

Отчет по Лабораторной работе №2

Первоначальна настройка git

Касымов Заур

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
3.1	Установка программного обеспечения	8
3.1.1	Установка git(рис.3.1)	8
3.1.2	Установка gh(рис.3.2)	8
3.2	Настройка github	9
3.2.1	Создание репозитория курса на основе шаблона(рис.3.3, 3.4, 3.5, 3.6)	9
3.3	Настройка каталога курса	11
3.3.1	Перейдем в каталог курса(рис.3.7):	11
3.3.2	Удалим лишние файлы(рис.3.8):	12
3.3.3	Создадим необходимые каталоги(рис.3.9):	12
3.3.4	Отправим файлы на сервер(рис.3.10)	13
4	Выводы	14
	Ссылки	15

Список иллюстраций

3.1	Установим git	8
3.2	Фетровая шляпа	9
3.3	Качаем репозиторий	9
3.4	Создаем свой репозиторий	10
3.5	Проверяем	10

Список таблиц

1 Цель работы

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

2 Теоретическое введение

- Системы контроля версий. Общие понятия 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.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию 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, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными

участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

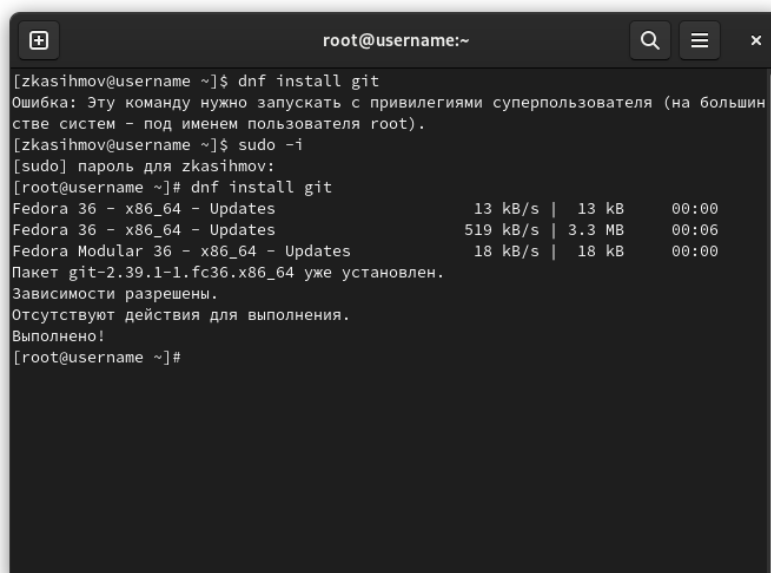
Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

3.1.1 Установка git(рис.3.1)

`dnf install git`



```
root@username:~  
[zkasihmov@username ~]$ dnf install git  
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на большин  
стве систем - под именем пользователя root).  
[zkasihmov@username ~]$ sudo -i  
[sudo] пароль для zkasihmov:  
[root@username ~]# dnf install git  
Fedora 36 - x86_64 - Updates          13 kB/s | 13 kB    00:00  
Fedora 36 - x86_64 - Updates          519 kB/s | 3.3 MB  00:06  
Fedora Modular 36 - x86_64 - Updates  18 kB/s | 18 kB    00:00  
Пакет git-2.39.1-1.fc36.x86_64 уже установлен.  
Зависимости разрешены.  
Отсутствуют действия для выполнения.  
Выполнено!  
[root@username ~]#
```

Рис. 3.1: Установим git

3.1.2 Установка gh(рис.3.2)

`dnf install gh`


```
root@username:~  
Fedora 36 - x86_64 - Updates 13 kB/s | 13 kB 00:00  
Fedora 36 - x86_64 - Updates 519 kB/s | 3.3 MB 00:06  
Fedora Modular 36 - x86_64 - Updates 18 kB/s | 18 kB 00:00  
Пакет git-2.39.1-1.fc36.x86_64 уже установлен.  
Зависимости разрешены.  
Отсутствуют действия для выполнения.  
Выполнено!  
[root@username ~]# dnf install gh  
Последняя проверка окончания срока действия метаданных: 0:00:29 назад, Чт 16 фев 2023 20:34:40.  
Зависимости разрешены.  
=====
```

Пакет	Архитектура	Версия	Репозиторий	Размер
Установка:				
gh	x86_64	2.22.1-1.fc36	updates	8.2 М

```
=====
```

Результат транзакции

Установка 1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]:

Рис. 3.2: Фетровая шляпа

3.2 Настройка github

3.2.1 Создание репозитория курса на основе шаблона(рис.3.3, 3.4, 3.5, 3.6)

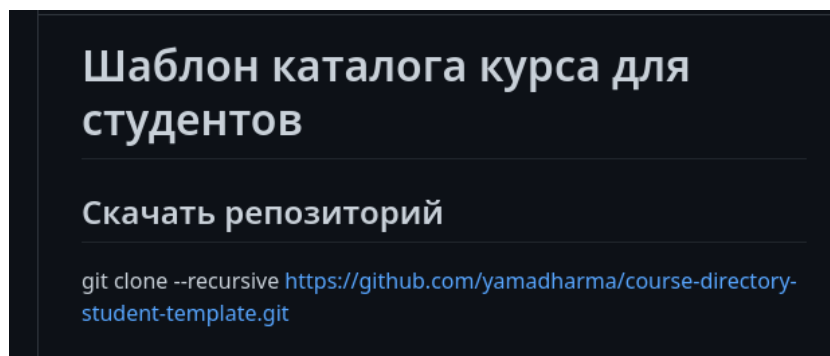


Рис. 3.3: Качаем репозиторий

Create a new repository from course-directory-student-template

The new repository will start with the same files and folders as yamadharm/course-directory-student-template.

Owner ^{*} ZaurKasymov / Repository name ^{*} study_2022-2023_os-intro ✓

Great repository names are short, lowercase, and contain only numbers, letters, and hyphens. study_2022-2023_os-intro is available. How about **crispy-umbrella**?

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 yamadharm/course-directory-student-template and not just master.

ⁱ You are creating a public repository in your personal account.

Create repository from template

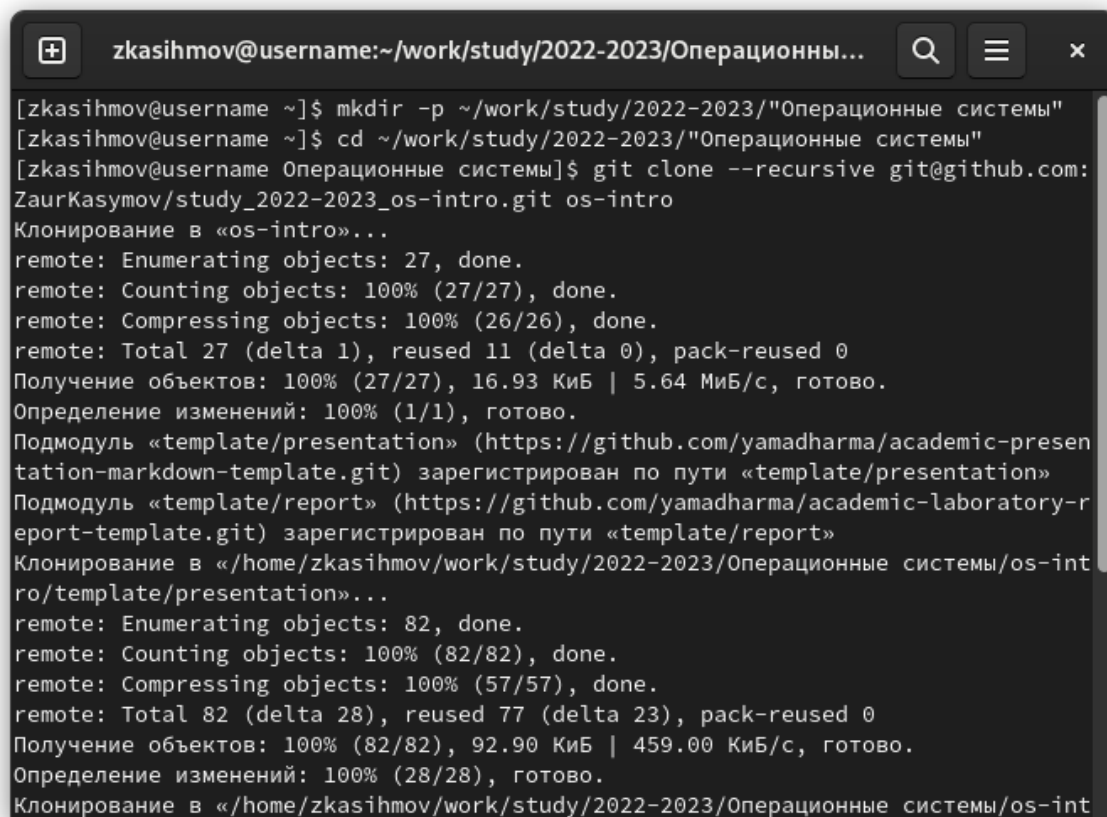
Рис. 3.4: Создаем свой репозиторий

master	1 branch	0 tags	Go to file	Add file	Code
ZaurKasymov Initial commit	e5b34fa	now	1 commit		
config	Initial commit	now			
template	Initial commit	now			
.gitattributes	Initial commit	now			
.gitignore	Initial commit	now			
.gitmodules	Initial commit	now			
CHANGELOG.md	Initial commit	now			
COURSE	Initial commit	now			
LICENSE	Initial commit	now			
Makefile	Initial commit	now			
README.en.md	Initial commit	now			
README.git-flow.md	Initial commit	now			
README.md	Initial commit	now			
package.json	Initial commit	now			

Рис. 3.5: Проверяем

Необходимо создать шаблон рабочего пространства (см. Рабочее пространство для лабораторной работы). Например, для 2022–2023 учебного года и предмета «Операционные системы» (код предмета os-intro) создание репозитория примет следующий вид `mkdir -p ~/work/study/2022-2023/“Операционные системы”`

```
cd ~/work/study/2022-2023/"Операционные системы" gh repo create study_2022-2023_os-intro --template=yamadharm/course-directory-student-template --public
git clone --recursive git@github.com:/study_2022-2023_os-intro.git os-intro
```



```
zkasihmov@username:~/work/study/2022-2023/Операционны...
[zkasihmov@username ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[zkasihmov@username ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[zkasihmov@username Операционные системы]$ git clone --recursive git@github.com:
ZaurKasymov/study_2022-2023_os-intro.git os-intro
Клонирование в «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 | 5.64 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presen
tation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-r
eport-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/zkasihmov/work/study/2022-2023/Операционные системы/os-int
ro/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 | 459.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/zkasihmov/work/study/2022-2023/Операционные системы/os-int
```

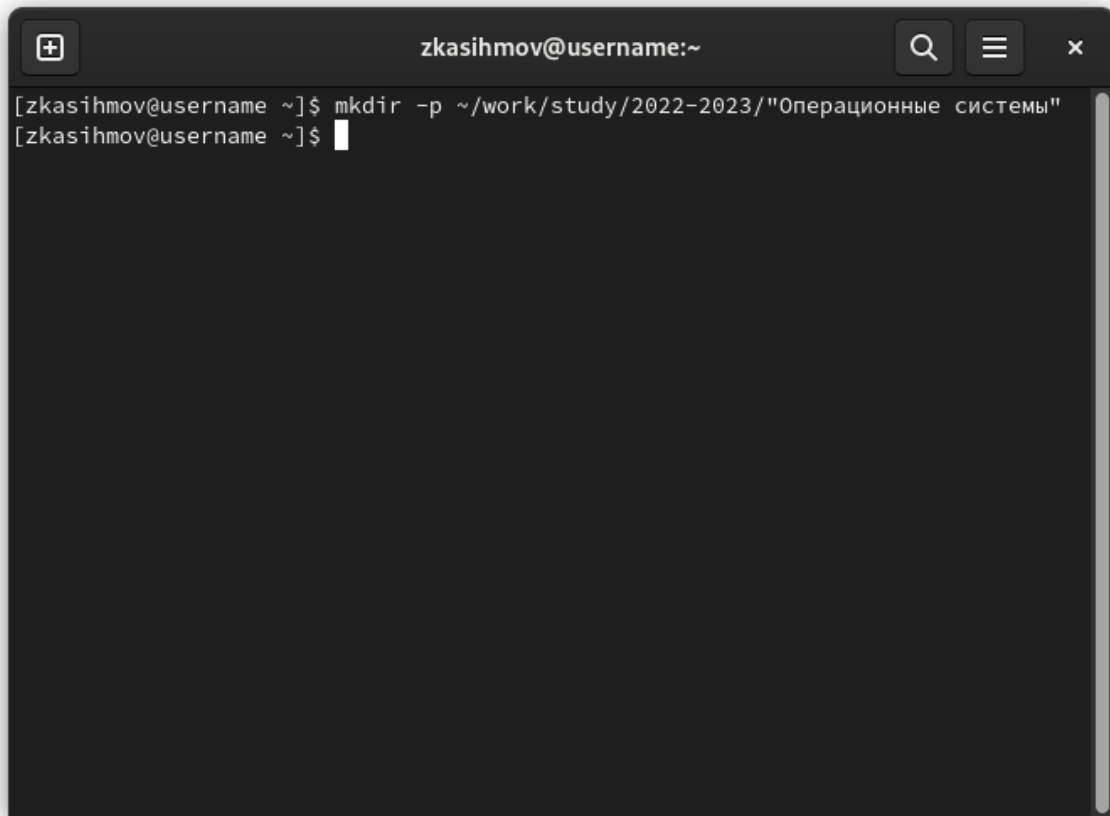
{#fig:001

width=70%

3.3 Настройка каталога курса

3.3.1 Перейдем в каталог курса(рис.3.7):

```
cd ~/work/study/2022-2023/"Операционные системы"/os-intro
```

A terminal window with a dark background. The title bar shows 'zkasihmov@username:~'. The terminal content shows two lines: the first line is '[zkasihmov@username ~]\$ mkdir -p ~/work/study/2022-2023/"Операционные системы"' and the second line is '[zkasihmov@username ~]\$' followed by a cursor. The window has standard macOS window controls (red, yellow, green buttons) and a search icon in the top right.

```
zkasihmov@username:~  
[zkasihmov@username ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"  
[zkasihmov@username ~]$
```

{#fig:001

width=70%

3.3.2 Удалим лишние файлы(рис.3.8):

```
rm package.json  
[zkasihmov@username os-intro]$ rm package.json  
[zkasihmov@username os-intro]$
```

{#fig:001

width=70%

3.3.3 Создадим необходимые каталоги(рис.3.9):

```
echo os-intro > COURSE make
```

```
[zkasihmov@username os-intro]$ ls
CHANGELOG.md  labs      prepare      README.en.md      template
config        LICENSE  presentation  README.git-flow.md
COURSE        Makefile  project-personal  README.md
[zkasihmov@username os-intro]$ ls labs
lab01 lab03 lab05 lab07 lab09 lab11 lab13 lab15 README.ru.md
lab02 lab04 lab06 lab08 lab10 lab12 lab14 README.md
[zkasihmov@username os-intro]$
```

{#fig:001

width=70%

3.3.4 Отправим файлы на сервер(рис.3.10)

```
root@username:~
Fedora 36 - x86_64 - Updates      13 kB/s | 13 kB    00:00
Fedora 36 - x86_64 - Updates      519 kB/s | 3.3 MB  00:06
Fedora Modular 36 - x86_64 - Updates 18 kB/s | 18 kB    00:00
Пакет git-2.39.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[root@username ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:29 назад, Чт 16 фев 2023 20:34:40.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.22.1-1.fc36  updates      8.2 М
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]:
```

{#fig:001

width=70%

4 Выводы

Изучили идеологию и применение средств контроля версий. Освоили умения по работе с git.

Ссылки

Ссылка на репозиторий: https://github.com/ZaurKasymov/study_2022-2023_os-intro.git