

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Операционные системы»

Выполнил:

Студент Чванова Ангелина Дмитриевна

Студенческой группы НПИбд-02-21

Студенческий билет № 1032212282

Москва 2022

Лабораторная работа № 2. Управление версиями

Цель работы: изучить идеологию и применение средств контроля версий, а также освоить умения по работе с git.

1. Установка git-flow в Fedora Linux

Установка проходила вручную с помощью данных команд(рис1.1-1.2)

```
cd /tmp
```

```
wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes
```

```
↪ /gitflow/develop/contrib/gitflow-installer.sh
```

```
chmod +x gitflow-installer.sh
```

```
sudo ./gitflow-installer.sh install stable
```

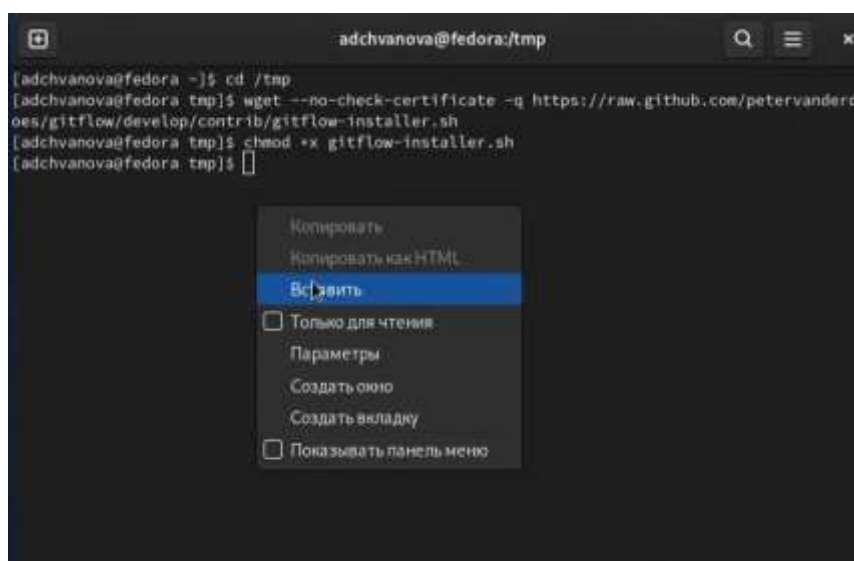


Рис.1.1 терминал с введенными командами для установки git-flow

```
adchvanova@fedora:/tmp — sudo ./gitflow-installer.sh install stable

[adchvanova@fedora ~]$ cd /tmp
[adchvanova@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[adchvanova@fedora tmp]$ chmod +x gitflow-installer.sh
[adchvanova@fedora tmp]$ sudo ./gitflow-installer.sh install stable

Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

№1) Уважайте частную жизнь других.
№2) Думайте, прежде что-то вводить.
№3) С большой властью приходит большая ответственность.

[sudo] пароль для adchvanova:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
```

Рис.1.2 терминал с введенными командами для установки git-flow

2. Установка gh в Fedora Linux с помощью команды

sudo dnf install gh (рис2)

```
adchvanova@fedora:/tmp — sudo dnf install gh

'gitflow/hooks/pre-flow-release-start' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-release-start'
'gitflow/hooks/pre-flow-release-track' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-release-track'
[adchvanova@fedora tmp]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 0:24:00 назад, Чт 21 апр 2022 21:48:12.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh         x86_64      2.7.0-1.fc35  updates      6.8 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 6.8 М
Объем изменений: 32 М
Продолжить? [д/н]: д
Загрузка пакетов:
gh-2.7.0-1.fc35.x86_64.rpm 43% [===== 1 601 kB/s | 3.0 MB 00:06 ETA
```

Рис.2 терминал с введенной командой для установки gh

3.Базовая настройка git(рис3.1-3.2)

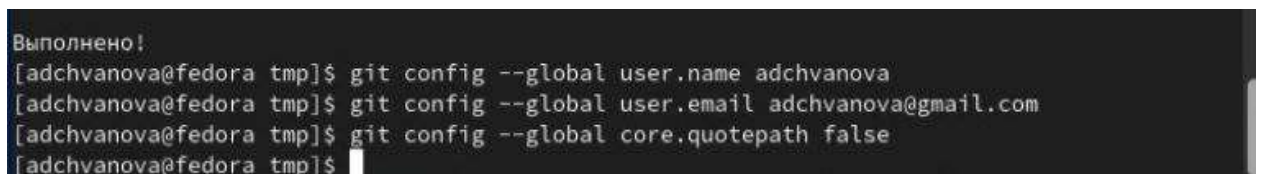
– Задаем имя и email владельца репозитория с помощью команд:

```
git config --global user.name "Name Surname"
```

```
git config --global user.email "work@mail"
```

Настройка utf-8 в выводе сообщений git:

```
git config --global core.quotepath false
```



Выполнено!
[adchvanova@fedora tmp]\$ git config --global user.name adchvanova
[adchvanova@fedora tmp]\$ git config --global user.email adchvanova@gmail.com
[adchvanova@fedora tmp]\$ git config --global core.quotepath false
[adchvanova@fedora tmp]\$

Рис.3.1 Ввод имени владельца и email , а также настройка utf-8

Настройте верификацию и подписание коммитов git.

– Задаем имя начальной ветки (будем называть её master):

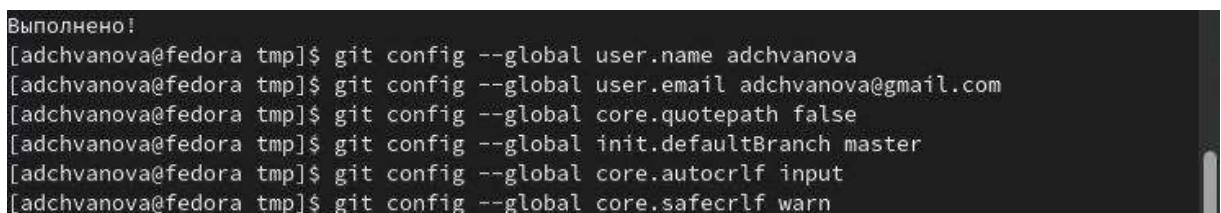
```
git config --global init.defaultBranch master
```

– Параметр autocrlf:

```
git config --global core.autocrlf input
```

– Параметр safecrlf:

```
git config --global core.safecrlf warn
```



Выполнено!
[adchvanova@fedora tmp]\$ git config --global user.name adchvanova
[adchvanova@fedora tmp]\$ git config --global user.email adchvanova@gmail.com
[adchvanova@fedora tmp]\$ git config --global core.quotepath false
[adchvanova@fedora tmp]\$ git config --global init.defaultBranch master
[adchvanova@fedora tmp]\$ git config --global core.autocrlf input
[adchvanova@fedora tmp]\$ git config --global core.safecrlf warn

Рис.3.2 Выбор имени начальной ветки и настройка параметров safecrlf и autocrlf

4.Создание ключей ssh(рис4)

Ключ создавался по алгоритму rsa с размером для ключа 4096 бит:

ssh-keygen -t rsa -b 4096

```
[adchvanova@fedora tmp]$ git config --global core.safecrlf warn
[adchvanova@fedora tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/adchvanova/.ssh/id_rsa):
Created directory '/home/adchvanova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adchvanova/.ssh/id_rsa
Your public key has been saved in /home/adchvanova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/drqRyh0p0N8rDFTxU2Y2R2LSUCu3XGalyJXpChWwAE adchvanova@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|      .++*oX+|
|      o.= X =|
|      + E = +|
|      o.@ = * .|
|      .S+.% * o|
|      . =.+ o |
|      . o. |
|      o. |
|      .+o. |
+-----[SHA256]-----+
[adchvanova@fedora tmp]$
```

Рис.4 Создание ключа ssh по алгоритму rsa с размером для ключа 4096 бит в консоле

5.Создание ключей pgp(рис5.1-5.2)

– Генерируем ключ

gpg --full-generate-key

– Из предложенных опций (рис5.1) выбираем:

– тип RSA and RSA;

– размер 4096;

– выберите срок действия; значение по умолчанию— 0 (срок действия не истекает

никогда).

– GPG запросит личную информацию, которая сохранится в ключе:

– Имя (не менее 5 символов).

– Адрес электронной почты.

– При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.

– Комментарий.

```
(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.
```

Рис.5.1 консоль с введенной командой для создания ключей gpg и предложенные опции

```

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: adchvanova
Адрес электронной почты: adchvanova@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "adchvanova <adchvanova@gmail.com>"

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

```

Рис.5.2 выбранные опции для создания ключей pgp

6.Добавление PGP ключа в GitHub(рис6.1-6.3)

– Выводим список ключей и копируем отпечаток приватного ключа:

gpg --list-secret-keys --keyid-format LONG

– Формат строки (рис6.1):

sec Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до] ID_ключа

```

sec  rsa4096/93DDF3A55158768F 2022-04-21 [SC]
    D47BBD4A856F85E430C6DD9893DDF3A55158768F
uid      [ абсолютно ] adchvanova <adchvanova@gmail.com>
ssb  rsa4096/C54F75CBDFFFDD02 2022-04-21 [E]

```

Рис.6.1 Сгенерированный в консоли ключ

– Копирование сгенерированного PGP ключа в буфер обмена(рис6.2):

1 gpg --armor --export <PGP Fingerprint> | xclip -sel clip

```
[adchvanova@fedora tmp]$ gpg --armor --export 93DDF3A55158768F | xclip -sel clip
```

Рис.6.2 Копирование в консоле сгенерированного PGP ключа в буфер обмена

GPG keys / Add new

Key

```
8VAI  
dv5Y9JGT0As6hl5fZCFWrKsjovt2op2WyXPX0C41pnA8D9gjwdM8WdC80K  
dvEVQW  
/KmELJqTAAOnYz0CON37Z38sw4UMBFbiOa6GgM7N5UZL1vvEvRyJjZ6TXn  
GZlu43U  
EA==  
==+5UX  
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

Рис.6.3 вставка полученного ключа в поле ввода (переход в настройки GitHub (<https://github.com/settings/keys>))

7.Настройка автоматических подписей коммитов git (рис7)

– Используя введённый email, указываем Git применять его при подписи КОММИТОВ:

```
git config --global user.signingkey <PGP Fingerprint>
```

```
git config --global commit.gpgsign true
```

```
git config --global gpg.program $(which gpg2)
```

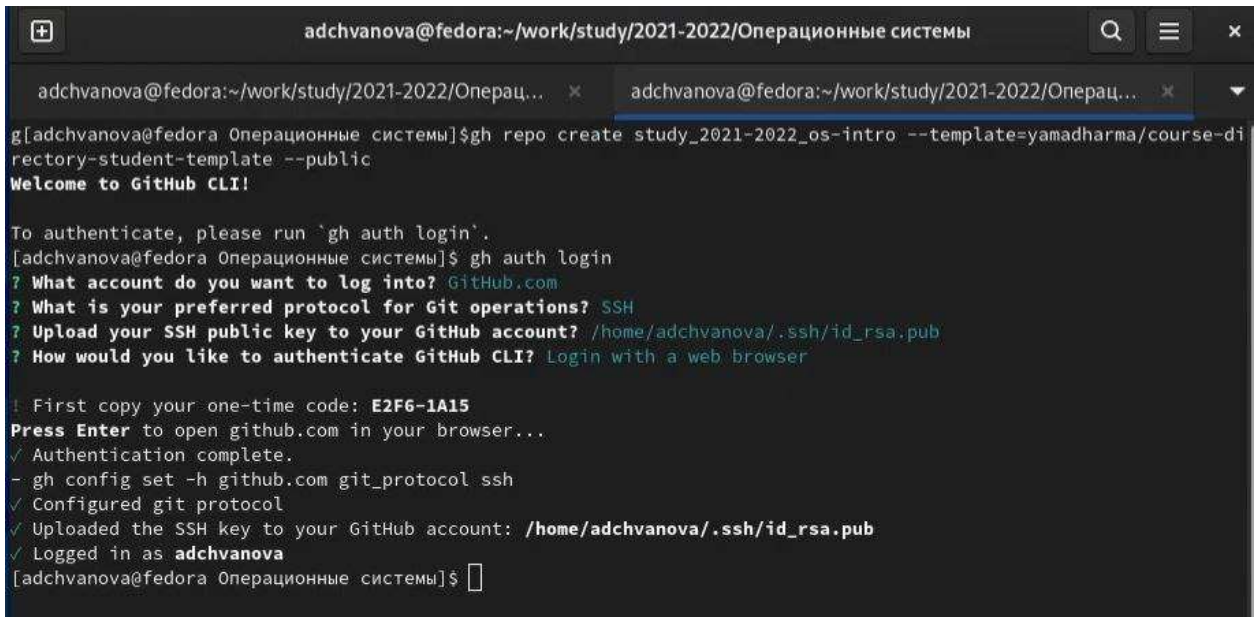
```
[adchvanova@fedora tmp]$ git config --global user.signingkey 93DDF3A55158768F  
[adchvanova@fedora tmp]$ git config --global commit.gpgsign true  
[adchvanova@fedora tmp]$ git config --global gpg.program $(which gpg2)
```

Рис.7 терминал с введенными командами для настройки автоматических подписей коммитов git

8.Настройка gh(рис8)

–команда для авторизации

gh auth login

A terminal window on a Fedora system showing the process of authenticating with GitHub CLI. The user runs 'gh repo create' with a template, then 'gh auth login'. The CLI prompts for account selection, protocol (SSH), and authentication method (web browser). A one-time code 'E2F6-1A15' is displayed. The user is then logged in as 'adchvanova' and the git protocol is configured to 'ssh'.

```
adchvanova@fedora:~/work/study/2021-2022/Операционные системы
g[adchvanova@fedora Операционные системы]$gh repo create study_2021-2022_os-intro --template=yamadharma/course-di
rectory-student-template --public
Welcome to GitHub CLI!

To authenticate, please run 'gh auth login'.
[adchvanova@fedora Операционные системы]$ 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/adchvanova/.ssh/id_rsa.pub
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: E2F6-1A15
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/adchvanova/.ssh/id_rsa.pub
✓ Logged in as adchvanova
[adchvanova@fedora Операционные системы]$
```

Рис.8 Авторизация в gh

9.Создание репозитория курса на основе шаблона и настройка каталога курса (рис9.1-9.2)

—создание шаблона рабочего пространства.

```
mkdir -p ~/work/study/2021-2022/"Операционные системы"
```

```
cd ~/work/study/2021-2022/"Операционные системы"
```

```
gh repo create study_2021-2022_os-intro
```

```
↪ --template=yamadharma/course-directory-student-template --public
```

```
git clone --recursive
```

```
↪ git@github.com:<owner>/study_2021-2022_os-intro.git os-intro
```

– Переход в каталог курса:

```
cd ~/work/study/2021-2022/"Операционные системы"/os-intro
```

– Удаление лишних файлов(рис9.1):

`rm package.json`

– Создайте необходимые каталоги(рис9.1):

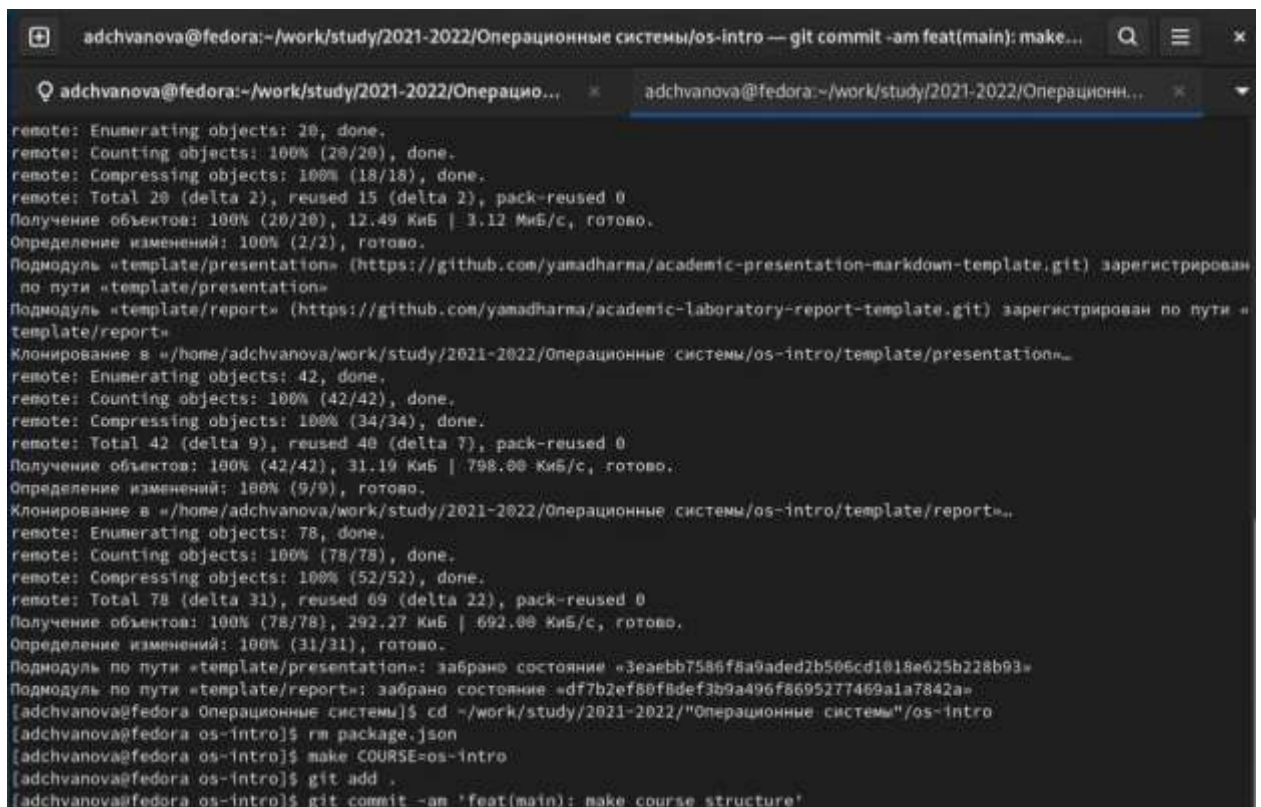
`make COURSE=os-intro`

– Отправьте файлы на сервер(рис9.1):

`git add .`

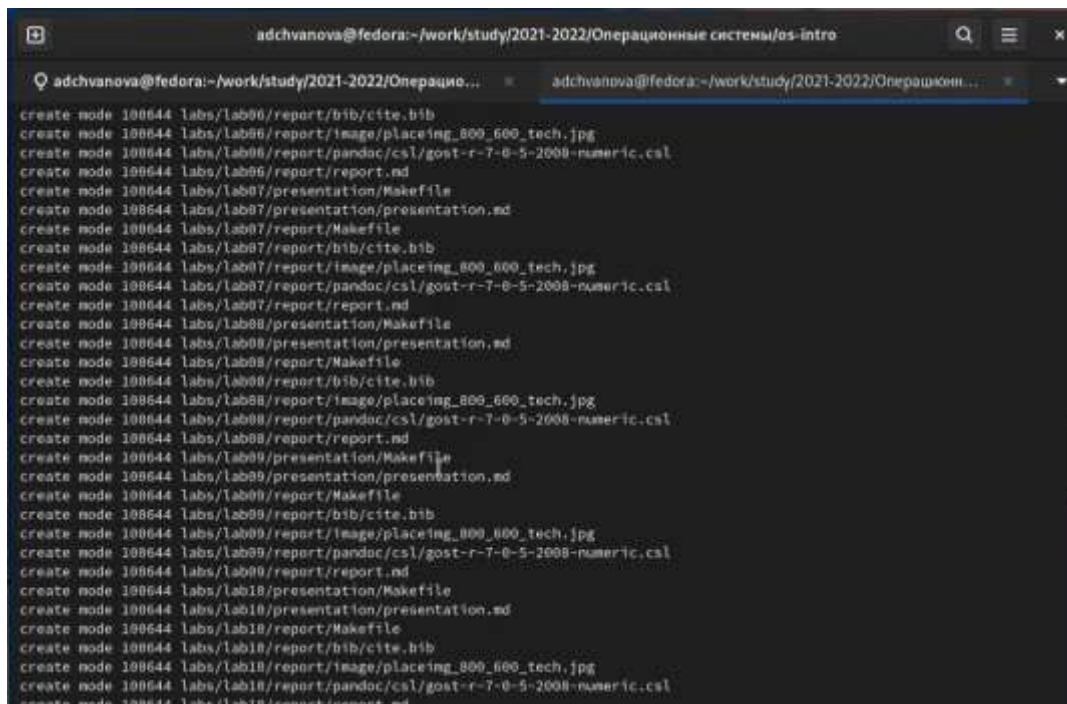
`git commit -am 'feat(main): make course structure'`

`git push`



```
adchvanova@fedora:~/work/study/2021-2022/Операционные системы/os-intro — git commit -am feat(main): make...
adchvanova@fedora:~/work/study/2021-2022/Операционн...
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 | 3.12 МБ/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/adchvanova/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 | 798.00 КиБ/с, готово.
Определение изменений: 100% (9/9), готово.
Клонирование в «/home/adchvanova/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 | 692.00 КиБ/с, готово.
Определение изменений: 100% (31/31), готово.
Подмодуль по пути «template/presentation»: забрано состояние «3eae6bb7586f8a9aded2b506cd1018e625b228b93»
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»
[adchvanova@fedora Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[adchvanova@fedora os-intro]$ rm package.json
[adchvanova@fedora os-intro]$ make COURSE=os-intro
[adchvanova@fedora os-intro]$ git add .
[adchvanova@fedora os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис.9.1 удаление лишних файлов, создание каталогов и отправка файлов на сервер



```
adchvanova@fedora:~/work/study/2021-2022/Операционные системы/os-intro
adchvanova@fedora:~/work/study/2021-2022/Операционные...
create mode 100644 labs/lab06/report/bib/cite.bib
create mode 100644 labs/lab06/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab06/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab06/report/report.nd
create mode 100644 labs/lab07/presentation/Makefile
create mode 100644 labs/lab07/presentation/presentation.md
create mode 100644 labs/lab07/report/Makefile
create mode 100644 labs/lab07/report/bib/cite.bib
create mode 100644 labs/lab07/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab07/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab07/report/report.nd
create mode 100644 labs/lab08/presentation/Makefile
create mode 100644 labs/lab08/presentation/presentation.md
create mode 100644 labs/lab08/report/Makefile
create mode 100644 labs/lab08/report/bib/cite.bib
create mode 100644 labs/lab08/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab08/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab08/report/report.nd
create mode 100644 labs/lab09/presentation/Makefile
create mode 100644 labs/lab09/presentation/presentation.md
create mode 100644 labs/lab09/report/Makefile
create mode 100644 labs/lab09/report/bib/cite.bib
create mode 100644 labs/lab09/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab09/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab09/report/report.nd
create mode 100644 labs/lab10/presentation/Makefile
create mode 100644 labs/lab10/presentation/presentation.md
create mode 100644 labs/lab10/report/Makefile
create mode 100644 labs/lab10/report/bib/cite.bib
create mode 100644 labs/lab10/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab10/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab10/report/report.nd
```

Рис9.2 отправка файлов на сервер

Вывод: были изучены некоторые команды в консоли для работы с GitHub и применены средства контроля версий, а также освоены умения по работе с git. Удалось создать репозиторий курса на основе шаблона и настроить его. Файлы были отправлены на сервер.

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 – место хранения файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов.

Commit позволяет отправлять изменения на сервер VCS.

История — список всех изменений проекта с возможностью отката в любую точку истории.

Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.

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

В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов. Обратите внимание, что нет единого объекта, который получает и отвечает на запрос.

Bitcoin. Давайте возьмем биткойны, например, потому что это самый популярный пример использования децентрализованных систем. Ни одна организация / организация не владеет сетью биткойнов. Сеть представляет собой сумму всех узлов, которые общаются друг с другом для поддержания количества биткойнов, которое есть у каждого владельца счета.

Централизованные системы — это системы, которые используют архитектуру клиент / сервер, где один или несколько клиентских узлов

напрямую подключены к центральному серверу. Это наиболее часто используемый тип систем во многих организациях, где клиент отправляет запрос на сервер компании и получает ответ.

Wikipedia. Рассмотрим огромный сервер, на который мы отправляем наши запросы, и сервер отвечает запрашиваемой статьей. Предположим, мы ввели поисковый запрос «нездоровая пища» в строке поиска Википедии. Этот поисковый запрос отправляется как запрос на серверы Википедии (в основном, расположенные в штате Вирджиния, США), которые затем возвращают статьи, основанные на релевантности. В этой ситуации мы являемся клиентским узлом, серверы Википедии являются центральным сервером.

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

Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория:

```
git config —global user.name"Имя Фамилия"
```

```
git config —global user.email"work@mail"
```

и настроив utf-8 в выводе сообщенийgit:

```
git config —global quotepath false
```

Для инициализации локального репозитория, расположенного, например, в каталоге ~/tutorial, необходимо ввести в командной строке:

```
cd
```

```
mkdir tutorial
```

```
cd tutorial
```

```
git init
```

5. Опишите порядок работы с общим хранилищем VCS.

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):

```
ssh-keygen -C"Имя Фамилия <work@mail>"
```

Ключи сохраняться в каталоге ~/.ssh/.

Скопировав из локальной консоли ключ в буфер обмена

```
cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

вставляем ключ в появившееся на сайте поле.

6. Каковы основные задачи, решаемые инструментальным средством git?

У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом. А также ведение истории изменений, фиксирование изменений, совмещение версий, веток, откат к прошлым версиям.

7. Назовите и дайте краткую характеристику командам git.

git init - инициализирует локальный репозиторий

git add * или add. - добавляет файлы в репозиторий

git commit - версия фиксации

git pull - загружает текущую версию проекта

git push - отправляет измененный проект на сервер

git checkout - позволяет переключаться между ветками

git status - текущий статус проекта

git branch - просмотреть доступные ветки

git remote add - добавить удаленный репозиторий

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

Использование git с локальными репозиториями используется для добавления, например, текстового файла в локальный репозиторий

```
git add file.txt
```

```
git commit -am ФАЙЛ
```

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

Ветви необходимы, чтобы иметь возможность "разделять" части работы и работать отдельно над каждой имплементацией. Использование ветвей дает возможность обрабатывать нововведения в основную ветвь, которая чаще всего является релизной.

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

Игнорировать файлы при commit можно с помощью .gitignore файла.

В нем указываются пути, названия, расширения и другие идентификации нежелательных объектов которые не будут учитываться в commit. Это полезно для исключения как "мусорных" файлов, которые не являются значимой частью проекта, а также конфиденциальных файлов, которые содержат в себе приватную информацию, такую как пароли и токены.