtualBox: ~  
File Правка Вид Поиск Терминал Справка  
| o = . o . |  
| = . |  
| + o |  
| + |  
+----[SHA256]-----+  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$ ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/nuriddin_polvonov/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/nuriddin_polvonov/.ssh/id_ed25519  
Your public key has been saved in /home/nuriddin_polvonov/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:5n3ynqaeht+Re100wZw/hgrc8+v15Vgt1r6cWcNrRU nuriddin_polvonov@nuriddin-polvonov-VirtualBox  
The key's randomart image is:  
+--[ED25519 256]--+  
|  o..o |  
| = |  
| . . .o |  
| o o . o+ |  
| S . + +Eo |  
| o . o .o=0 |  
| . + +.oX |  
| . +.oB |  
| .+X=.o |  
+----[SHA256]-----+  
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
Компьютер
nuriddin_polvonov@nuriddin-polvonov-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: nuriddinpolvonov
Адрес электронной почты: 1032215493@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"nuriddinpolvonov <1032215493@pfur.ru>"

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/nuriddin_polvonov/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ B33054608A456983 помечен как абсолютно доверенный
gpg: создан каталог "/home/nuriddin_polvonov/.gnupg/openpgp-revocs.d"
gpg: сертификат отзыва записан в "/home/nuriddin_polvonov/.gnupg/openpgp-revocs.d/690502E946CA3A3B40B273A9B33054608A456983.rev"
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-08-28 [SC]
      690502E946CA3A3B40B273A9B33054608A456983
uid
sub   rsa4096 2022-08-28 [E]

nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~$
```

Figure 2.5: GPG ключ

## Добавляем GPG ключ в аккаунт

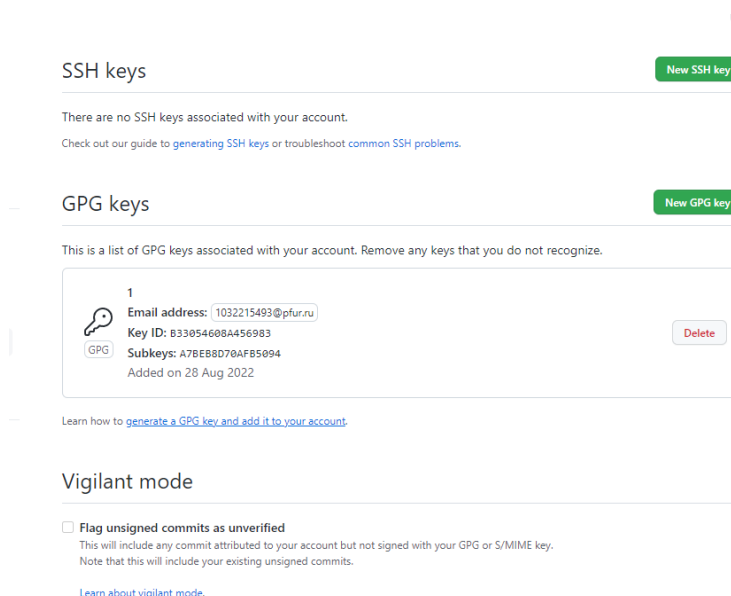


Figure 2.6: GPG ключ

## Настройка автоматических подписей коммитов git

```
nuriddin_polonov@nuriddin-polonov-VirtualBox: ~$
/ta+homEpbGHRyHdX6gPurb50JzdlJ8BstMy+Id9KMBU3155LYV4U9987cNL7M
F5Ush0v00507Lz93VotBf1s5mlr0zVxmap+pxHZcPhR+Sz/Dy2u6Ykba0/V59+zj
LYmoD3jFEaZdcZFL+ozECly+E3+Y+VRz9FarRgGdJFXQZ2TTeoN8wSx0n4Dw
laldQWZ03MwRwR0r1237Wd0z18T5b3C=mdert0B8K0v+H8ad/h82Ng4Wb
M2gaF5fSUCoMag6d5XJpEG3WtAl0/8/yENZq8bPrYhd9Hpp0BwUDWrb20Vv
eQhLz5w30YmsJauCqgDhXLM1s1MNIE8:IEz7Yb/0lyj8KH5eHYTKAKLx1Xp8
jKhtjBXZw+acDfahM070YpKf1WfN25Xfup2125CefFqg9uh9ZJ24y3HA/9Z5NU
+U7fC91h1P0h0Wm1+u69j3/cdM021r7h0xCUL7h9bEa1J9K0CJug3LKW7y
koIAEIEAAYKcNg0YA0a1BYR8GnVau1Gy07QLJ2q0MwGCKRm0BQJ1CzVeAhS
AA0JELWwGCKRm0hrgp/2yTMBWsJ38vGw0ZEpU06vCEVXx/F8db920307pL7w
BhBt/EUFKOAIN1h1wASvZ8P1db07OTxM21JnPasodFr11/9ya4EnLxYZ/uN8p+
bmNgQ2ZdyufP01m+E3aTP3LogF7HMTa8GhLMW20PChmVzWwWf31Ln/PUir
3v6pRS02XhZus+qAc0Em0d/2MLT20VuosXAVB2KxVOMW9hcg50VRCF9
fb+QME8XFXDxcBBMvAK2TMYvSS20LUfwg28o5Mxtd0F6+dFhL00J2No9v4qzU
sPlv8ynkPA1vjsegXUR95tn/Gp0310ybG0dg/brmLWULV6z20Ac/G2scrn+Ahhu
Rqls80eSpFN3jxm6W0oSRAHS+aIdb5EflGx0Xhp2243ghUB2vkn2IEgF97P+I6w
U9pELkfh2UXR+z1F0D121T6pdxIopw7gl0ozDR0py0vAU2Hc13v9y0n1/vB
oDlp4mky4c6p8n0sl0PM1J+2jvG5QV/MpK8Nc1pChas3DrjBw1W/S40LjX
4oSUNW1h1CnEtGbc24JeeCeIWyvPccAAA7cwVOEU+83bnYxRtF6gnW5Lpp0Xz
DJHqQ3LYVPE40/S05d1FYF+KAC2ND0FH3D6edJDN9rzQZUwQ2Kh/D350A+WNHG
=hoql
-----END PGP PUBLIC KEY BLOCK-----
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ git config --global user.signingkey B33054608A456983
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ git config --global commit.gpgsign true
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ git config --global gpg.program $(which gpg2)
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

## Настройка gh

```
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ git config --global user.signingkey B33054608A456983
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ git config --global commit.gpgsign true
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ git config --global gpg.program $(which gpg2)
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$ 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/nuriddin_polonov/.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: D460-76E8
Press Enter to open github.com in your browser...
2022-08-28T09:31:36Z ERROR viaduct::backend:ffl Missing HTTP status
2022-08-28T09:31:36Z ERROR viaduct::backend:ffl Missing HTTP status

Authentication complete.
- gh config set -h github.com git protocol ssh
Configured git protocol
Uploaded the SSH key to your GitHub account: /home/nuriddin_polonov/.ssh/id_rsa.pub
Logged in as nuriddinpolonov
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
nuriddin_polonov@nuriddin-polonov-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

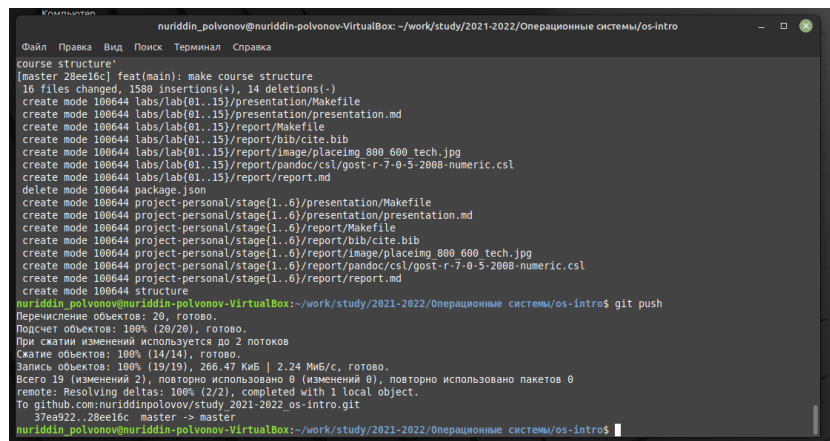
## Загрузка шаблона репозитория и синхронизация

```
nuriddin_polonov@nuriddin-polonov-VirtualBox: ~/work/study/2021-2022/Операционные системы
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: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 Кб | 3.12 Мб/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/nuriddin_polonov/work/study/2021-2022/Операционные системы/os-intro/template/presentation»:
remote: Enumerating objects: 58, done.
remote: Counting objects: 100% (58/58), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 58 (delta 17), reused 55 (delta 14), pack-reused 0
Получение объектов: 100% (58/58), 82.94 Кб | 20.73 Мб/с, готово.
Определение изменений: 100% (17/17), готово.
Клонирование в «/home/nuriddin_polonov/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
Получение объектов: 100% (78/78), 292.27 Кб | 1.79 Мб/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '3aebbb7586f8a9aded2b56ccdd1018e625b228b93'
Submodule path 'template/report': checked out 'df7b2ef80f8de3ab9a9ef8695277469a1a7842a'
nuriddin_polonov@nuriddin-polonov-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона



## Подготовка репозитория и коммит изменений



```
nuriddin_polvonov@nuriddin-polvonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro$ git push
course structure'
[master 28ee16c] feat(main): make course structure
16 files changed, 1580 insertions(+), 14 deletions(-)
create mode 100644 labs/lab(01..15)/presentation/Makefile
create mode 100644 labs/lab(01..15)/presentation/presentation.md
create mode 100644 labs/lab(01..15)/report/Makefile
create mode 100644 labs/lab(01..15)/report/bib/cite.bib
create mode 100644 labs/lab(01..15)/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab(01..15)/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab(01..15)/report/report.md
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.csl
create mode 100644 project-personal/stage(1..6)/report/report.md
create mode 100644 structure
nuriddin_polvonov@nuriddin-polvonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (19/19), 266.47 Киб | 2.24 Миб/с, готово.
Всего 19 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:nuriddinpolovov/study_2021-2022_os-intro.git
  37ee922..28ee16c master -> master
nuriddin_polvonov@nuriddin-polvonov-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