

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

№ 2

«Управление версиями»

Коняева Марина Александровна

Содержание

Цель работы	3
Теоретическое введение	4
Выполнение лабораторной работы	5
Выводы	14
Контрольные вопросы	15

Цель работы

Изучить идеологию и применение средств контроля версий, освоить умения по работе с git.

Теоретическое введение

Markdown — язык текстовой разметки, созданный писателем и блогером Джоном Грубером. Он предназначен для создания красиво оформленных текстов в обычных файлах формата TXT. Вам не нужны громоздкие процессоры вроде Word или Pages, чтобы создавать документы с жирным или курсивным начертанием, цитатами, ссылками и даже таблицами. Достаточно запомнить простые правила Markdown, и можно писать хоть в «Блокноте». Хотя специализированные Markdown-редакторы, конечно, намного удобнее.

Выполнение лабораторной работы

1. Заведем учетную запись на сайте github.com, свой аккаунт я создала ранее, введя необходимые данные
2. Скачивание `git-flow` и выдача прав на выполнение

```
[makonyaeva@makonyaeva tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[makonyaeva@makonyaeva tmp]$ ls
gitflow-installer.sh
gitflow-installer.sh.1
systemd-private-2bd953f9d1e34965a1426d30a17a2629-chronyd.service-5g573c
systemd-private-2bd953f9d1e34965a1426d30a17a2629-color.service-n655ke
systemd-private-2bd953f9d1e34965a1426d30a17a2629-dbus-broker.service-A030k0
systemd-private-2bd953f9d1e34965a1426d30a17a2629-fwupd.service-rdxXc5
systemd-private-2bd953f9d1e34965a1426d30a17a2629-geoclue.service-U4cAog
systemd-private-2bd953f9d1e34965a1426d30a17a2629-low-memory-monitor.service-Azn845
systemd-private-2bd953f9d1e34965a1426d30a17a2629-ModemManager.service-e
systemd-private-2bd953f9d1e34965a1426d30a17a2629-power-profiles-daemon.
systemd-private-2bd953f9d1e34965a1426d30a17a2629-rtkit-daemon.service-B
systemd-private-2bd953f9d1e34965a1426d30a17a2629-switcheroo-control.ser
systemd-private-2bd953f9d1e34965a1426d30a17a2629-systemd-logind.service
systemd-private-2bd953f9d1e34965a1426d30a17a2629-systemd-oomd.service-G
systemd-private-2bd953f9d1e34965a1426d30a17a2629-systemd-resolved.servi
systemd-private-2bd953f9d1e34965a1426d30a17a2629-upower.service-b198qt
```

Изображение 2.1 Скачивание `git-flow`

```
[makonyaeva@makonyaeva tmp]$ chmod +x gitflow-installer.sh
[makonyaeva@makonyaeva tmp]$
```

Изображение 2.2 выдача прав на выполнение

3. Установка `git-flow`

```
[makonyaeva@makonyaeva tmp]$ sudo ./gitflow-installer.sh install stable
```

Изображение 3.1 Установка `git-flow`

```
[sudo] пароль для makonyaeva:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование в «gitflow»...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Получение объектов: 100% (4270/4270), 1.74 МиБ | 3.06 МиБ/с, готово.
Определение изменений: 100% (2533/2533), готово.
```

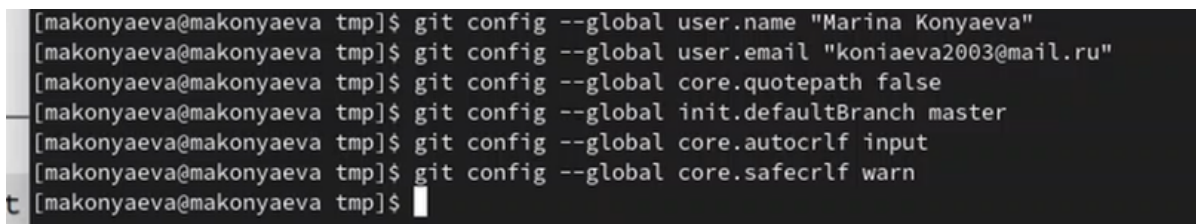
Изображение 3.2 Установка `git-flow`

4. Установка пакета gh



Изображение 4.1 Установка пакета gh

5. Базовая настройка git, включающая имя, email, кодировку, имя начальной ветки, autocrlf и safecrlf



Изображение 5.1 Базовая настройка git

6. Создание ssh ключей с двумя алгоритмами шифрования: RSA и ed25529

```

[makonyaeva@makonyaeva tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/makonyaeva/.ssh/id_rsa):
Created directory '/home/makonyaeva/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/makonyaeva/.ssh/id_rsa
Your public key has been saved in /home/makonyaeva/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:r0sPXjSB7mF0zakMn4+oHIS2var30qFDshciHxSp/Sw makonyaeva@makonyaeva
The key's randomart image is:
+---[RSA 4096]-----+
|      .               |
|     o . . o .       |
|    o . + o +        |
|   . o. o * +        |
|  .oo. + S           |
| +oE+oo = +         |
| +=.=o = o .        |
| + =. = +           |
| .o==.+..           |
+---[SHA256]-----+
[makonyaeva@makonyaeva tmp]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/makonyaeva/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/makonyaeva/.ssh/id_ed25519
Your public key has been saved in /home/makonyaeva/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:pZFlyCPK97iv2M+uCOVmrh75+VG0Kk/QV1hqJ3Y642M makonyaeva@makonyaeva
The key's randomart image is:
+---[ED25519 256]---+
|      . . . . .o      |
|     ..o+=           |
|    +.0.+            |
|   . B 0             |
|  . S * o            |
|   = o B             |
|  + = E .            |
|   .0 B =            |
|   .o+0.==          |
+---[SHA256]-----+
[makonyaeva@makonyaeva tmp]$

```

Изображение 6.1 Создание ssh ключей

```

[makonyaeva@makonyaeva OperatingSystems]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCs6+wFdX3b1jaBoHD336fGgPHd37fxdnuRg/YUPB2bGIQ9v/ftU0uj/z8XEbQutwq88+hLGHmb1Ho814KhMQ56YRtWe9q1+PH0CIaIN6/mNjH7ZVvOgwEnyQYImY4NhOH
+gFSWngI6XEJo5vgbhCKYD4Wmo2poVhIPiFfJbBOCelg10axVf7wPM8B14IDgZ12YTUIyd26LanhW3DYzZ2bWda6YXjtZtSJoTjFnNevf18K6Zm3YlfrZw7p8G6XS1ub3kyhnJgy1ShpLDVRW9U/Tpb7r8j1XJHL31gekq8P
ywGx+qM3iTHEE3P/qH4v80beCt2cqrzXhGI5W5Uc8AV0tEqc6R3VKLxKQZrum/EAF6az1PX0G3nbMVoJ0CbLw9TrfYD6VXkLMKBH+S+k5oawy4SH7gM3pIzppKQyc5TexwXXwnT0xulejxUodc7/TbZ3vO3LD8ECdW2
IZ/vQsuQ== makonyaeva@makonyaeva
[makonyaeva@makonyaeva OperatingSystems]$

```

Изображение 6.2 Создание ssh ключей

7. Создание pgr ключа

```

[makonyaeva@makonyaeva tmp]$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/makonyaeva/.gnupg'
gpg: создан щит с ключами '/home/makonyaeva/.gnupg/pubring.kbx'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
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: Marina
Адрес электронной почты: koniaeva2003@mail.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Marina <koniaeva2003@mail.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0

```

Изображение 7.1 Создание pgp ключа

8. Добавление ключа PGP на Github


```

[makonyaeva@makonyaeva tmp]$ gpg --list-secret-keys --keyid-format LONG
gpg: invalid option "--keyid-format"
[makonyaeva@makonyaeva tmp]$
[makonyaeva@makonyaeva tmp]$ 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
/home/makonyaeva/.gnupg/pubring.kbx
-----
sec   rsa4096/5A6AC3D30160D9A6 2022-04-21 [SC]
      D016E419899706D4829E94CC5A6AC3D30160D9A6
uid           [ абсолютно ] Marina <koniaeva2003@mail.ru>
ssb   rsa4096/AC835A6705FFDA68 2022-04-21 [E]

[makonyaeva@makonyaeva tmp]$ gpg --armor --export ^C
[makonyaeva@makonyaeva tmp]$ gpg --armor --export 5A6AC3D30160D9A6 | xclip -sel clip
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y] y

* Waiting in queue...
The following packages have to be installed:
xclip-0.13-15.git11cba61.fc35.x86_64  Command line clipboard grabber
Proceed with changes? [N/y] y

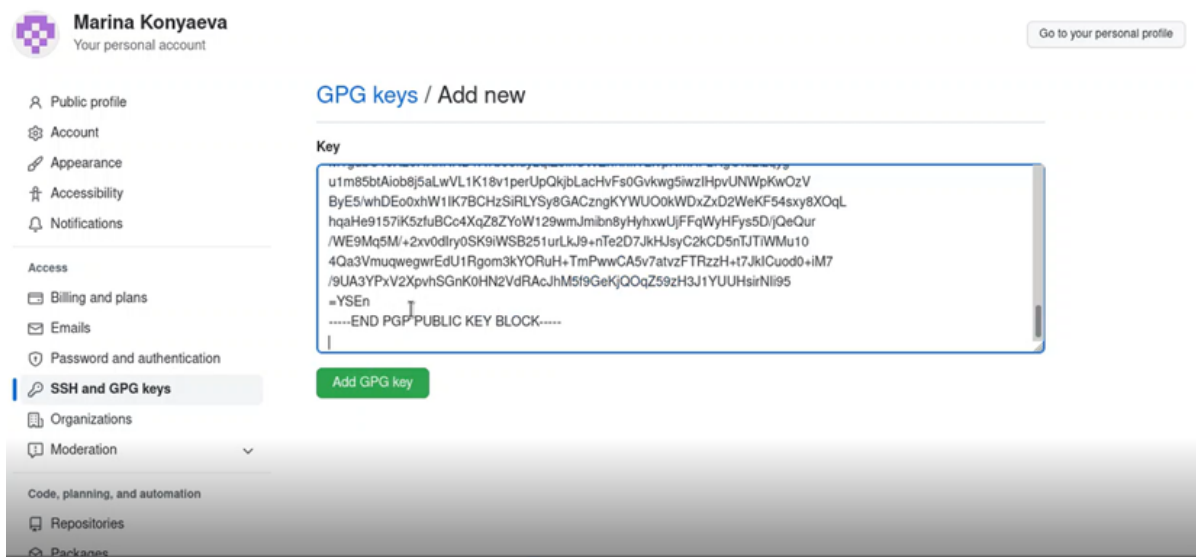
* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...

[makonyaeva@makonyaeva tmp]$ gpg --armor --export 5A6AC3D30160D9A6 | xclip -sel clip
[makonyaeva@makonyaeva tmp]$

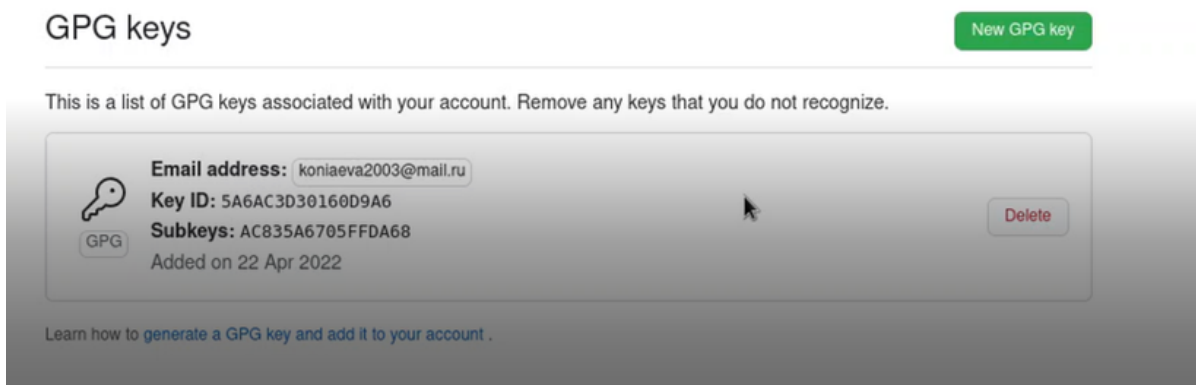
```

Изображение 8.1 Добавление ключа PGP на Github

9. Просмотр отпечатка и копирования в буфер ключа pgp



Изображение 9.1 Просмотр отпечатка и копирования



Изображение 9.2 Просмотр отпечатка и копирования

10. Создание и клонирования репозитория

```
[makonyaeva@makonyaeva OperatingSystems]$ gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
```

Изображение 10.1 Создание и клонирования репозитория

```

[makonyaeva@makonyaeva Operatingsystems]$ git clone --recursive https://github.com/Marrina6735/study_2021-2022_os-intro.git
Клонирование в «study_2021-2022_os-intro»...
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), 12.50 КиБ | 180.00 КиБ/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/makonyaeva/work/study/2021-2022/Operatingsystems/study_2021-2022_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 | 2.60 МиБ/с, готово.
Определение изменений: 100% (9/9), готово.
Клонирование в «/home/makonyaeva/work/study/2021-2022/Operatingsystems/study_2021-2022_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 | 1.87 МиБ/с, готово.
Определение изменений: 100% (31/31), готово.
Подмодуль по пути «template/presentation»: забрано состояние «3eae6b7586f8a9aded2b506cd1018e625b228b93»

```

Изображение 10.2 Создание и клонирования репозитория

11. Настройки каталога курса, а именно, удаление json файла и создание структуры

```

[makonyaeva@makonyaeva course-directory-student-template]$ rm package.json
[makonyaeva@makonyaeva course-directory-student-template]$ make COURSE=os-intro

```

Изображение 11.1 Настройки каталога

12. Создание commit

```

[makonyaeva@makonyaeva study_2021-2022_os-intro]$ git add .
[makonyaeva@makonyaeva study_2021-2022_os-intro]$ git commit -am 'feat(main): make course structure'
[master 356ccd3] feat(main): make course structure
148 files changed, 16590 insertions(+)
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
create mode 100644 labs/lab03/presentation/presentation.md
create mode 100644 labs/lab03/report/Makefile
create mode 100644 labs/lab03/report/bib/cite.bib
create mode 100644 labs/lab03/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab03/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab03/report/report.md
create mode 100644 labs/lab04/presentation/Makefile
create mode 100644 labs/lab04/presentation/presentation.md
create mode 100644 labs/lab04/report/Makefile
create mode 100644 labs/lab04/report/bib/cite.bib
create mode 100644 labs/lab04/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab04/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab04/report/report.md
create mode 100644 labs/lab05/presentation/Makefile
create mode 100644 labs/lab05/presentation/presentation.md
create mode 100644 labs/lab05/report/Makefile
create mode 100644 labs/lab05/report/bib/cite.bib
create mode 100644 labs/lab05/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab05/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab05/report/report.md
create mode 100644 labs/lab06/presentation/Makefile
create mode 100644 labs/lab06/presentation/presentation.md

```

Изображение 12.1 Создание commit

13. Push на сервер

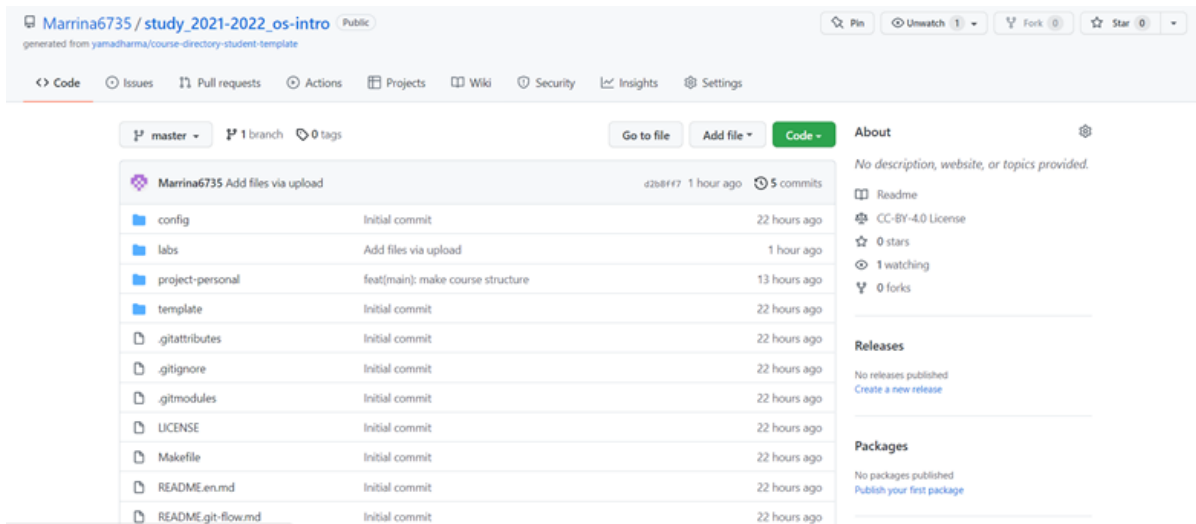
```

[makonyaeva@makonyaeva study_2021-2022_os-intro]$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (19/19), 266.53 КиБ | 2.18 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 https://github.com/Marrina6735/study_2021-2022_os-intro.git
   43e089f..356ccd3  master -> master
0:00:17
[makonyaeva@makonyaeva study_2021-2022_os-intro]$

```

Изображение 13.1 Push

14. Проверим, что репозиторий появился на Github



Изображение 14.1 репозиторий на Github

Выводы

В ходе данной лабораторной работы я получила знания о системах контроля версий, изучила идеологию и применение данной технологии, также получила практические навыки по настройке системы и по созданию репозитория и обработки файлов.

Контрольные вопросы

1. Система контроля версий (VCS) — это место хранения кода. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы.
2. Репозиторий - хранилище версий - в нем хранятся все документы вместе с историей их изменения и другой служебной информацией Commit («[трудо-вой] вклад», не переводится) — процесс создания новой версии Рабочая копия (working copy) — текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней). Версия (revision), или ревизия, — состояние всех файлов на определенный момент времени, сохраненное в репозитории, с дополнительной информацией.
3. Централизованные системы — это системы, которые используют архитектуру клиент / сервер, где один или несколько клиентских узлов напрямую подключены к центральному серверу. (Пример — Wikipedia.) В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов. (Пример — Bitcoin).
4. Изначально разработчик работает с веткой master. При реализации отдельных частей проекта может создать ветки для них. При завершении изменений

разработчик коммитит и пушит изменения на сервер. Если разработка на сторонней ветке завершена, то её можно смирджить (merge), например с основной веткой master.

5. Каждый разработчик работает в своей ветке над отдельной частью проекта. Если на ветке работает несколько разработчиков они как правило достать изменения, сделанные другим и работать уже с ними (git pull). После завершения или заканчивая какой-то логический кусок они делают коммит и пушат на сервер. Если работа закончена необходимо смирджить ветки, например с главной веткой(master).
6. У Git есть две основные задачи: хранить информацию обо всех изменениях в коде, начиная с самой первой строчки, и обеспечить удобства командной работы над кодом.
7. – создание основного дерева репозитория: git init – получение обновлений (изменений) текущего дерева из центрального репозитория: git pull – отправка всех произведённых изменений локального дерева в центральный репозиторий: git push – просмотр списка изменённых файлов в текущей директории: git status – просмотр текущих изменения: git diff – сохранение текущих изменений: – добавить все изменённые и/или созданные файлы и/или каталоги: git add . – добавить конкретные изменённые и/или созданные файлы и/или каталоги: git add имена_файлов – удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): git rm имена_файлов – сохранение добавленных изменений: – сохранить все добавленные изменения и все изменённые файлы: git commit -am ‘Описание коммита’ – сохранить добавленные изменения с внесением комментария через встроенный редактор: git commit – создание новой ветки, базирующейся на текущей: git checkout -b имя_ветки – переключение на некоторую ветку: git checkout имя_ветки (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) – отправка измене-

ний конкретной ветки в центральный репозиторий: `git push origin имя_ветки`
– слияние ветки с текущим деревом: `git merge --no-ff имя_ветки` – удаление ветки: – удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки` – принудительное удаление локальной ветки: `git branch -D имя_ветки` – удаление ветки с центрального репозитория: `git push origin :имя_ветки`.

8. Работа с таким удалённым репозиторием подразумевает выполнение стандартных операций отправки и получения, как и с любым другим удалённым репозиторием. Просмотр удалённых репозиториев. Для того, чтобы просмотреть список настроенных удалённых репозиториев, вы можете запустить команду `git remote add`. Когда вы клонируете репозиторий, команда `clone` автоматически добавляет этот удалённый репозиторий под именем «origin». Таким образом, `git fetch origin` извлекает все наработки, отправленные на этот сервер после того, как вы его клонировали (или получили изменения с помощью `fetch`).
9. ‘Git branch’ – это команда для управления ветками в репозитории Git. Ветка – это просто «скользящий» указатель на один из коммитов. Когда мы создаём новые коммиты, указатель ветки автоматически сдвигается вперёд, к вновь созданному коммиту. Ветки используются для разработки одной части функционала изолированно от других. Каждая ветка представляет собой отдельную копию кода проекта. Ветки позволяют одновременно работать над разными версиями проекта. Ветвление («ветка», `branch`) — один из параллельных участков истории в одном хранилище, исходящих из одной версии (точки ветвления). Ветки нужны для того, чтобы программисты могли вести совместную работу над проектом и не мешать друг другу при этом.
10. Игнорировать файлы коммита можно с помощью файла `.gitignore`. Туда обычно помещаются файлы, которые занимают много места или не особо нужны для проекта. Например, картинки, папки с кешем, веса для нейросетей и другие файлы.