

Отчёт по лабораторной работе №2

Операционные системы

Трусова Алина Александровна

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	9
Выводы	14
Список литературы	15

Список иллюстраций

1	Установка git и gh	9
2	Настройка git	9
3	Ключ rsa	10
4	Ключ pgr	10
5	Ключ pgr	10
6	Ключ pgr	10
7	Добавление ключа на github	11
8	Настройка подписей коммитов git	11
9	Авторизация	12
10	Шаблон репозитория	12
11	Настройка каталога курса	13
12	Коммит	13
13	Коммит	13

Список таблиц

Цель работы

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

Задание

1. Установка программного обеспечения
2. Базовая настройка git
3. Создание ключей ssh
4. Создание ключей pgr
5. Добавление pgr ключа в github
6. Настройка автоматических подписей коммитов git
7. Настройка gh
8. Создание шаблона для рабочего пространства
9. Настройка каталога курса
10. Контрольные вопросы

Теоретическое введение

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.

##Примеры использования git

1. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.
2. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

Выполнение лабораторной работы

##1. Установка программного обеспечения

Установила git и gh (рис. [-@fig:001]).

```
aatrusova@vbox:~$ sudo dnf -y install git
[sudo] пароль для aatrusova:
Fedora 40 - x86_64 - Updates                24 kB/s | 23 kB    00:00
Последняя проверка окончания срока действия метаданных: 0:00:01 назад, Вт 04 мар
2025 12:05:34.
Пакет git-2.48.1-1.fc40.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
aatrusova@vbox:~$ sudo dnf -y install gh
Последняя проверка окончания срока действия метаданных: 0:00:19 назад, Вт 04 мар
2025 12:54:14
```

Рис. 1: Установка git и gh

##2. Базовая настройка git

Задала имя и email владельца репозитория, настроила utf-8 в выводе сообщений git, задала имя начальной ветки и назвала её master, добавила параметры autocrlf и safecrlf (рис. [-@fig:002]).

```
aatrusova@vbox:~$ git config --global user.name "Alina Trusova"
aatrusova@vbox:~$ git config --global user.email "ainatrusova18@gmail.com"
aatrusova@vbox:~$ git config --global core.quotepath false
aatrusova@vbox:~$ git config --global init.defaultBranch master
aatrusova@vbox:~$ git config --global core.autocrlf input
aatrusova@vbox:~$ git config --global core.safecrlf warn
aatrusova@vbox:~$
```

Рис. 2: Настройка git

##3. Создание ключа ssh

Создала ключ по алгоритму rsa с ключом размером 4096 бит (рис. [-@fig:003]).

```

aatrusova@vbox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aatrusova/.ssh/id_rsa):
Created directory '/home/aatrusova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aatrusova/.ssh/id_rsa
Your public key has been saved in /home/aatrusova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:WsjeiAckL8HhhcohVFiQSjNzdnQDC4HSBiesxIu4ybM aatrusova@vbox
The key's randomart image is:

```

Рис. 3: Ключ rsa

##4. Создание ключа pgr

Создала ключ с нужными параметрами (рис. [-@fig:004]).

```

aatrusova@vbox:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.4; Copyright (C) 2024 g10 Code GmbH
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/aatrusova/.gnupg'
Выберите тип ключа:
(1) RSA and RSA
(2) DSA and Elgamal

```

Рис. 4: Ключ pgr

##5. Добавление pgr ключа в github

Вывела список ключей и скопировала отпечаток приватного ключа (рис. [-@fig:005]).

```

aatrusova@vbox:~$ 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: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]

```

Рис. 5: Ключ pgr

Вывела сгенерированный ключ и скопировала его в буфер обмена (рис. [-@fig:006]).

```

aatrusova@vbox:~$ gpg --armor --export D47200EC613B37A5
-----BEGIN PGP PUBLIC KEY BLOCK-----

```

Рис. 6: Ключ pgr

Добавила новый ключ gpg на github (рис. [-@fig:007]).

Add new GPG key

Title

aatrusova

Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Sf3nb9J5f/3Njd9QqpDMDHcIN+x9Vg4VHluLLTBr/
O9yJXjxcWdUnCSiWh1YuSRR
PHUzwYHRvxp1qMFM+xd+OxhzwC6EqkvyLaZcHMSmmd45
PvXZagRCzjGjkZxx6Y69
W0WRxBdGVsVxAnS8bpWuUZGX2YqPoVwyUvrhkcqH5zd9y
KM1v6z5tl65kDEN0sU=
=HKeN
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

Рис. 7: Добавление ключа на github

##6. Настройка автоматических подписей коммитов git

Указала git применять введённый email для подписи коммитов (рис. [-@fig:008]).

```
aatrusova@vbox:~$ git config --global user.signingkey D47200EC613B37A5
aatrusova@vbox:~$ git config --global commit.gpgsign true
aatrusova@vbox:~$ git config --global gpg.program $(which gpg2)
aatrusova@vbox:~$
```

Рис. 8: Настройка подписей коммитов git

##7. Настройка gh

Авторизовалась на гитхабе (рис. [-@fig:009]).

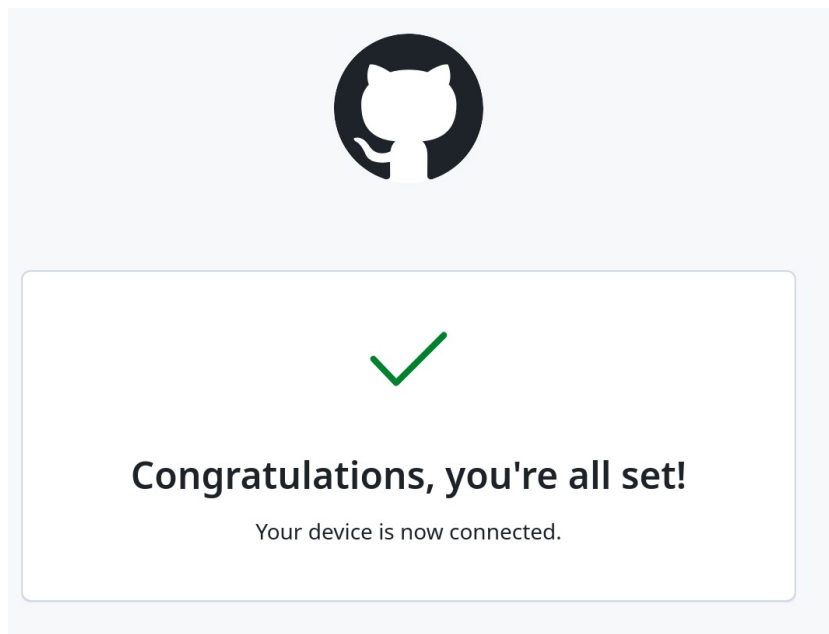


Рис. 9: Авторизация

##8. Создание шаблона рабочего пространства

Создала рабочее пространство по шаблону (рис. [-@fig:010]).

```
aatrusova@vbox: ~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
aatrusova@vbox: ~$ cd ~/work/study/2024-2025/"Операционные системы"
aatrusova@vbox: ~/work/study/2024-2025/Операционные системы$ gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository alas-aline/study_2024-2025_os-intro on GitHub
https://github.com/alas-aline/study_2024-2025_os-intro
aatrusova@vbox: ~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:alas-aline/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
```

Рис. 10: Шаблон репозитория

##9. Настройка каталога курса

Перешла в каталог курса, удалила лишние файлы и создала новые (рис. [-@fig:011]).

```

aatrusova@vbox:~/work/study/2024-2025/Операционные системы$ cd os-intro
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.
json
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-int
ro > COURSE
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submodules
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$

```

Рис. 11: Настройка каталога курса

Прописала коммит (рис. [-@fig:012]).

```

aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 39, готово.
Подсчет объектов: 100% (39/39), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.28 КиБ | 2.65 МиБ/с, готово.
Total 38 (delta 4), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:alas-aline/study_2024-2025_os-intro.git
   735b3e9..0a1c580  master -> master
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$

```

Рис. 12: Коммит

Отправила изменения на репозиторий (рис. [-@fig:013]).

```

aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-int
ro > COURSE
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ make prepar
e
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git commit
-am ;feat(main): make course structure'
bash: синтаксическая ошибка рядом с неожиданным маркером «main»
aatrusova@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git commit
-am 'feat(main): make course structure'
[master 0a1c580] feat(main): make course structure
 403 files changed, 98412 insertions(+)
 create mode 100644 labs/README.md
 create mode 100644 labs/README.ru.md
 create mode 100644 labs/lab01/presentation/.projectile
 create mode 100644 labs/lab01/presentation/.texlabroot
 create mode 100644 labs/lab01/presentation/Makefile

```

Рис. 13: Коммит

Выводы

Я изучила идеологию и применение средств контроля версий и освоила умения по работе с git.

Список литературы

1. Архитектура компьютеров и операционные системы. Раздел “Операционные системы”. “Лабораторная работа №2” <https://esystem.rudn.ru/mod/page/view.php?id=12>