

Отчет по лабораторной работе №2

Операционные системы

Пакавира Арсениу

Нкабд-04-23

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.2	Базовая настройка git	7
3.3	Создание ключа SSH	8
3.4	Создание ключа GPG	9
3.5	Регистрация на Github	11
3.6	Добавление ключа GPG в Github	11
3.7	Настроить подписи Git	13
3.8	Настройка gh	13
3.9	Создание репозитория курса на основе шаблона	14
4	Выводы	17
5	Ответы на контрольные вопросы.	18
	Список литературы	21

Список иллюстраций

3.1	Установка git и gh.....	Erro! Indicador não definido.
3.2	Задаю имя и email владельца репозитория	6
3.3	Настройка utf-8 в выводе сообщений git	7
3.4	Задаю имя начальной ветки	7
3.5	Задаю параметры autocrlf и safecrlf	7
3.6	Генерация ssh ключа по алгоритму rsa	8
3.7	Генерация ssh ключа по алгоритму ed25519	9
3.8	Генерация ключа.....	10
3.9	Защита ключа GPG.....	11
3.10	Аккаунт на Github	12

3.11 Вывод списка ключей	12
3.12 Копирование ключа в буфер обмена	12
3.13 Настройки GitHub	13
3.14 Добавление нового PGP ключа	13
3.15 Добавленный ключ GPG	Erro! Indicador não definido.
3.16 Настройка подписей Git	Erro! Indicador não definido.
3.17 Авторизация в gh	Erro! Indicador não definido.
3.18 Завершение авторизации через браузер	14
3.19 Завершение авторизации.....	Erro! Indicador não definido.
3.20 Создание репозитория.....	15
3.21 Перемещение между директориями	15
3.22 Удаление файлов и создание каталогов	16
3.23 Отправка файлов на сервер	16
3.24 Отправка файлов на сервер	17

Список таблиц

1 Цель работы

Цель данной лабораторной работы – изучение идеологии и применения средств контроля версий, освоение умения по работе с git.

2 Задание

1. Создать базовую конфигурацию для работы с git
2. Создать ключ SSH
3. Создать ключ GPG
4. Настроить подписи Git
5. Зарегистрироваться на GitHub
6. Создать локальный каталог для выполнения заданий по предмету.

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Устанавливаю необходимое программное обеспечение git и gh через терминал с помощью команд: `dnf install git` и `dnf install gh`

```
[alpakavira@fedora ~]$ sudo dnf -y install git
[sudo] пароль для alpakavira:
Fedora 34 - x86_64 - Updates          13 kB/s | 5.0 kB      00:00
Fedora Modular 34 - x86_64 - Updates 27 kB/s | 4.9 kB      00:00
Пакет git-2.34.3-1.fc34.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[alpakavira@fedora ~]$
```

Установка git и gh

3.2 Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту

```
[alpakavira@fedora ~]$ git config user.name " Pacavira Arsenio"
fatal: not in a git directory
[alpakavira@fedora ~]$ git config --global user.name " Pacavira Arsenio"
[alpakavira@fedora ~]$ git config --global user.email " 1032225105@pfur.ru"
[alpakavira@fedora ~]$
```

Задаю имя и email владельца репозитория

Настраиваю utf-8 в выводе сообщений git для их корректного отображения

```
[alpakavira@fedora ~]$ git config --global core.quotePath false
[alpakavira@fedora ~]$
```

Настройка utf-8 в выводе сообщений git

Начальной ветке задаю имя master

```
[alpakavira@fedora ~]$ git config --global init.defaultbranch master  
[alpakavira@fedora ~]$
```

Задаю имя начальной ветки

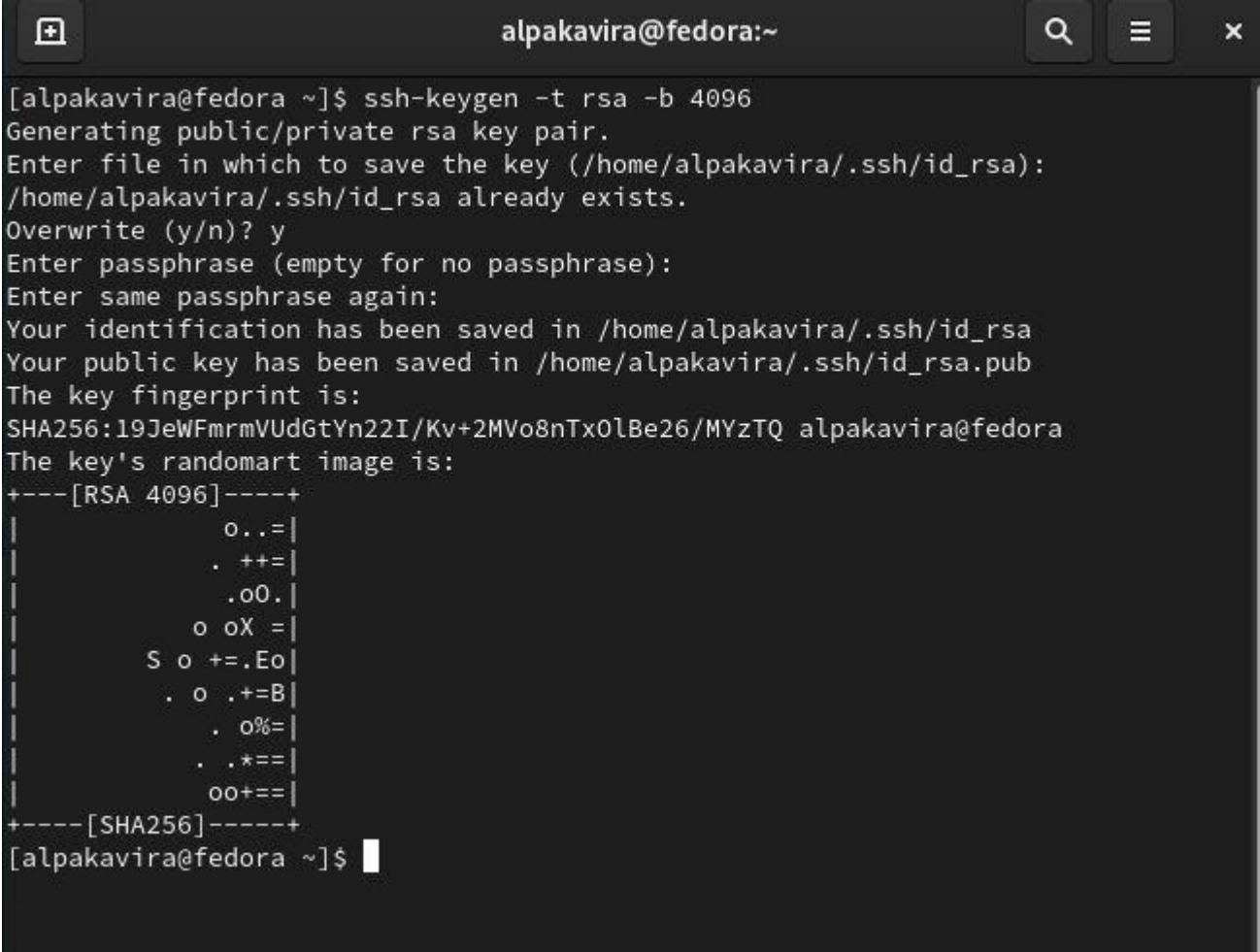
Задаю параметры autocrlf и safecrlf для корректного отображения конца строки

```
[alpakavira@fedora ~]$ git config --global core.safecrlf warn  
[alpakavira@fedora ~]$
```

Задаю параметры autocrlf и safecrlf

3.3 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa

A terminal window titled 'alpakavira@fedora:~' with search, menu, and close icons in the title bar. The terminal shows the execution of 'ssh-keygen -t rsa -b 4096'. It prompts for a file to save the key, which already exists at '/home/alpakavira/.ssh/id_rsa'. It asks to overwrite (y/n) and the user enters 'y'. It prompts for a passphrase, which is entered twice. It then displays the key fingerprint: 'SHA256:19JewFmrMVUdGtYn22I/Kv+2MVo8nTx0lBe26/MYzTQ alpakavira@fedora'. Finally, it shows the key's randomart image for RSA 4096 and SHA256.

```
[alpakavira@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alpakavira/.ssh/id_rsa):
/home/alpakavira/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alpakavira/.ssh/id_rsa
Your public key has been saved in /home/alpakavira/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:19JewFmrMVUdGtYn22I/Kv+2MVo8nTx0lBe26/MYzTQ alpakavira@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|          o..=|
|          . +=|
|          .o0.|
|          o oX =|
|        S o +=.Eo|
|          . o .+=B|
|          . o%=|
|          . .*==|
|          oo+==|
+----[SHA256]-----+
[alpakavira@fedora ~]$
```

Генерация ssh ключа по алгоритму rsa

Создаю ключ ssh по алгоритму ed25519


```

[alpakavira@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/alpakavira/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alpakavira/.ssh/id_ed25519
Your public key has been saved in /home/alpakavira/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:q68nnHWrJLAyKCluJQk7bKldlX/wqUFC4GVygpsgVeA alpakavira@fedora
The key's randomart image is:
+--[ED25519 256]--+
| .o+=.+          |
|o... *.         |
|..Eo... .       |
|. o   + o        |
|o.o .. +So .    |
|oO ..o   +.=    |
|Bo=....oo= .    |
|=..o   ++o .    |
|..      o=o.     |
+-----[SHA256]-----+

```

Генерация ssh ключа по алгоритму ed25519

3.4 Создание ключа GPG

Генерирую ключ GPG, затем выбираю тип ключа RSA and RSA, задаю максимальную длину ключа: 4096, оставляю неограниченный срок действия ключа.

Далее отвечаю на вопросы программы о личной информации

```

[alpakavira@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.2.27; 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/alpakavira/.gnupg'
gpg: создан щит с ключами '/home/alpakavira/.gnupg/pubring.kbx'
Выберите тип ключа:
  (1) RSA и RSA (по умолчанию)
  (2) DSA и Elgamal
  (3) DSA (только для подписи)
  (4) RSA (только для подписи)
  (14) Имеющийся на карте ключ
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

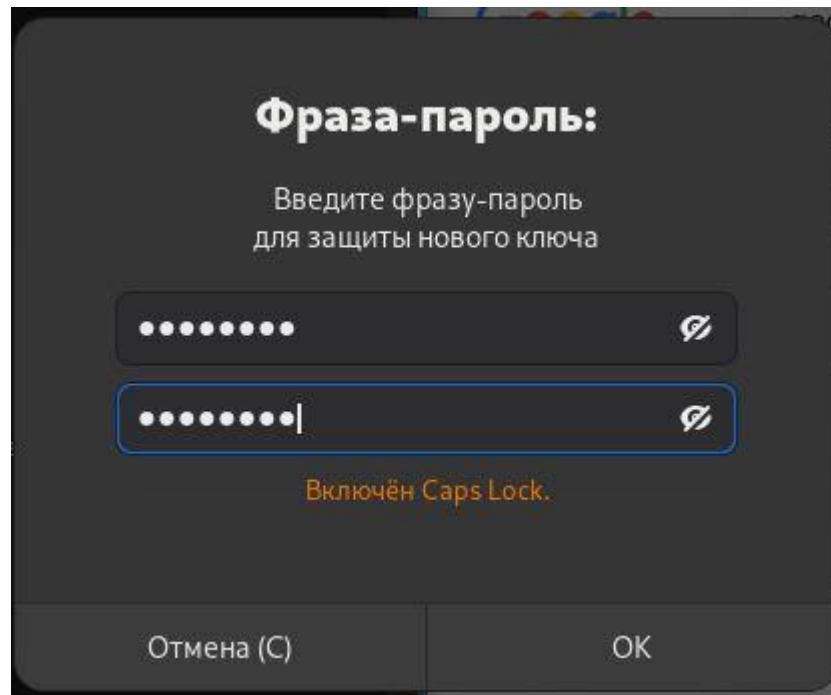
GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: alpakavira
Адрес электронной почты: 1032225105@pfur.ru

```

Генерация ключа

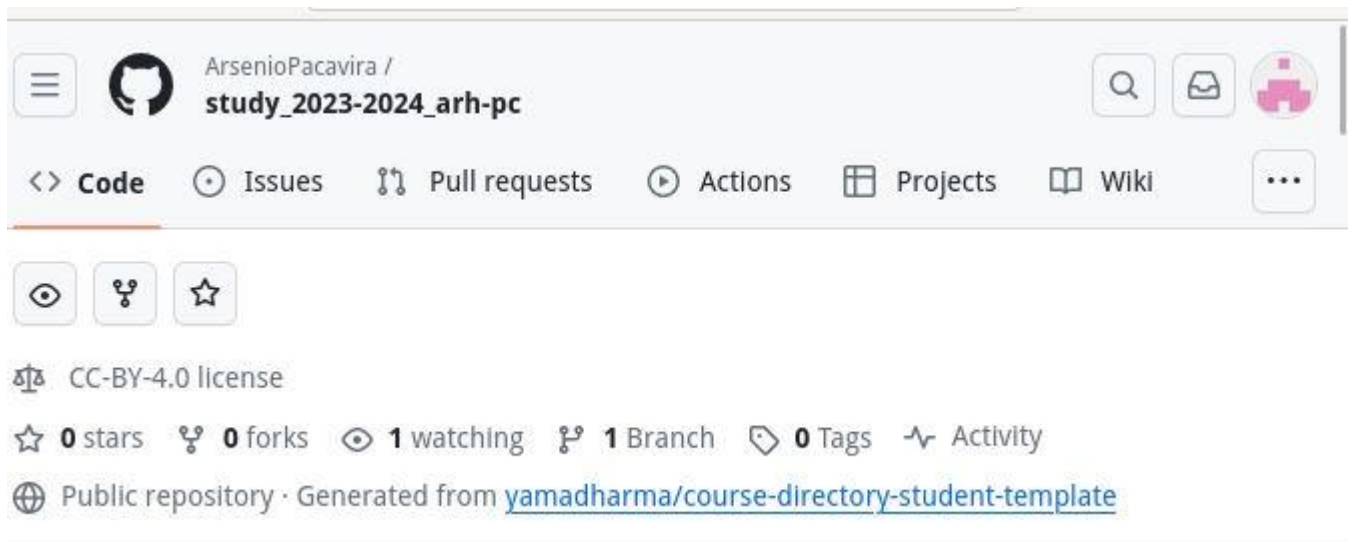
Ввожу фразу-пароль для защиты нового ключа



Защита ключа GPG

3.5 Регистрация на Github

У меня уже был создан аккаунт на Github, соответственно, основные данные аккаунта я так же заполняла и проводила его настройку, поэтому просто вхожу в свой аккаунт



3.6 Добавление ключа GPG в Github

Вывожу список созданных ключей в терминал, ищу в результате запроса отпечаток ключа (последовательность байтов для идентификации более длинного, по сравнению с самим отпечатком, ключа), он стоит после знака следа, копирую его в буфер обмена

```
[alpakavira@fedora ~]$ 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: 3 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 3u
/home/alpakavira/.gnupg/pubring.kbx
-----
sec   rsa4096/231145341ECFEAB9 2024-03-02 [SC]
      0D4385E47C95904D6216DF8E231145341ECFEAB9
uid   [ абсолютно ] Pacavira (pacavira) <1032225105@pfur.ru>
ssb   rsa4096/D923A84554E89D1E 2024-03-02 [E]

sec   rsa4096/75D9DC01BB784EC6 2024-03-02 [SC]
      A67B28CE8F74A3AE3EC1457975D9DC01BB784EC6
uid   [ абсолютно ] Pacavira (Pacavira) <1032225105@pfur.ru>
ssb   rsa4096/E8DE833C8C8EB932 2024-03-02 [E]

sec   rsa4096/10E02BE0C1DBD900 2024-03-02 [SC]
      1C6E92DBB77D46160745574210E02BE0C1DBD900
uid   [ абсолютно ] Pacavira (Pacavira) <1032225105@pfur.ru>
ssb   rsa4096/686FBBB5A3BF2698 2024-03-02 [E]
```

Вывод списка ключей

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена, за это отвечает утилита xclip

```
[alpakavira@fedora ~]$ gpg --armor --export E2FFC767DA4458F | xclip -sel clip
```

Копирование ключа в буфер обмена

Открываю настройки Github, ищу среди них добавление GPG ключа

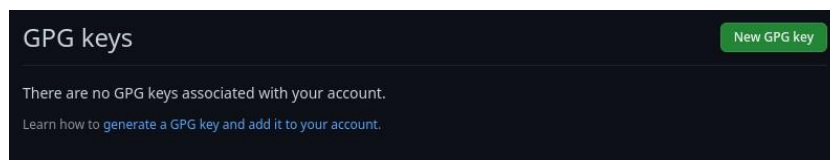
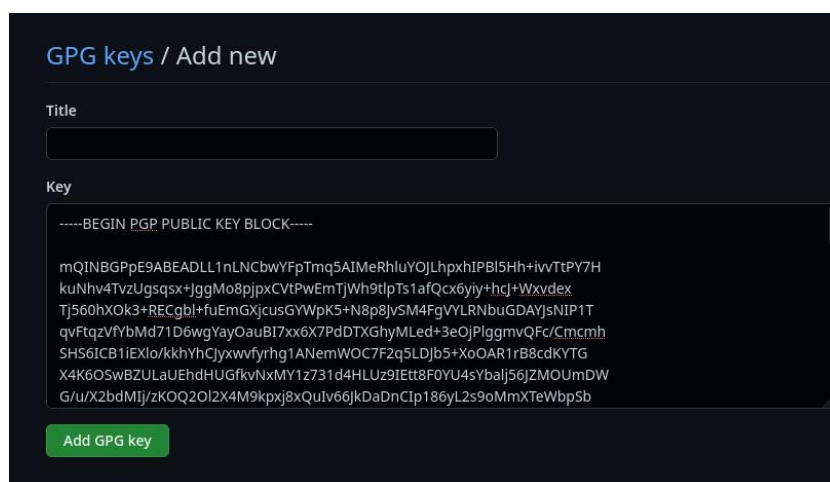


Рис. 3.13: Настройки GitHub

Нажимаю на “New GPG key” и вставляю в поле ключ из буфера обмена (рис. 3.14).



Добавление нового PGP ключа



Congratulations, you're all set!

Your device is now connected.

Завершение авторизации через браузер

3.9 Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты `mkdir` и флага `-p`, который позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты `cd` перехожу в только что созданную директорию “Операционные системы”. Далее в терминале ввожу команду `gh repo create study_2023-2024_os-intro --template yamadharm/course-directory-student-trmplate --public`, чтобы создать репозиторий на основе шаблона репозитория. После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом `https`, а не `ssh`, потому что при авторизации в `gh` выбрала протокол `https`


```
[alpakavira@fedora Операционные системы]$ git clone --recursive git@github.com:ArsenioPacavira/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 КиБ | 6.20 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/alpakavira/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.99 КиБ | 936.00 КиБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/home/alpakavira/work/study/2022-2023/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Получение объектов: 100% (126/126), 335.80 КиБ | 1.60 МиБ/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca'
```

Создание репозитория

Перехожу в каталог курса с помощью утилиты `cd`, проверяю содержание каталога с помощью утилиты `ls`

```
[alpakavira@fedora Операционные системы]$ cd os-intro
[alpakavira@fedora os-intro]$ ls
CHANGELOG.md  COURSE  Makefile      README.en.md  README.md
config        LICENSE package.json  README.git-flow.md  template
[alpakavira@fedora os-intro]$
```

Перемещение между директориями

Удаляю лишние файлы с помощью утилиты `rm`, далее создаю необходимые каталоги используя `makefile`

```
[alpakavira@fedora os-intro]$ rm package.json
[alpakavira@fedora os-intro]$ echo os-intro > COURSE
[alpakavira@fedora os-intro]$ make
Usage:
  make <target>

Targets:
  list                List of courses
  prepare             Generate directories structure
  submodule            Update submules
```

Удаление файлов и создание каталогов

Добавляю все новые файлы для отправки на сервер (сохраняю добавленные изменения) с помощью команды `git add` и комментирую их с помощью `git commit`

```
[alpakavira@fedora os-intro]$ git add
Ничего не указано, ничего не добавлено.
подсказка: Maybe you wanted to say 'git add .'
подсказка: Turn this message off by running
подсказка: "git config advice.addEmptyPathspec false"
[alpakavira@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master b7faa33] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
```

Отправка файлов на сервер

Отправляю файлы на сервер с помощью `git push`

```
[alpakavira@fedora os-intro]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 291 байт | 291.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ArzenioPacavira/study_2023-2024_os-intro.git
 4c70e44..b7faa33 master -> master
```


Отправка файлов на сервер

4 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, освоила умение по работе с git.

5 Ответы на контрольные вопросы.

1. Системы контроля версий (VCS)-программное обеспечение для облегчения работы с изменяющейся информацией. Они позволяют хранить несколько версий изменяющейся информации, одного и того же документа, может предоставить доступ к более ранним версиям документа. Используется для работы нескольких человек над проектом, позволяет посмотреть, кто и когда внес какое-либо изменение и т.д. VCS применяются для: Хранения полной истории изменений, сохранения причин всех изменений, поиска причин изменений и совершивших изменение, совместной работы над проектами.
2. Хранилище – репозиторий, хранилище версий, в нем хранятся все документы, включая историю их изменения и прочей служебной информацией. commit – отслеживание изменений, сохраняет разницу в изменениях. История – хранит все изменения в проекте и позволяет при необходимости вернуться/обратиться к нужным данным. Рабочая копия – копия проекта, основанная на версии из хранилища, чаще всего последней версии.
3. Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать изменения из любого репозитория. В отличие от классических, в распределенных (децентрализованных) системах контроля версий центральный репозиторий не является обязательным.
4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.

5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.
6. Хранение информации о всех изменениях в вашем коде, обеспечение 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 изменений: `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 изменений 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. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
 9. Ветвление-один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т.е. их концами возможно их слияние. Используются для разработки новых функций.
 10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

Список литературы

1. Лабораторная работа №2 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/page/view.php?id=12345>