

Отчёт по лабораторной работе №2

Управление версиями

Филиппова Екатерина 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

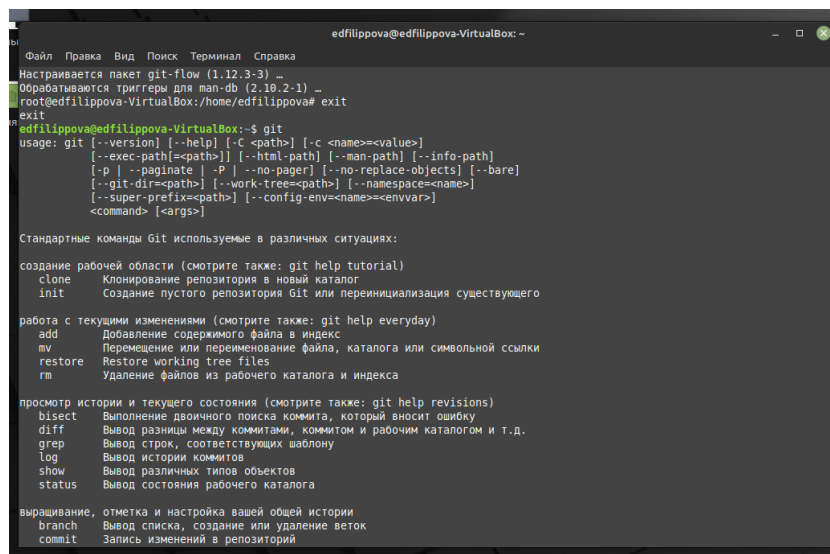
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	6
2.3	rsa-4096	6
2.4	ed25519	7
2.5	GPG ключ	7
2.6	GPG ключ	8
2.7	Параметры репозитория	8
2.8	Связь репозитория с аккаунтом	9
2.9	Загрузка шаблона	9
2.10	Первый коммит	10

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.



```
edfilippova@edfilippova-VirtualBox: ~  
Настраивается пакет git-flow (1.12.3-3) _  
Обрабатываются триггеры для man-db (2.10.2-1) _  
root@edfilippova-VirtualBox:/home/edfilippova# exit  
exit  
edfilippova@edfilippova-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>]  
        [--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
```

Figure 2.1: Загрузка пакетов

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

```
edfilippova@edfilippova-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
git 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' 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.
edfilippova@edfilippova-VirtualBox:~$ git config --global user.name "edfilippova"
edfilippova@edfilippova-VirtualBox:~$ git config --global user.email "1032217606@pfur.ru"
edfilippova@edfilippova-VirtualBox:~$ git config --global core.quotepath false
edfilippova@edfilippova-VirtualBox:~$ git config --global init.defaultBranch master
edfilippova@edfilippova-VirtualBox:~$ git config --global core.autocrlf input
edfilippova@edfilippova-VirtualBox:~$ git config --global core.safecrlf warn
edfilippova@edfilippova-VirtualBox:~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
edfilippova@edfilippova-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
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.
edfilippova@edfilippova-VirtualBox:~$ git config --global user.name "edfilippova"
edfilippova@edfilippova-VirtualBox:~$ git config --global user.email "1032217606@pfur.ru"
edfilippova@edfilippova-VirtualBox:~$ git config --global core.quotepath false
edfilippova@edfilippova-VirtualBox:~$ git config --global init.defaultBranch master
edfilippova@edfilippova-VirtualBox:~$ git config --global core.autocrlf input
edfilippova@edfilippova-VirtualBox:~$ git config --global core.safecrlf warn
edfilippova@edfilippova-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/edfilippova/.ssh/id_rsa):
Created directory '/home/edfilippova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/edfilippova/.ssh/id_rsa
Your public key has been saved in /home/edfilippova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:J2YEWpTehmkIatK/IowyHQ+7lCTKc62jYr1whxjw edfilippova@edfilippova-VirtualBox
The key's randomart image is:
+---[RSA 4096]---+
|.+.
|..O..
|...O.O..
|..+ + O..
|..+ = E + S..
|ooX.* +ooo
|++OX + O
|==+ +
|..O.O
+---[SHA256]-----+
edfilippova@edfilippova-VirtualBox:~$
```

Figure 2.3: rsa-4096

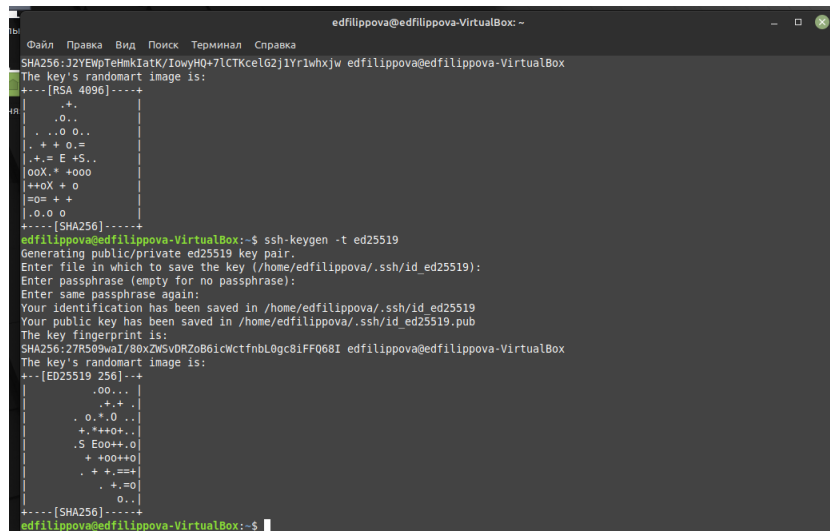


Figure 2.4: ed25519

Создаем GPG ключ

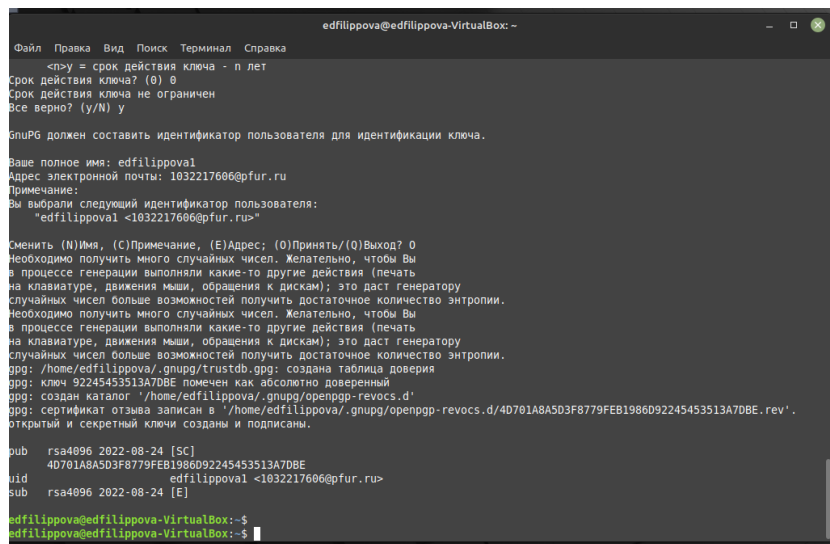


Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

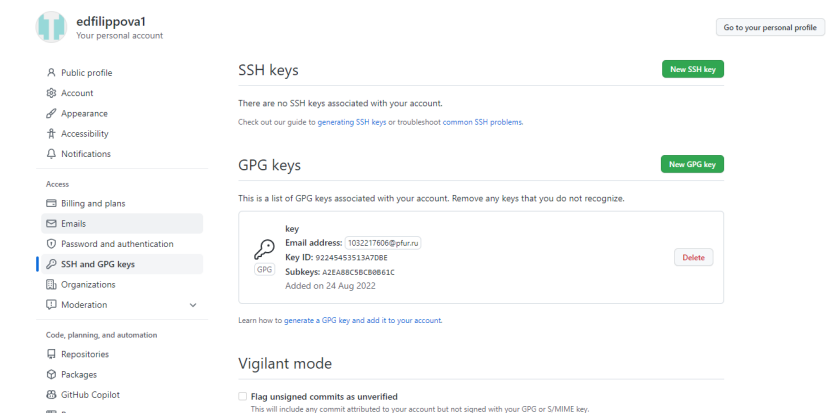


Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

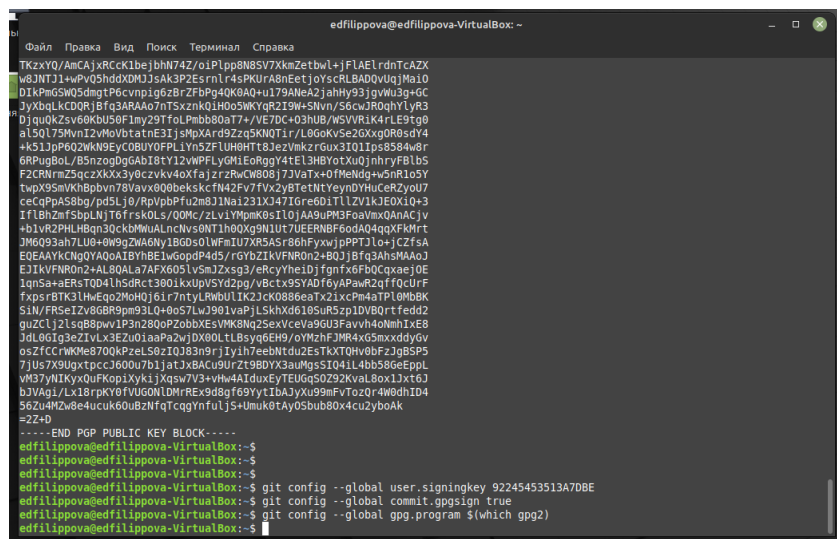


Figure 2.7: Параметры репозитория

Настройка gh


```
edfilippova@edfilippova-VirtualBox: ~  
Может быть обновлено 243 пакета. Запустите «apt list --upgradable» для их показа.  
Чтение списков пакетов... Готово  
Построение дерева зависимостей... Готово  
Чтение информации о состоянии... Готово  
Следующие НОВЫЕ пакеты будут установлены:  
  gh  
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 243 пакетов не обновлено.  
Необходимо скачать 8 087 КБ архивов.  
После данной операции объём занятого дискового пространства возрастёт на 34,8 МБ.  
Полн:1 https://cli.github.com/packages/stable/main amd64 gh amd64 2.14.6 [8 087 КБ]  
Получено 8 087 КБ за 1с (9 151 КБ/с)  
Выбор ранее не выбранного пакета gh.  
(Чтение базы данных ... на данный момент установлен 284941 файл и каталог.)  
Подготовка к распаковке ./archives/gh_2.14.6_amd64.deb ...  
Распаковывается gh (2.14.6) ...  
Настраивается пакет gh (2.14.6) ...  
Обрабатываются триггеры для man-db (2.10.2-1) ...  
edfilippova@edfilippova-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/edfilippova/.ssh/id_rsa.pub  
? Title for your SSH key: yamadharmalab  
? How would you like to authenticate GitHub CLI? Login with a web browser  
  
! First copy your one-time code: 8E6F-C15B  
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/edfilippova/.ssh/id_rsa.pub  
  Logged in as edfilippova  
edfilippova@edfilippova-VirtualBox:~$ 2022-08-24T10:21:44Z ERROR viaduct::backends::ffi Missing HTTP status  
2022-08-24T10:21:44Z ERROR viaduct::backends::ffi Missing HTTP status  
edfilippova@edfilippova-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
edfilippova@edfilippova-VirtualBox: ~/work/study/2021-2022/Операционные системы  
edfilippova@edfilippova-VirtualBox:~/work/study/2021-2022/Операционные системы$ git clone --recursive git@github.com:edfilippova  
val/study_2021-2022_os-intro.git os-intro  
Клонирование в «os-intro»..  
The authenticity of host 'github.com (140.82.121.3)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvV6TujHbpZisF/zLDA0zPMSvHdkr4Uvc0Qy.  
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 Киб | 12.49 Миб/с, готово.  
Определение изменений: 100% (2/2), готово.  
Подмодуль «template/presentation» (https://github.com/yamadharmalab/academic-presentation-markdown-template.git) зарегистрирован  
по пути «template/presentation»  
Подмодуль «template/report» (https://github.com/yamadharmalab/academic-laboratory-report-template.git) зарегистрирован по пути «t  
emplate/report»  
Клонирование в «/home/edfilippova/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  
Получение объектов: 100% (42/42), 31.19 Киб | 638.00 Киб/с, готово.  
Определение изменений: 100% (9/9), готово.  
Клонирование в «/home/edfilippova/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 Киб | 3.48 Миб/с, готово.  
Определение изменений: 100% (31/31), готово.  
Submodule path 'template/presentation': checked out '3eae7b7586f8a9aded2b506cd1018e625b228b93'  
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'  
edfilippova@edfilippova-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
edfilippova@edfilippova-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro
М/Оs-Intro
edfilippova@edfilippova-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ rm package.json
edfilippova@edfilippova-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ make COURSE=os-intro
edfilippova@edfilippova-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git add .
edfilippova@edfilippova-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 90e1c42] feat(main): make course structure
16 files changed, 1500 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
edfilippova@edfilippova-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 4 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (19/19), 266.46 КИБ | 2.13 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:edfilippova/study-2021-2022-os-intro.git
dc8e428..90e1c42 master -> master
edfilippova@edfilippova-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