

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютера

Студент: Арфонос Дмитрий

Группа: НММбд-04-23

МОСКВА

2023 г.

Цель работы

Ознакомиться с принципами работы средств контроля версий. Настроить git для начала работы. Используя git, создать рабочее пространство и репозиторий курса, после чего загрузить файлы на github.

Настройка github

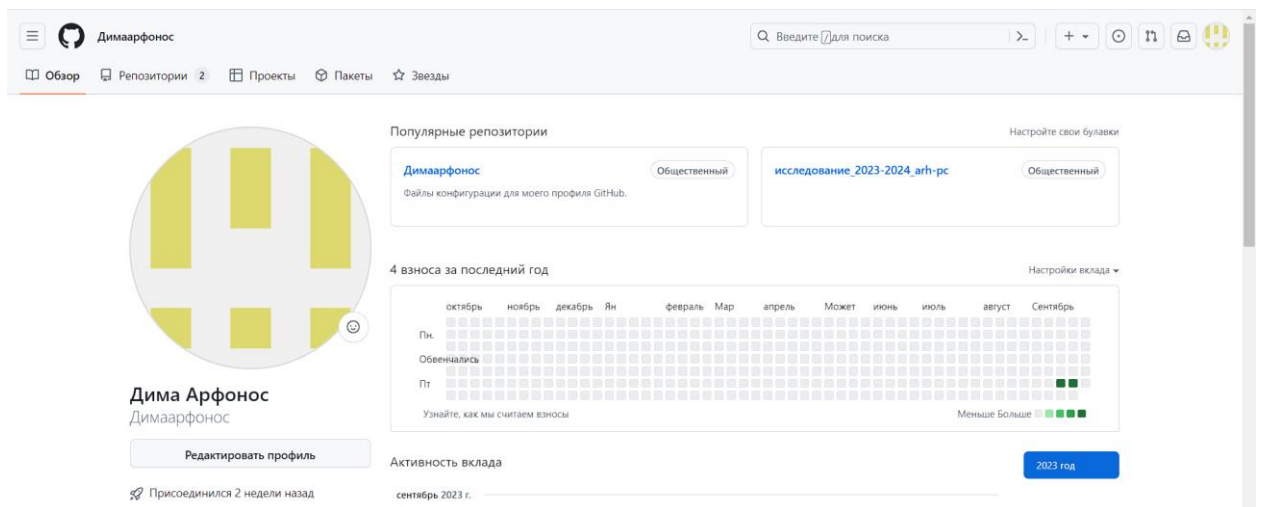


Рис. 1.1. Создали учётную запись на сайте github

```
[darfonos@fedora ~]$ git config --global user.name
[darfonos@fedora ~]$ git config --global user.name "Darfonos"
[darfonos@fedora ~]$ git config --global user.email "arfonosdima13@gmail.com">
```

Рис. 2.1. Указали имя и e-mail владельца репозитория

```
[darfonos@fedora ~]$ git config --global core.quotePath false
[darfonos@fedora ~]$ git config --global init.defaultBranch master
[darfonos@fedora ~]$ git config --global core.autocrlf input
[darfonos@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 2.1.

- Настроили utf-8 в выводе сообщений git
- Задали имя “мастер” для начальной ветки.
- Настроили параметры autocrlf и safecrlf

Создание SSH КЛЮЧА

```
[darfonos@fedora ~]$ ssh-keygen -C "Darfonos <arfonosdimal3@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/darfonos/.ssh/id_rsa):
Created directory '/home/darfonos/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/darfonos/.ssh/id_rsa
Your public key has been saved in /home/darfonos/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+LJA3shtreYETkBRsZGftrzHiQV27VMRdfnLKX/T+ps Darfonos <arfonosdimal3@gmail.com>
The key's randomart image is:
+---[RSA 3072]-----+
|  oo+o      .o. o |
| . .o      . o  |
| . . . . . . . |
| . * . . . . . |
| ++.+S. . . o |
| * =oo. o . + |
| * *=o. . . o . |
| +++++      o+ |
| o+.      .E= |
+---[SHA256]-----+
```

Рис. 3.1. Сгенерировали пару ключей (приватный и открытый)

```
[darfonos@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[darfonos@fedora ~]$ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCF/Hz4k/FC2JMzcQHams9r6A++32zHaQajnYGwnXLC2GDz7DPRmuEghZsmT8zP1gUoLjQPTwoUeq$
YuVEPdBRmb+iOV+075vDsZ50gII7XuVLBF3RQqpbTn6y9GAKXA/VUy9pPJPNgjAPtNmgyD78Cl117DMkSXsuvhg715zWjP0AiDbu2qR0k289P0hTb4tXqjSyt49+Aa/zw2KdV+Ox8CV
JX/VPHLG3Wyans0gWqZPSImsXt5UKdda8dIgL93+FHu6Q
bash: ssh-rsa: команда не найдена...
```

Рис. 3.2. Скопировали из локальной консоли ключ в буфер обмена

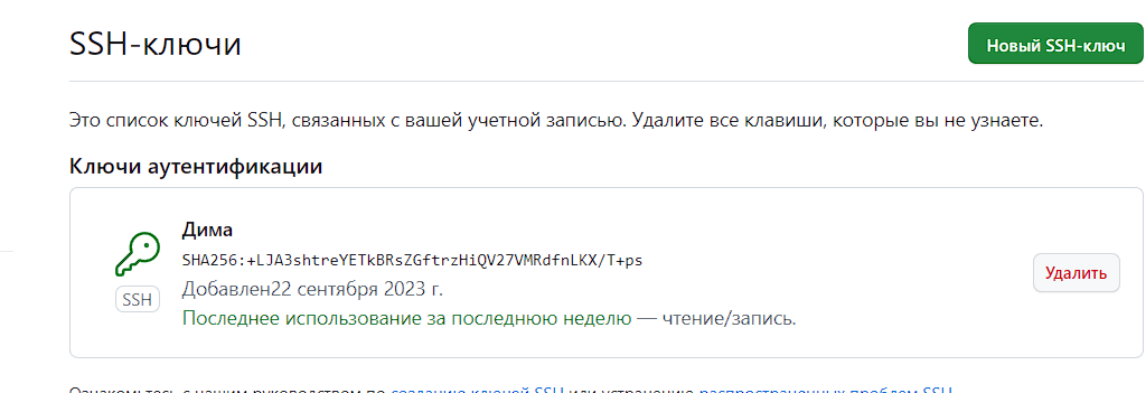


Рис. 3.3. Создали SSH-ключ на сайте github и дали ей “Дима”.

```
[darfonos@fedora ~]$
[darfonos@fedora ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис. 4.1. Создали каталог для предмета «Архитектура компьютера»

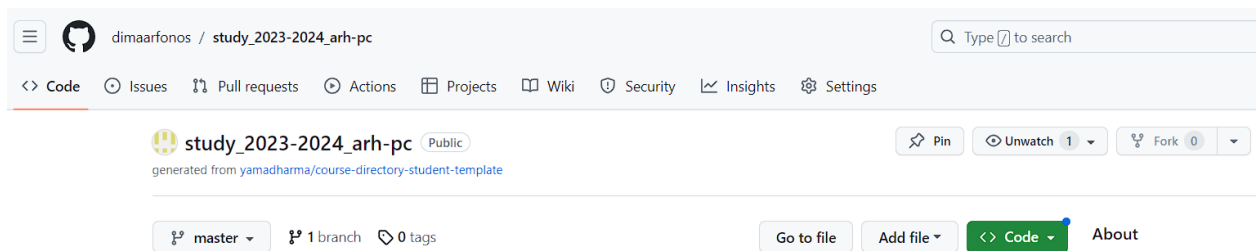


Рис. 4.2. Создали репозиторий курса на основе шаблона через web-интерфейс github.

```
[darfonos@fedora ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"
[darfonos@fedora Архитектура компьютера]$ git clone --recursive git@github.com:dimaarfonos/study_2023-2024_arh-pc.git
arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 КиБ | 16.93 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/darfonos/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 1.12 МиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/darfonos/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 2.34 МиБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
```

Рис. 4.3. Перешли в каталог курса и клонировали созданный на сайте репозиторий в новый каталог *arch-pc*.

```
[darfonos@fedora Архитектура компьютера]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[darfonos@fedora arch-pc]$ rm package.json
[darfonos@fedora arch-pc]$ echo arch-pc > COURSE
[darfonos@fedora arch-pc]$ make
```

Рис. 5. Перешли в новый каталог и удалили лишние файлы и создали необходимые каталоги.

```
[darfonos@fedora arch-pc]$ git add .
[darfonos@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 6bdafc3] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
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 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
```

```
[darfonos@fedora arch-pc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.59 МиБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:dimaarfonos/study_2023-2024_arh-pc.git
 4a13571..6bdafc3 master -> master
[darfonos@fedora arch-pc]$
```

Рис. 5.1. и рис 5.2. Вводим нужные команды и отправляем файлы на сервер.

Самостоятельная работа

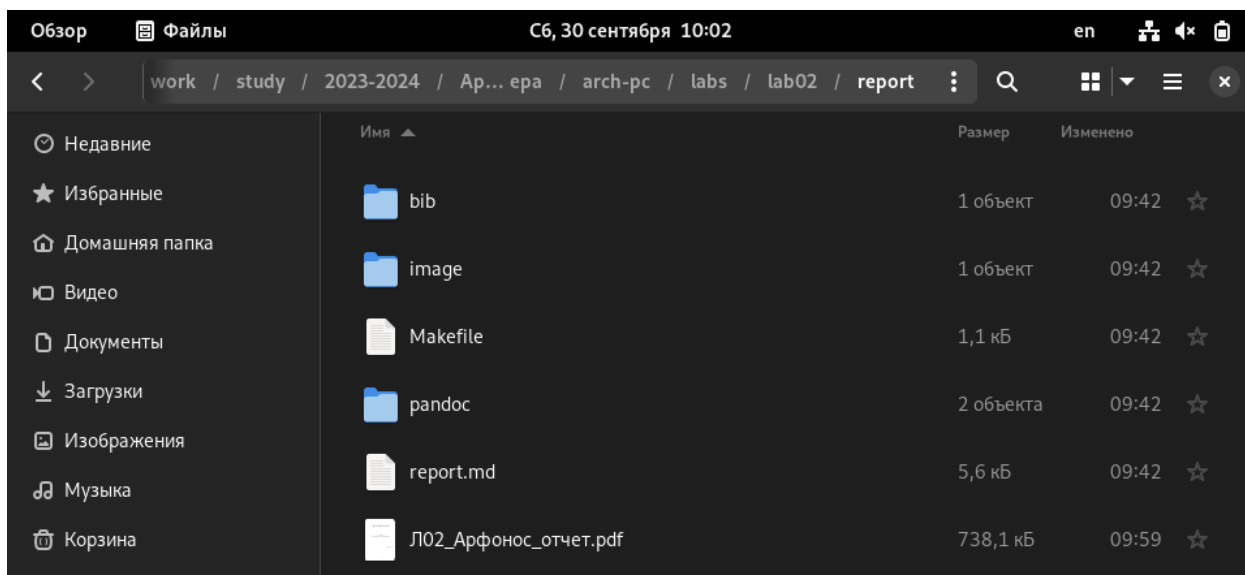


Рис. 1. Создаю отчет по выполнению лабораторной работы и вставляю соответствующий каталог *lab02*.

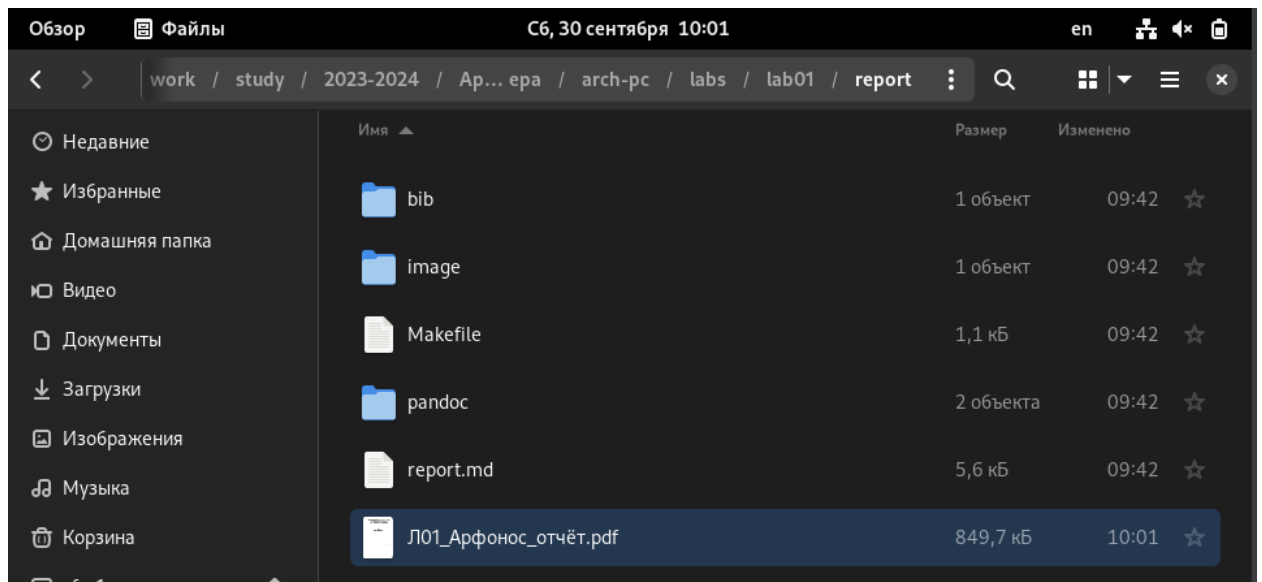


Рис. 2. Копирую предыдущий отчет по первой лабораторной работе в соответствующий ей каталог.

```
[darfonos@fedora arch-pc]$ git add .
[darfonos@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 7ad6a07] feat(main): make course structure
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Арфонос_отчёт.pdf
create mode 100644 labs/lab02/report/Л02_Арфонос_отчет.pdf
[darfonos@fedora arch-pc]$ git push
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (13/13), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 1.41 МиБ | 2.24 МиБ/с, готово.
Всего 9 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:dimaarfonos/study_2023-2024_arh-pc.git
   bf216d7..7ad6a07  master -> master
[darfonos@fedora arch-pc]$
```

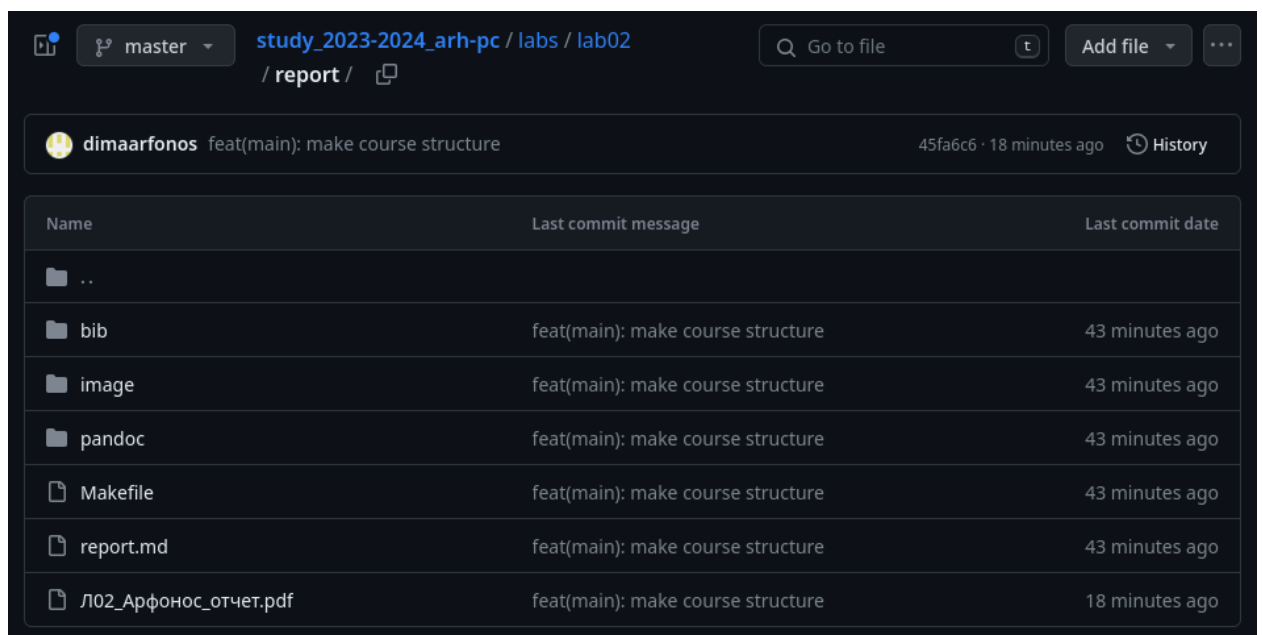


Рис. 3. и рис. 4. Загружаю файлы на *github*.

Вывод

В этой лабораторной работе я ознакомился с принципами работы средств контроля версий, настроил git для начала работы на линуксе. Используя git, создал рабочее пространство и репозиторий курса, после чего научился загружать файлы на github.

Вопросы для самопроверки

1. Системы контроля версий (VCS), а также системы управления версиями или SCM (Управление исходным кодом), представляют собой программные инструменты, предназначенные для внесения изменений в коды и версии файлов и проектов управления.
2. Хранилище (репозиторий) в системе контроля представляет собой центральное хранилище данных, в котором хранятся все версии файлов и история изменений проекта.
3.
 - Commit (или коммит) — это действие, при котором разработчик сохраняет (зафиксирует) текущее состояние рабочей копии проекта в хранилище. Коммит включает в себя измененные файлы, комментарии (описание) изменений и информацию об авторе. Каждый коммит создает новую версию проекта и записывает этот момент в историю.
 - Рабочая копия — это локальная копия проекта, с которой работают разработчики. Она представляет собой нынешнее состояние проекта на компьютере разработчика.
 - История в системе контроля представляет собой записи всех предыдущих коммитов и изменений, которые были внесены в проект. Она содержит информацию о том, кто, когда и что изменили в коде, а также комментарии к каждому коммиту. История позволяет разработчикам отслеживать эволюцию проекта и переходить к современной версии при необходимости.
4. Централизованные системы контроля управления (Centralized VCS): имеют большую центральную память, в которой хранятся все версии файлов и история изменений. Примеры: Subversion (SVN) и Perforce.

Децентрализованные системы контроля управления (Распределенная VCS): не имеют центрального хранилища. Вместо этого каждый разработчик имеет локальное хранилище (репозиторий) со всем аспектом проекта.

Примеры: Git, Mercurial и Bazaar.

5.

- Инициализация репозитория.
- Добавление файлов в репозиторий.
- Коммит изменений.
- Просмотр истории коммитов.
- Восстановление предыдущей версии.
- Синхронизация удаленными репозиториями (по желанию).

6.

- Клонирование репозитория :
- Работа с локальной копией :
- Дополнение и коммит изменений :
- Отправка изменений в удаленный репозиторий :
- Получение изменений из удаленного репозитория :
- Разумные решения (при необходимости) :
- Повторение шагов 3–6 :
- Завершение работы и сохранение изменений :

7. Основные задачи, решаемые инструментальной промывкой Git:

- Отслеживание версии и изменений в коде и проектах.

- Совместная работа и синхронизация изменений между разработчиками.
- Ветвление и слияние для разработки новых функций и управления разными версиями проекта.
- Восстановление предыдущей версии и управление существующим проектом.
- Резервное копирование и защита данных проекта от потери.

8. Команды git и их краткая характеристика

- git init:

Инициализирует новый локальный репозиторий Git в текущем каталоге.

- git clone <URL>:

Клонирует удаленные репозитории на локальном компьютере.

git add <файл>:

Добавляет изменения файла в индекс, готовя их к коммиту.

- git commit -m "Сообщение коммита":

Создан коммит с описанием изменений, зафиксированных в индексе.

- git status:

Показывает текущий статус рабочих копий и индекса (отслеживаемых изменений).

- git log:

Отображает историю коммитов в репозитории.

- git branch:

Показан список веток и текущую активную ветку.

- git checkout <ветка>:

Переключается на указанную ветку.

- git merge <ветка>:

Сливает изменения указанной ветки в текущую ветку.

- git pull:

Получает и решает изменения из удаленного репозитория в настоящее время.

- git push:

Отправляет изменения текущей ветки на удаленном репозитории.

- git remote:

Приводится список удаленных репозитория, категории с текущим .

9. Примеры использования Git при работе с локальными и удаленными репозиториями:

Локальный репозиторий:

- Инициализация локальной репозитории : *git init*
- Добавление файлов в индекс : *git add* <файл>

Удаленный репозиторий :

- Клонирование удаленного репозитория: *git clone* <url>
- Отправка изменений в удаленный репозиторий: *git push* <ветка >
- Получение изменений из удаленного репозитория: *git pull* "имя репозитория" <ветка>