

Лабораторная работа №2

НКАбд-01-22

Демидович Михайлович Никита

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	15
5	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	Команда git в терминале	7
3.2	Начало процесса генерации нового ssh-ключа	8
3.3	Содержимое нового файла ssh1.pub	8
3.4	Добавление нового ssh-ключа на github	9
3.5	Начало процесса генерации нового gpg-ключа	9
3.6	Добавление нового gpg-ключа на github	10
3.7	Добавление нового gpg-ключа на github	10
3.8	Копирование репозитория на сайте github	11
3.9	Репозитории на github	11
3.10	Создание каталога курса и его настройка	12
3.11	Настройка каталога курса	12
3.12	Создание персонального токена для авторизации	13
3.13	Авторизация с помощью токена и отправка файлов на сервер	13
3.14	Проверка выгрузки содержимого локального репозитория	13
3.15	Локальный репозиторий и репозиторий на github	14

Список таблиц

1 Цель работы

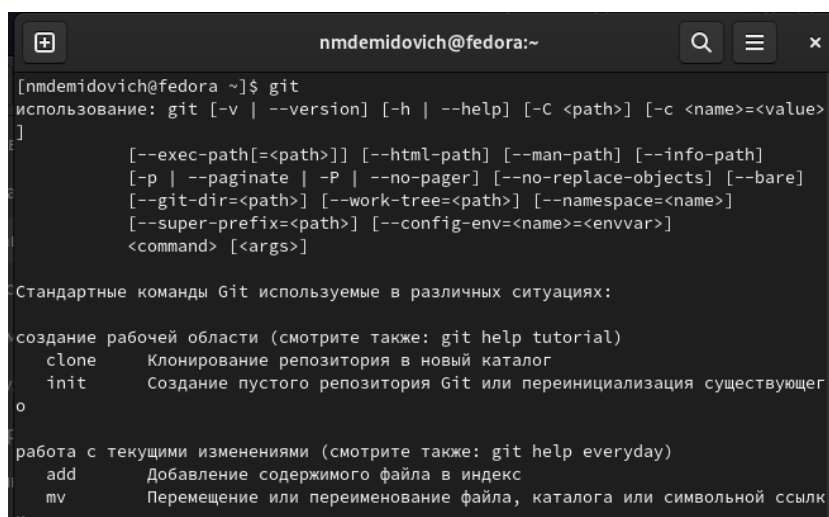
Изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

2 Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий по предмету.

3 Выполнение лабораторной работы

На первом этапе выполнения лабораторной работы необходимо произвести установку и базовую настройку git. Это было сделано ранее, при выполнении лабораторной работы №3 на курсе “Архитектура компьютера” (рис. [3.1]).



```
[nmdemidovich@fedora ~]$ git
использование: git [-v | --version] [-h | --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 существующег
o

работа с текущими изменениями (смотрите также: git help everyday)
  add        Добавление содержимого файла в индекс
  mv         Перемещение или переименование файла, каталога или символической ссылк
и
```

Рис. 3.1: Команда git в терминале

Далее я приступил к процессу создания нового репозитория для работ по курсу “Операционные системы”. И для начала мною был сгенерирован новый ssh-ключ (рис. [3.2]) и добавлен на github (рис. [3.4]).

```

nmdemidovich@fedora:~$ ssh-keygen -C "Демидович Никита <demid.nikita2016@yandex.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nmdemidovich/.ssh/id_rsa): ssh1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh1
Your public key has been saved in ssh1.pub
The key's randomart image is:
+----[RSA 3072]-----+
|      .o.o...      |
|      .. +ooo      |
|      ==+++=o      |
|      .o+=0=. + o   |
|      o +o*S. . * o |
|      = o+o      * o |
|      + +=      . ..|
|      . . .+      E |
|      .           |
+-----[SHA256]-----+
[nmdemidovich@fedora ~]$

```

Рис. 3.2: Начало процесса генерации нового ssh-ключа

```

+-----[SHA256]-----+
|      .           |
+-----[SHA256]-----+
[nmdemidovich@fedora ~]$ cd
[nmdemidovich@fedora ~]$ ls
pandoc-2.19                ssh1.pub      Музыка
pandoc-2.19-linux-amd64.tar.gz work          Общедоступные
pandoc-crossref            Видео        'Рабочий стол'
pandoc-crossref.1          Документы   Шаблоны
pandoc-crossref-Linux.tar.xz Загрузки
ssh1                       Изображения
[nmdemidovich@fedora ~]$ cat ssh1.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC8wBCuG2st+jj0uAWpD4U2z+3bichRTwT+vzp/9o/6
POWUH9Kb1f4qpDXLBGG49JP6gCQE3wMtTtqEFQrdCuYco0Zq2fDzrNRSUFgB5+dmT0LUJzKwOVZJtNpI
NoIiMVDECqP123MJX3xDtG8R9QKJOHP7Sz5GloPpN3OLAKjAqWZJta69IBtjZLT/frmyRTQSevbLMNSQ
K9rHhAy82+LFpzJArt27dR6XIp0NhZVn4QiCJfLy0hozLSty8TCdXm6Nci06c8xLZnkFlsMENBD6Q9/P
68Iaop3IX6Fe5BjQbnKujbvQ3VGVFBdLf+T3ty1s41iG/ipM+0L/BClqKZqSHkbCDUxyN7FV2m6MHwv
xhY4EW3aoDakmDeRW4Zl2GezgZNU/bJixFkQvym88jexj5VaUHCeulaJLPbwPf3u/9vRnqj80MskqE2K
aAIIfLunTA0x3FDdL9BN92de4GSCMTg9pr9GkAwa08kz+AUQIoQ9b+1qW9ANF0ajr54KLCic= Демидов
ич Никита <demid.nikita2016@yandex.ru>
[nmdemidovich@fedora ~]$

```

Рис. 3.3: Содержимое нового файла ssh1.pub

SSH keys / Add new

Title
ssh1

Key type
Authentication Key

Key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC8wBCuG2st+jj0uAWpD4U2z+3bichRTwT+vzp/9o
/6POWUHN9Kb1f4qpDXLBGG49JP6gCQE3wMTtqEFQrdCuYcoOZq2fDzrNRSUFgB5+dmT0LjzKwOVZjtNpINoliMVD
ECqP123MJX3xDtG8R9QKJONHP75z5GloPpN3OLAKjAqWZJta69IBtjZLT
/frmyRTQSevbLMNSQK9rHhAy82+LFpzJArt27dR6Xlp0NhZVn4QICjLy0hozLsty8TCdXm6Nci06c8xlZnkFlsMENBD6
Q9/P68Iqaop3IX6Fe5BjQbnKujbvQ3VGvFBdLf+T3ty1s411G/IpM+0L
/BCLqKZqShkbCDUxyN7FV2m6MHWxhy4EW3aoDakmDeRW4Zl2GezgZNU
/bjixFkQvym88jexj5VaUHceulajLPbwPF3u
/9vRnqj8OMskqE2KaAifLUntAOx3FDdL9BN92de4GSCMTg9pr9GkAwa08kz+AUQIoQ9b+1qW9ANF0ajr54KICic=
Демидович Никита <demid.nikita2016@yandex.ru>

Add SSH key

Рис. 3.4: Добавление нового ssh-ключа на github

Далее я создал gpg-ключ (рис. [3.5]).

```
[nmdemidovich@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.7; 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.

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
```

Рис. 3.5: Начало процесса генерации нового gpg-ключа

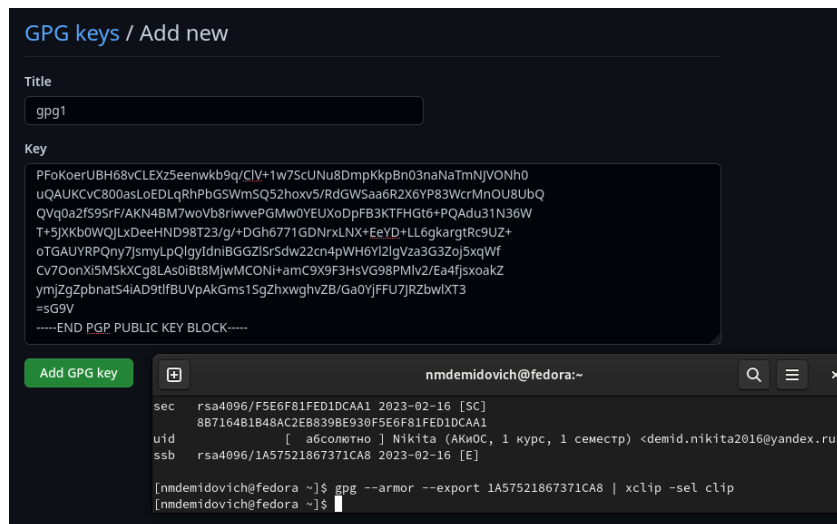


Рис. 3.6: Добавление нового gpg-ключа на github

P.S: В описании была допущена ошибка: семестр не первый, а второй (уже исправлено).

Затем мною была произведена настройка автоматических подписей коммитов git (рис. [3.7]).

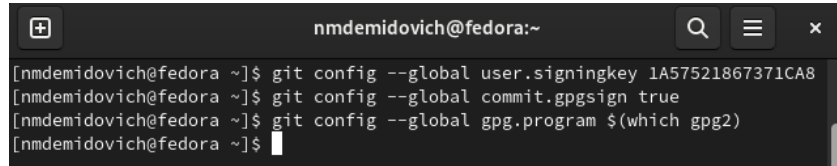




Рис. 3.7: Добавление нового gpg-ключа на github

И создан репозиторий курса на основе шаблона (рис. [3.9]).


Create a new repository from course-directory-student-template
The new repository will start with the same files and folders as [yamadharma/course-directory-student-template](#).


Owner * Repository name *

 nikdem1 / 


Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-winner?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☐ **Include all branches**
Copy all branches from yamadharma/course-directory-student-template and not just master.

 You are creating a public repository in your personal account.

[Create repository from template](#)

Рис. 3.8: Копирование репозитория на сайте github

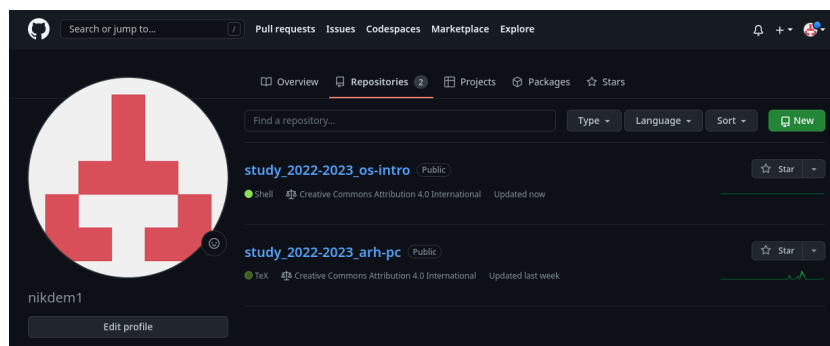


Рис. 3.9: Репозитории на github

```
nmdemidovich@fedora:~/work/study/2022-2023/Операционные системы$ git clone --recursive https://github.com/nikdem1/study_2022-2023_os-intro
Клонирование в «study_2022-2023_os-intro»...
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 | 1.88 МиБ/с, готово.
Определение изменений: 100% (1/1), 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/nmdemidovich/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro/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 | 970.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/nmdemidovich/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro/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 | 1.11 МиБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
[nmdemidovich@fedora Операционные системы]$
```

Рис. 3.10: Создание каталога курса и его настройка

```
nmdemidovich@fedora:~/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro$ cd study_2022-2023_os-intro
[nmdemidovich@fedora study_2022-2023_os-intro]$ ls
CHANGELOG.md  config  COURSE  LICENSE  Makefile  package.json  README.en.md  README.git-flow.md  README.md  template
[nmdemidovich@fedora study_2022-2023_os-intro]$ rm package.json
[nmdemidovich@fedora study_2022-2023_os-intro]$ echo os-intro > COURSE
[nmdemidovich@fedora study_2022-2023_os-intro]$ make
[nmdemidovich@fedora study_2022-2023_os-intro]$ git add
[nmdemidovich@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make course structure'
[master 568424d] feat(main): make course structure
361 files changed, 100327 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
```

Рис. 3.11: Настройка каталога курса

После этого я создал персональный токен для авторизации в терминале и отправил файлы на сервер (рис. [3.13])

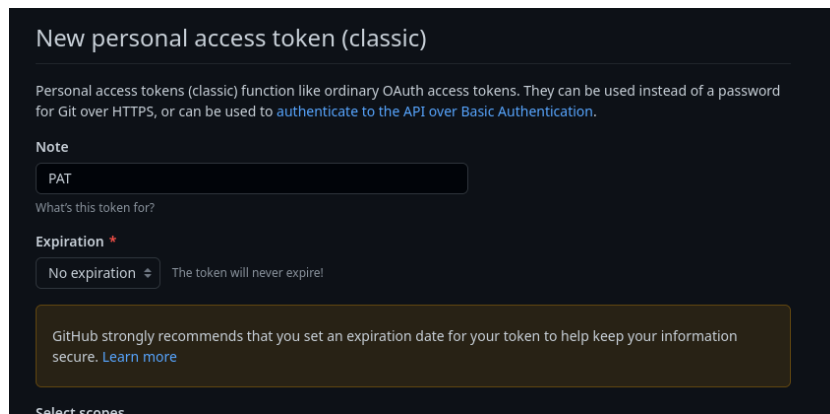


Рис. 3.12: Создание персонального токена для авторизации

```
nmdemidovich@fedora:~/work/study/2022-2023/Операционные системы/study_202...  
[nmdemidovich@fedora study_2022-2023_os-intro]$ git push  
Username for 'https://github.com': nikdem1  
Password for 'https://nikdem1@github.com':  
Перечисление объектов: 40, готово.  
Подсчет объектов: 100% (40/40), готово.  
Сжатие объектов: 100% (30/30), готово.  
Запись объектов: 100% (38/38), 343.04 КиБ | 13.72 МиБ/с, готово.  
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0  
remote: Resolving deltas: 100% (4/4), completed with 1 local object.  
To https://github.com/nikdem1/study_2022-2023_os-intro  
73008d3..8813152 master -> master  
[nmdemidovich@fedora study_2022-2023_os-intro]$
```

Рис. 3.13: Авторизация с помощью токена и отправка файлов на сервер

```
nmdemidovich@fedora:~/work/study/2022-2023/Операционн...  
[nmdemidovich@fedora study_2022-2023_os-intro]$ git pull  
Уже актуально.  
[nmdemidovich@fedora study_2022-2023_os-intro]$
```

Рис. 3.14: Проверка выгрузки содержимого локального репозитория

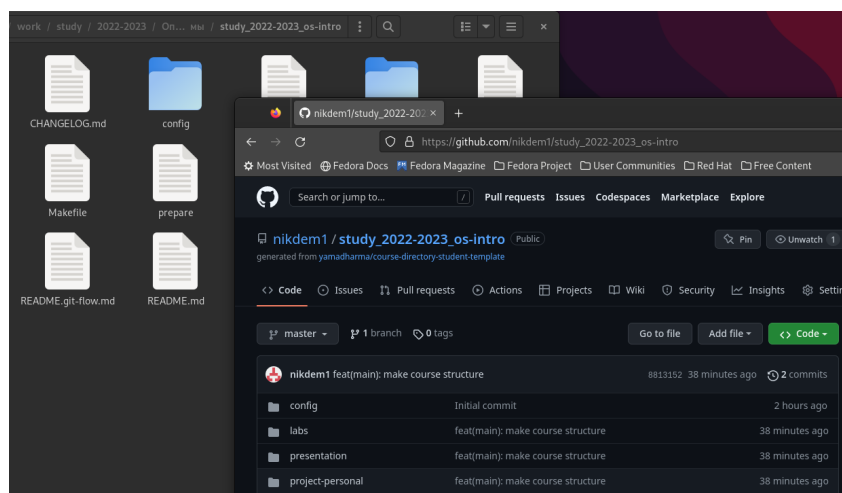


Рис. 3.15: Локальный репозиторий и репозиторий на github

4 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?

Система контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. 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. Однако они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище – репозиторий - место хранения всех версий и служебной информации. Commit - это команда для записи индексированных изменений в репозиторий. История – место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах. Рабочая копия – текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы – это системы, в которых одно основное хранилище всего проекта, и каждый пользователь копирует необходимые ему файлы, изменяет и вставляет обратно. Пример – Subversion. Децентрализованные системы – система, в которой каждый пользователь имеет свой вариант репозитория

и есть возможность добавлять и забирать изменения из репозитория. Пример – Git.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Система контроля версий (Version Control System, VCS) — это программное обеспечение для облегчения работы с изменяющейся информацией. Репозиторий (или хранилище) - в нем хранятся все документы вместе с историей их изменения и другой служебной информацией. Тэг commit позволяет сохранять и изменять изменения в репозитории. Рабочая копия - копия проекта, связанная с репозиторием.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

4. Опишите действия с VCS при единоличной работе с хранилищем.

В рабочей копии, которую исправляет человек, появляются правки, которые отправляются в хранилище на каждом из этапов. То есть в правки в рабочей копии появляются, только если человек делает их (отправляет их на сервер) и никак по-другому.

5. Опишите порядок работы с общим хранилищем VCS.

Если хранилище общее, то в рабочую копию каждого, кто работает над проектом, приходят изменения, отправленные на сервер одним из команды. Рабочая правка каждого может изменяться вне зависимости от того, делает ли конкретный человек правки или нет.

6. Каковы основные задачи, решаемые инструментальным средством git?

У Git две основных задачи: первая — хранить информацию обо всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

7. Назовите и дайте краткую характеристику командам git.

– 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 изменений: `git diff` – сохранение текущих изменений: `git add .` – добавить все изменённые и/или созданные файлы и/или каталоги: `git add` – добавить конкретные изменённые и/или созданные файлы и/или каталоги: `git add` – удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): `git rm` имена_файлов – 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 изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки` – слияние ветки с текущим деревом: `git merge --no-ff имя_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. Приведите примеры использования при работе с локальным и удалённым репозиториями.

Работа с удалённым репозиторием: `git remote` – просмотр списка настроенных удалённых репозиториев.

Работа с локальным репозиторием: `git status` – выводит информацию обо всех изменениях, внесенных в дерево директорий проекта по сравнению с последним коммитом рабочей ветки.

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветка (англ. branch) — это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала. Ветки нужны, чтобы несколь-

ко программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

10. Как и зачем можно игнорировать некоторые файлы при commit?

Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. В Git нет специальной команды для указания игнорируемых файлов: вместо этого необходимо вручную отредактировать файл . Временно игнорировать изменения в файле можно командой `git update-index--assumeunchanged`.

5 Выводы

В результате выполнения данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git.

Список литературы

Лабораторная работа №2 (Архитектура ОС).