

Отчёт по лабораторной работе №2

Управление версиями

Карими Мохаммад Валид НФИбд-02-21

Содержание

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

List of Figures

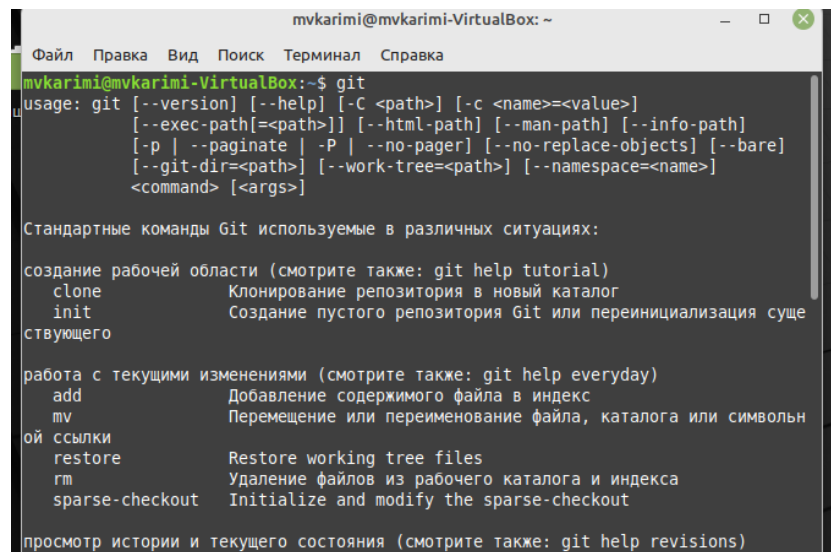
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	6
2.3	rsa-4096	6
2.4	ed25519	7
2.5	GPG ключ	7
2.6	GPG ключ	8
2.7	Параметры репозитория	8
2.8	Связь репозитория с аккаунтом	9
2.9	Загрузка шаблона	9
2.10	Первый коммит	10

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

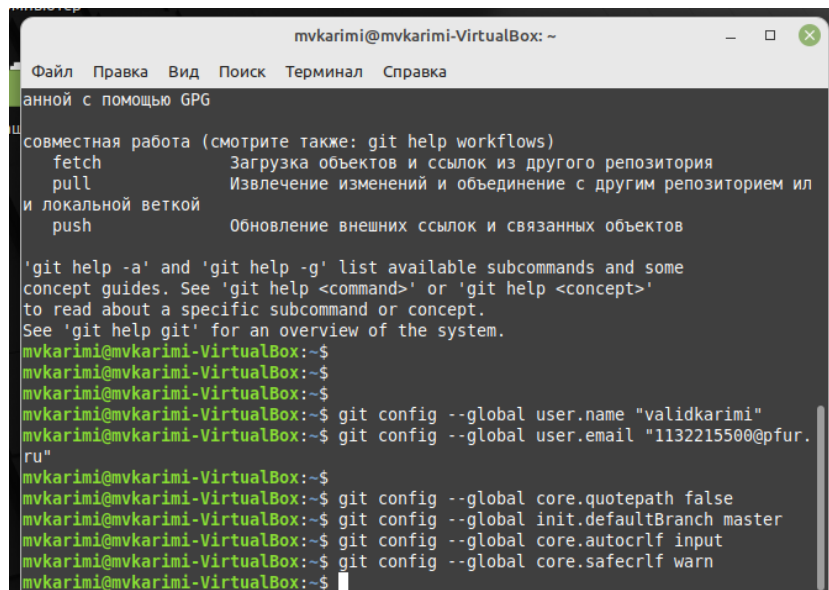
Устанавливаем git, git-flow и gh.



```
mvkarimi@mvkarimi-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
mvkarimi@mvkarimi-VirtualBox:~$ git  
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>]  
        <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 суще  
ствующего  
  
работа с текущими изменениями (смотрите также: git help everyday)  
  add        Добавление содержимого файла в индекс  
  mv         Перемещение или переименование файла, каталога или символ  
ой ссылки  
  restore    Restore working tree files  
  rm         Удаление файлов из рабочего каталога и индекса  
  sparse-checkout Initialize and modify the sparse-checkout  
  
просмотр истории и текущего состояния (смотрите также: git help revisions)
```

Figure 2.1: Загрузка пакетов

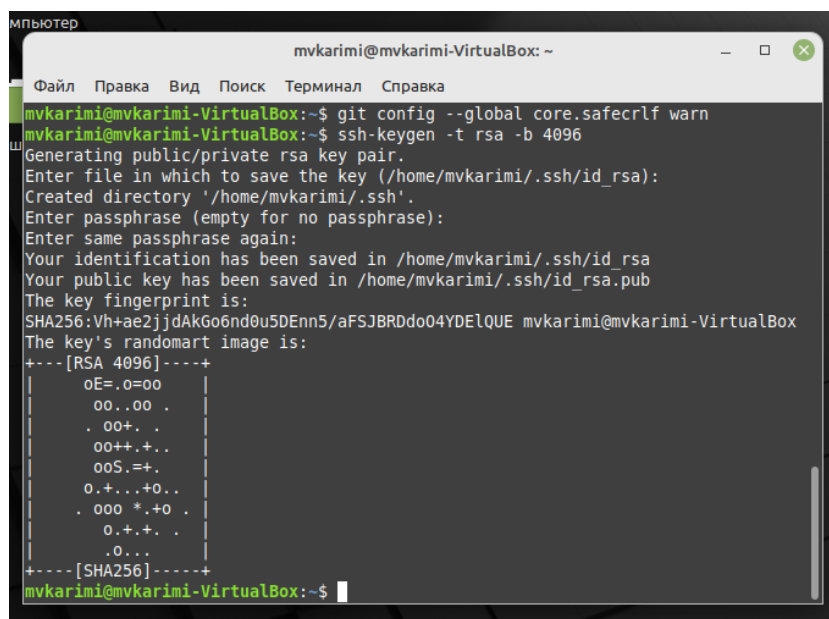
Зададим имя и email владельца репозитория, кодировку и прочие параметры.



```
mvkarimi@mvkarimi-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
анной 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 репозитием ил  
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.  
mvkarimi@mvkarimi-VirtualBox:~$  
mvkarimi@mvkarimi-VirtualBox:~$  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global user.name "validkarimi"  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global user.email "1132215500@pfur.  
ru"  
mvkarimi@mvkarimi-VirtualBox:~$  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global core.quotepath false  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global init.defaultBranch master  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global core.autocrlf input  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global core.safecrlf warn  
mvkarimi@mvkarimi-VirtualBox:~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи



```
mvkarimi@mvkarimi-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global core.safecrlf warn  
mvkarimi@mvkarimi-VirtualBox:~$ ssh-keygen -t rsa -b 4096  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/mvkarimi/.ssh/id_rsa):  
Created directory '/home/mvkarimi/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/mvkarimi/.ssh/id_rsa  
Your public key has been saved in /home/mvkarimi/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:Vh+ae2jjdAkGo6nd0u5DEnn5/aFSJBRDdo04YDElQUE mvkarimi@mvkarimi-VirtualBox  
The key's randomart image is:  
+---[RSA 4096]---+  
| oE=.o=oo      |  
| oo..oo .      |  
| . oo+. .      |  
| oo+++. .      |  
| ooS.=+.      |  
| o.+...+o. .   |  
| . ooo *.+o .  |  
| o.++. .      |  
| .o...         |  
+---[SHA256]-----+  
mvkarimi@mvkarimi-VirtualBox:~$
```

Figure 2.3: rsa-4096

```
mvkarimi@mvkarimi-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
|-----[SHA256]-----+  
mvkarimi@mvkarimi-VirtualBox:~$ ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/mvkarimi/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/mvkarimi/.ssh/id_ed25519  
Your public key has been saved in /home/mvkarimi/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:ww39QZV1pvu/3oqfH1/vFrwkCwqRys6D06miXmyzKNY mvkarimi@mvkarimi-VirtualBox  
The key's randomart image is:  
+---[ED25519 256]---+  
|  
| ..o+ |  
| . . .o. |  
| . o . . |  
| . = o . . |  
| . S + o + |  
| . o + o = |  
| . = o o = |  
| ..+Eoo * . o0 |  
| ++....o . . =B0 |  
+---[SHA256]-----+  
mvkarimi@mvkarimi-VirtualBox:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
mvkarimi@mvkarimi-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
"validkarimi <1132215500@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/mvkarimi/.gnupg/trustdb.gpg: создана таблица доверия  
gpg: ключ 8079D14AD0AE9B2E помечен как абсолютно доверенный  
gpg: создан каталог '/home/mvkarimi/.gnupg/openpgp-revocs.d'  
gpg: сертификат отзыва записан в '/home/mvkarimi/.gnupg/openpgp-revocs.d/FFF2CC9  
C0AE481D5FE20CE498079D14AD0AE9B2E.rev'.  
открытый и секретный ключи созданы и подписаны.  
  
pub   rsa4096 2022-04-30 [SC]  
      FFF2CC9C0AE481D5FE20CE498079D14AD0AE9B2E  
uid           validkarimi <1132215500@pfur.ru>  
sub   rsa4096 2022-04-30 [E]  
mvkarimi@mvkarimi-VirtualBox:~$
```

Figure 2.5: GPG ключ


Добавляем GPG ключ в аккаунт

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

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: 1132215500@pfur.ru
Key ID: 8079D14AD0AE9B2E
Subkeys: 3DC16E0A85F04AC0
Added on 30 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

```
mvkarimi@mvkarimi-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
XzCkqjxrfl7G551wTEzL03fKtGDIMS+6Tiggm973GZqLB5qccTF1UeKkrUdsh/Vz  
QY6jBbgvM8y0uRZH7IqFf6I4U8dkPgrjV0fERgoXcjm7co60Pvhg040nqr8acFVI  
x3ezslbH01amGc8S531CwsGVMYE9cHVRf5frm5CDy+eB06Agy7jMzwgPyGKH+QAR  
AQABiQI2BBgBCgAgFiEE//LMnArkgdX+IM5JgHnRStCummy4FAMJtGPUCGwwACgkQ  
gHnRStCummy5a+w//TvB7ARV1LG40gZq64I/MNN3XMhhSTjGbfu+onXNU3t7e3cVR  
cZiRiya/McoHU0kwZ794nTlavkC0p+h2j/Lw2d2euMrWKRAT0hdc+A9nA4Py72kI  
kMJdyrWVg67IG5BXx2Pes+gyf5kTuNpr58xowRqgvLG2niU8pmH2T1mHDWqsoo83  
bW4oGdJEETwRH1h70XLg7iGDecXrRgTncWg9Jm20HFNeS01R2FQ3xDqHFczEZhZC  
cBFCpnmS4q2sFdWa4aufsgxC1Ck0vrMofclw7l0LA9d+tTCP0wxa1FYS0Y0qu7T9  
9mAXKd1yTULHe3wvk0eVSuR2Q0nz0xhruzj8TVankeHSRLed68GLzGDp5Q5a6hb2  
XJ8IT3LzS+Z+N/L0aGgI47KwtmUfwNfLcVN9Ll3irlz/QbMQXbnuWfSD0bimcFcR1  
LSx863tLbZu0i17yjp7/yNJ7GHph6FzRoC26G3JNvgj2aHfXN/7X7RdI6yjoV3  
aXxIvcCIDtf+k1hK6wHCh9VBs+VJt12fZCYGDYSKSeLoVzpqH/tr0iZraGdeNHI  
qWHT6VbWnBnUMX8Durf3Fo+tkda1LM9L7JLnBiX9EmEzWfRsQBx6bwbvlyacdAV  
7JxJNR4ZepBXoJ0eX06NERuPLCIwPPG7GkIVvelqdTZ0Pu04sps0Vr0m298=  
=h48x  
-----END PGP PUBLIC KEY BLOCK-----  
mvkarimi@mvkarimi-VirtualBox:~$  
mvkarimi@mvkarimi-VirtualBox:~$  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global user.signingkey 8079D14AD0AE  
9B2E  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global commit.gpgsign true  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global gpg.program $(which gpg2)  
mvkarimi@mvkarimi-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

Настройка gh


```
mvkarimi@mvkarimi-VirtualBox: ~  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global user.signingkey 8079D14AD0AE9B2E  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global commit.gpgsign true  
mvkarimi@mvkarimi-VirtualBox:~$ git config --global gpg.program $(which gpg2)  
mvkarimi@mvkarimi-VirtualBox:~$ 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/mvkarimi/.ssh/id_rsa.pub  
? How would you like to authenticate GitHub CLI? Login with a web browser  
! First copy your one-time code: SF20-BE52  
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/mvkarimi/.ssh/id_rsa.pub  
✓ Logged in as validkarimi  
mvkarimi@mvkarimi-VirtualBox:~$  
###!!! [Parent][RunMessage] Error: Channel closing: too late to send/recvd, messages will be lost  
mvkarimi@mvkarimi-VirtualBox:~$
```

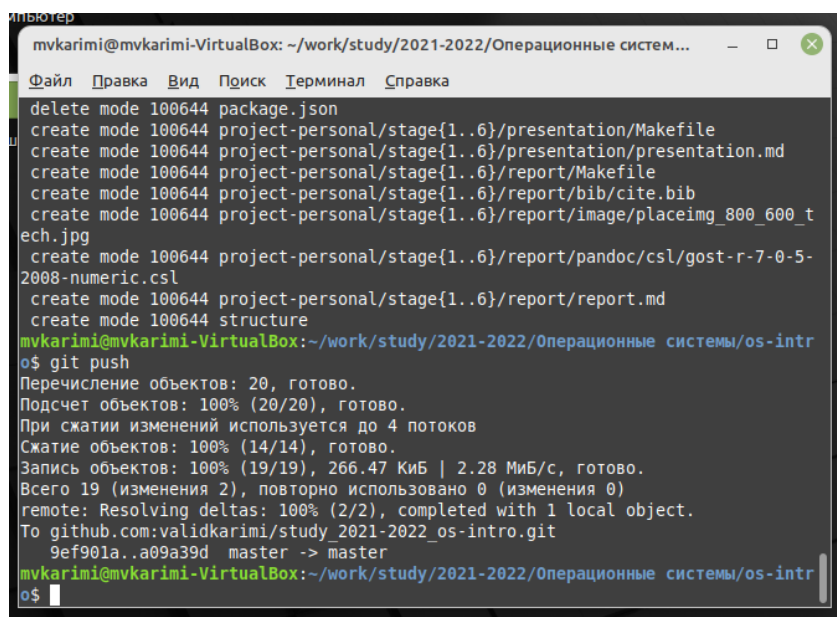
Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
mvkarimi@mvkarimi-VirtualBox: ~/work/study/2021-2022/Операционные системы  
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0  
Получение объектов: 100% (20/20), 12.49 КиБ | 6.24 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/mvkarimi/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/mvkarimi/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 «3eae7b7586f8a9aded2b506cd1018e625b228b93»  
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»  
mvkarimi@mvkarimi-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений



```
mvkarimi@mvkarimi-VirtualBox: ~/work/study/2021-2022/Операционные систем...
Файл  Правка  Вид  Поиск  Терминал  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_t
ech.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
mvkarimi@mvkarimi-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intr
o$ git push
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 | 2.28 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:validkarimi/study_2021-2022_os-intro.git
   9ef901a..a09a39d  master -> master
mvkarimi@mvkarimi-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intr
o$
```

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