

Отчёт по лабораторной работе №2

Управление версиями

Озкан Аминат НБИбд-01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	10
4	Контрольные вопросы	11
	Список литературы	15

List of Figures

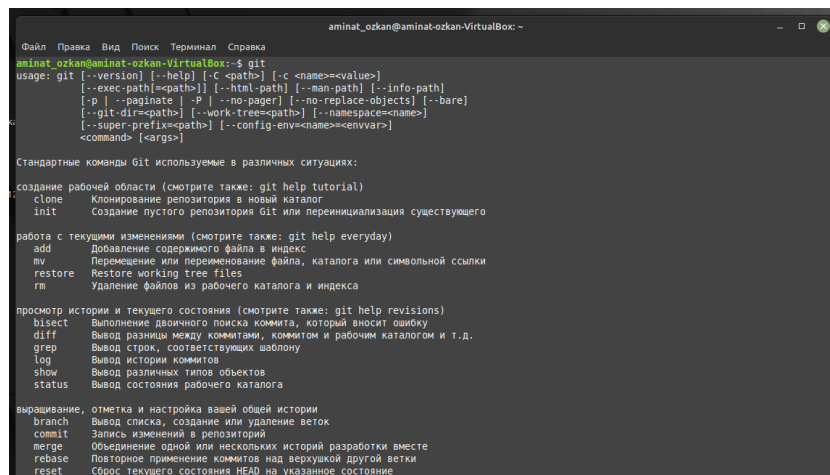
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	5
2.3	rsa-4096	6
2.4	ed25519	6
2.5	GPG ключ	7
2.6	GPG ключ	7
2.7	Параметры репозитория	7
2.8	Связь репозитория с аккаунтом	8
2.9	Загрузка шаблона	8
2.10	Первый коммит	9

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.



```
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git help
usage: git [--version] [--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>=<envvars>]
       <command> [<args>]

Стандартные команды Git используются в различных ситуациях:

создание рабочей области (смотрите также: git help tutorial)
  clone  Клонирование репозитория в новый каталог
  init   Создание пустого репозитория Git или переинициализация существующего

работа с текущими изменениями (смотрите также: git help everyday)
  add    Добавление содержимого файла в индекс
  mv     Перемещение или переименование файла, каталога или символической ссылки
  restore Restore working tree files
  rm     Удаление файлов из рабочего каталога и индекса

просмотр истории и текущего состояния (смотрите также: git help revisions)
  bisect Выполнение двоичного поиска коммита, который вносит ошибку
  diff   Вывод различий между коммитами, коммитом и рабочим каталогом и т.д.
  grep  Вывод строк, соответствующих шаблону
  log    Вывод истории коммитов
  show   Вывод различных типов объектов
  status Вывод состояния рабочего каталога

выращивание, отметка и настройка вашей общей истории
  branch Вывод списка, создание или удаление веток
  commit Запись изменений в репозиторий
  merge  Объединение одной или нескольких историй разработки вместе
  rebase Повторное применение коммитов над вершущей другой ветки
  reset  Сброс текущего состояния HEAD на указанное состояние

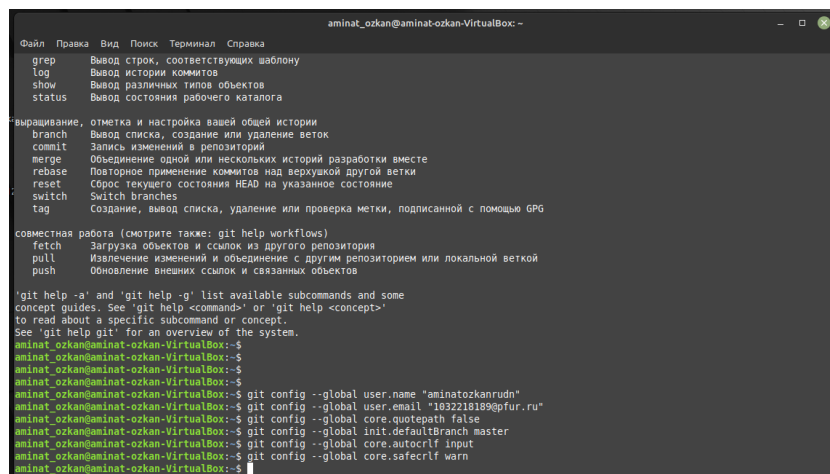
выращивание, отметка и настройка вашей общей истории
  branch Вывод списка, создание или удаление веток
  commit Запись изменений в репозиторий
  merge  Объединение одной или нескольких историй разработки вместе
  rebase Повторное применение коммитов над вершущей другой ветки
  reset  Сброс текущего состояния HEAD на указанное состояние
  switch Switch branches
  tag     создание, вывод списка, удаление или проверка метки, подписанной с помощью GPG

совместная работа (смотрите также: git help workflows)
  fetch  Загрузка объектов и ссылок из другого репозитория
  pull   Извлечение изменений и объединение с другим репозиторием или локальной веткой
  push   Обновление внешних ссылок и связанных объектов

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

Figure 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.



```
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git config --global user.name "aminatozkanrudn"
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git config --global user.email "1832218189@pfur.ru"
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git config --global core.quotepath false
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git config --global init.defaultbranch master
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git config --global core.autocrlf input
aminat_ozkan@aminat-ozkan-VirtualBox: ~$ git config --global core.safecrlf warn
aminat_ozkan@aminat-ozkan-VirtualBox: ~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global user.name "aminatozkanrudn"
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global user.email "1032218189@pfur.ru"
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global core.quotepath false
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global init.defaultBranch master
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global core.autocrlf input
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global core.safecrlf warn
aminat_ozkan@aminat-ozkan-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aminat_ozkan/.ssh/id_rsa):
Created directory '/home/aminat_ozkan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aminat_ozkan/.ssh/id_rsa
Your public key has been saved in /home/aminat_ozkan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zvrfPFWrARWE6yqjmfG0f0qisa0q3cocQYH3IQ/SSrw aminat_ozkan@aminat-ozkan-VirtualBox
The key's randomart image is:
+--[RSA 4096]--+
|++0      00.  |
|*+..      .  |
|0=0       0   |
|.E+       0   |
|.0.0 . S . . .|
|.++0 0 . . . .|
|.0.0+ 0 . . 0 |
|.+++. + + .. |
|=.++ 0.+++.  |
+-----[SHA256]-----
aminat_ozkan@aminat-ozkan-VirtualBox:~$
```

Figure 2.3: rsa-4096

```
aminat_ozkan@aminat-ozkan-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/aminat_ozkan/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aminat_ozkan/.ssh/id_ed25519
Your public key has been saved in /home/aminat_ozkan/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:EIghBqahMY7919wx00ja2X4b79mGH81tXfJ7ZrEE aminat_ozkan@aminat-ozkan-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|Xo .oo.  E  |
|Xo 0. . . .  |
|0 0 0 . . . 0|
|. + + + . + +|
|. + .050 . + .|
|. 0 B + . + .|
|. B 0 .      |
|. 0 0 0 0    |
|... 40mg    |
+-----[SHA256]-----
aminat_ozkan@aminat-ozkan-VirtualBox:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
aminat_ozkan@aminat-ozkan-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
* Ваше полное имя: aminat_ozkanrudn
Адрес электронной почты: 1032218189@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"aminat_ozkanrudn <1032218189@pfur.ru>"
* Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/aminat_ozkan/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ 5591471FC6DB19F4 помечен как абсолютно доверенный
gpg: создан каталог '/home/aminat_ozkan/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/aminat_ozkan/.gnupg/openpgp-revocs.d/0CC02FA8592B56F8E6525F6C5591471FC6DB19F4.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-08-27 [SC]
      0CC02FA8592B56F8E6525F6C5591471FC6DB19F4
uid           aminat_ozkanrudn <1032218189@pfur.ru>
sub   rsa4096 2022-08-27 [E]

aminat_ozkan@aminat-ozkan-VirtualBox:~$
aminat_ozkan@aminat-ozkan-VirtualBox:~$
```

Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

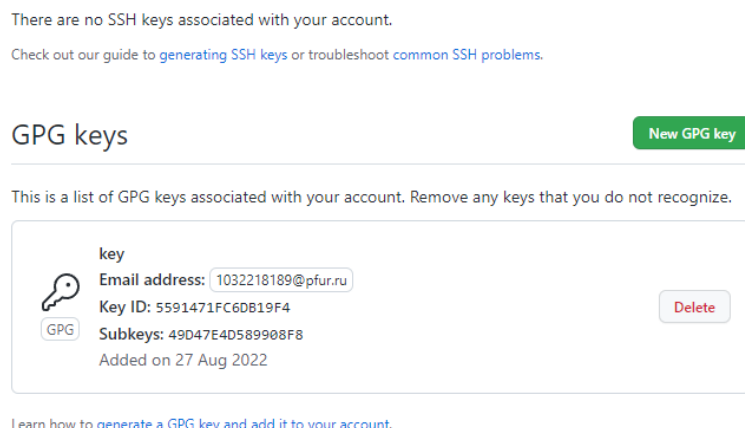


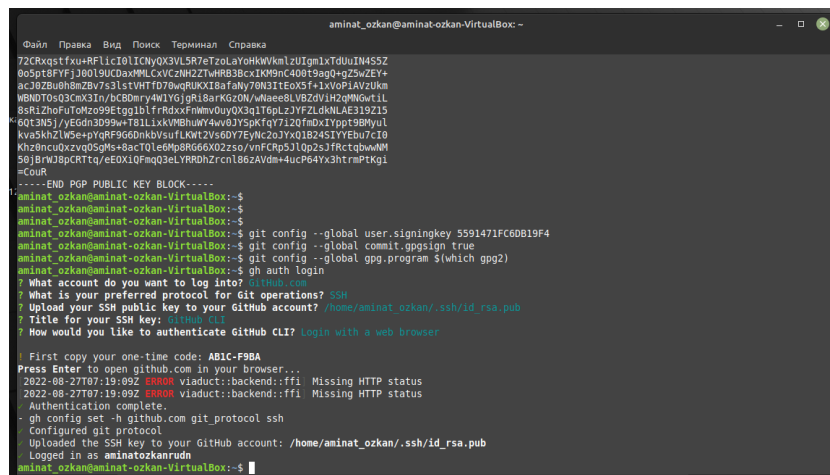
Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
acJ0ZB0u9hmZ8v7s3stVhT070wqRUKXl8afaly/8N3lfe0Xsf+lxv0P1AVZUkn
WBNDT05Q3cmX3In/bCB0mry4w1YgJgI8RkG20N/wMae8LVBZdVh2qMwGwtiL
8sR1ZhoFuToMz09Etgg1b1frRdxXfmmvduy0X3q1T6pLz3YFZLdkMLAE319Z15
603NSJ/y6d0n3D099w+T81LlxXMBHw4w03JSpkf7J20fndx1Yp9S8Byul
KvaSkh2Lwe+pvGRF9660nbvufkwtZv560Y7EjNc203yx01B2451YEEbu7z10
Kh20ncU0xzvQ5gMs+8acTQle6Mp8R666X02zso/vnFCRpSj1Qp2s3FRctqbwwMM
50jBrW38pCRRtq/e0X1QfmQ3eLYRRDhZrcnL86zAVdm+4ucP64Yx3htrmPtKg1
=Cour
-----END PGP PUBLIC KEY BLOCK-----
aminat_ozkan@aminat-ozkan-VirtualBox:~$
aminat_ozkan@aminat-ozkan-VirtualBox:~$
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global user.signingkey 5591471FC6DB19F4
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global commit.gpgsign true
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global gpg.program $(which gpg2)
aminat_ozkan@aminat-ozkan-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

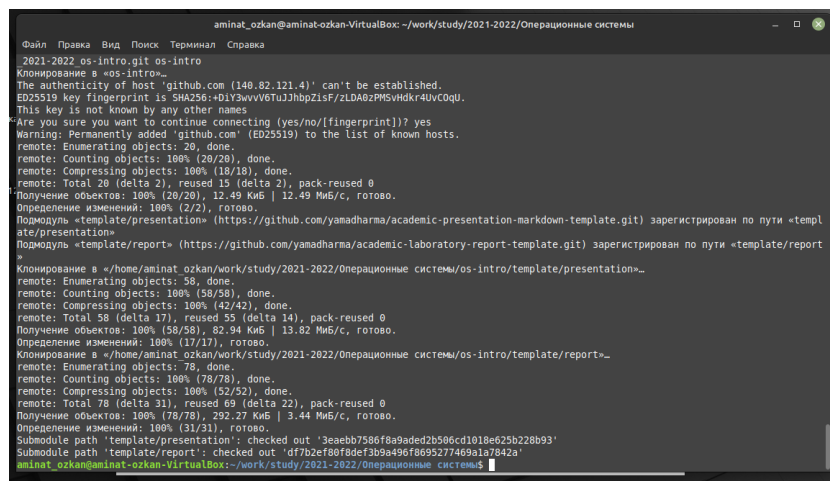
Настройка gh



```
aminat_ozkan@aminat-ozkan-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
72CRxqstfxu+RflicI0LICNyOX3VL5R7eTz0Lay0HKW/kmLzUgm1XTdluIM4SSZ  
005pt8FYfJ300L9UCDaxWMLCxCzNH2ZTWHRB3BcxIKM9nC400t9agQ+gZ5wZEY+  
acJ0ZBubgh8mZbV7531stVHTFD70wqRUKXI8afaky70N3IIEoX5f+1kVoPIAvZukm  
WBNDT03CmK3L7bCSD0rry4wIY0gR18aKKGz0N/WaaesLVB20VIR2QW0wtLL  
8812hoRtQM2099EtgplD1rRdxKfmmvduy0X3qJlT6pLz7YfZLd0LAE319215  
60t3M5j/YEGdn3099w+T8L1LxkUMBhuWY4uy0Y5pKfqY7I20fmdX1Yp9t8Myul  
kva5khZLW5e+PyqRF9G6DnkbVsuFLKwT2V56D7Y5Wc2oJYx01B245IYYEbu7Ci0  
Kh20ncuqxvq05gms+8acTQleMp8RG66X0Zzso/vnFCRpSjL0p253fRctqbWMM  
50jBnJ3pCnTtq/eE0X1QFaqQ3eLTH0DnZrcn186zAVdm+4ucP64Y3htrmpTKgi  
=C0uR  
-----END PGP PUBLIC KEY BLOCK-----  
aminat_ozkan@aminat-ozkan-VirtualBox:~$  
aminat_ozkan@aminat-ozkan-VirtualBox:~$  
aminat_ozkan@aminat-ozkan-VirtualBox:~$  
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global user.signingkey 5591473FC0B19F4  
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global commit.gpgsign true  
aminat_ozkan@aminat-ozkan-VirtualBox:~$ git config --global gpg.program $(which gpg2)  
aminat_ozkan@aminat-ozkan-VirtualBox:~$ 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/aminat_ozkan/.ssh/id_rsa.pub  
? Title for your SSH key:   
? How would you like to authenticate GitHub CLI? Login with a web browser  
First copy your one-time code: AB1C-F98A  
Press Enter to open github.com in your browser...  
2022-08-27T07:19:09Z ERROR viaduct::backend::ffi Missing HTTP status  
2022-08-27T07:19:09Z ERROR viaduct::backend::ffi Missing HTTP status  
Authentication complete.  
- gh config set -h github.com git_protocol ssh  
Configured git protocol  
Uploaded the SSH key to your GitHub account: /home/aminat_ozkan/.ssh/id_rsa.pub  
Logged in as aminatozkanrudn  
aminat_ozkan@aminat-ozkan-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация



```
aminat_ozkan@aminat-ozkan-VirtualBox: ~/work/study/2021-2022/Операционные системы  
Файл Правка Вид Поиск Терминал Справка  
2021-2022 os-Intro.git os-Intro  
Клонирование в «os-Intro»..  
The authenticity of host 'github.com (140.82.121.4)' can't be established.  
ED25519 key fingerprint is SHA256:0D1Y3wvV6TuJhbpZisF/zLDA0zPMSvHdk4UvC0Qu.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.  
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 KiB | 12.49 MiB/c, готово.  
Определение изменений: 100% (2/2), готово.  
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «templ  
ate/presentation»  
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report  
»  
Клонирование в «/home/aminat_ozkan/work/study/2021-2022/Операционные системы/os-Intro/template/presentation»..  
remote: Enumerating objects: 58, done.  
remote: Counting objects: 100% (58/58), done.  
remote: Compressing objects: 100% (42/42), done.  
remote: Total 58 (delta 17), reused 55 (delta 14), pack-reused 0  
Получение объектов: 100% (58/58), 82.94 KiB | 13.82 MiB/c, готово.  
Определение изменений: 100% (17/17), готово.  
Клонирование в «/home/aminat_ozkan/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 KiB | 3.44 MiB/c, готово.  
Определение изменений: 100% (21/21), готово.  
Submodule path 'template/presentation': checked out '3eabb750f8a9aded2b506cd1018e625b228b93'  
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'  
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений


```
aminat_ozkan@aminat-ozkan-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ rm package.json
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ make COURSE=os-intro
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git add .
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 6341185] feat(main): make course structure
16 files changed, 1580 insertions(+), 14 deletions(-)
create mode 100644 labs/lab(01..15)/presentation/Makefile
create mode 100644 labs/lab(01..15)/presentation/presentation.md
create mode 100644 labs/lab(01..15)/report/Makefile
create mode 100644 labs/lab(01..15)/report/bib/cite.bib
create mode 100644 labs/lab(01..15)/report/image/placemg_800_600_tech.jpg
create mode 100644 labs/lab(01..15)/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab(01..15)/report/report.md
delete mode 100644 package.json
create mode 100644 project-personal/stage(1..6)/presentation/Makefile
create mode 100644 project-personal/stage(1..6)/presentation/presentation.md
create mode 100644 project-personal/stage(1..6)/report/Makefile
create mode 100644 project-personal/stage(1..6)/report/bib/cite.bib
create mode 100644 project-personal/stage(1..6)/report/image/placemg_800_600_tech.jpg
create mode 100644 project-personal/stage(1..6)/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage(1..6)/report/report.md
create mode 100644 structure
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (19/19), 266.47 Кб | 2.07 Мб/с, готово.
Всего 19 (изменений: 2), повторно использовано 0 (изменений: 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:aminat_ozkanrudn/study_2021-2022_os-intro.git
c35e0de..6341185 master -> master
aminat_ozkan@aminat-ozkan-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
```

Figure 2.10: Первый коммит

3 Вывод

Мы приобрели практические навыки работы с сервисом github.

4 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 - пространство на накопителе где расположен репозиторий
 - commit - сохранение состояния хранилища
 - история - список изменений хранилища (коммитов)
 - рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

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

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- git config - установка параметров
- git status - полный список изменений файлов, ожидающих коммита
- git add . - сделать все измененные файлы готовыми для коммита.
- git commit -m "[descriptive message]" - записать изменения с заданным сообщением.
- git branch - список всех локальных веток в текущей директории.
- git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.
- git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.
- git push - запустить текущую ветку в удаленную ветку.
- git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

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

- git remote add [имя] [url] — добавляет удалённый репозиторий с заданным именем;
- git remote remove [имя] — удаляет удалённый репозиторий с заданным именем;
- git remote rename [старое имя] [новое имя] — переименовывает удалённый репозиторий;
- git remote set-url [имя] [url] — присваивает репозиторию с именем новый адрес;

- `git remote show [имя]` — показывает информацию о репозитории.

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

Ветвление — это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

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

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить:

Список литературы

1. Лекция Системы контроля версий
2. GitHub для начинающих