

Отчет по лабораторной работе №2

Дисциплина архитектура компьютера

Ахатов Эмиль Эрнстович

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
5	Выводы	14

Список иллюстраций

4.1	установка	8
4.2	установка	8
4.3	базовая настройка git	9
4.4	создание ключей	10
4.5	окно подтверждения	10
4.6	ключ	11
4.7	копирование ключа	11
4.8	добавление ключа	11
4.9	авторизация	12
4.10	создание репозитория	12
4.11	отправка на сервер	13

1 Цель работы

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

2 Задание

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

3 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.

4 Выполнение лабораторной работы

Устанавливаю git: dnf install git

```
root@fedora:~  
emilefedora:~$ dnf install git  
Ошибка: 3-ю команду нужно запускать с привилегиями суперпользователя (на большинстве систем - под именем пользователя root).  
emilefedora:~$ sudo -i  
[sudo] пароль для emil:  
root@fedora:~# dnf install git  
Copr repo for PyCharm owned by phracek 5.3 kB/s | 2.1 kB 00:00  
Copr repo for PyCharm owned by phracek 2.8 kB/s | 2.4 kB 00:00  
Fedora 40 - x86_64 23 kB/s | 28 kB 00:01  
Fedora 40 openh264 (From Cisco) - x86_64 1.6 kB/s | 989 B 00:00  
Fedora 40 - x86_64 - Updates 28 kB/s | 25 kB 00:00  
Fedora 40 - x86_64 - Updates 5.3 MB/s | 8.9 MB 00:01  
google-chrome 6.4 MB/s | 1.3 kB 00:00  
google-chrome 4.8 kB/s | 1.9 kB 00:00  
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Driver 19 kB/s | 9.3 kB 00:00  
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Driver 6.1 kB/s | 5.2 kB 00:00  
RPM Fusion for Fedora 40 - Nonfree - Steam 21 kB/s | 8.9 kB 00:00  
RPM Fusion for Fedora 40 - Nonfree - Steam 2.1 kB/s | 1.5 kB 00:00  
Пакет git-2.47.0-1.fc40.x86_64 уже установлен.  
Зависимости разрешены.  
нет действий для выполнения.  
выполнено!  
root@fedora:~#
```

Рис. 4.1: установка

Устанавливаю gh: dnf install gh

```
root@fedora:~# dnf install gh  
Последняя проверка окончания срока действия метаданных: 0:00:52 назад, Вс 02 мар 2025 16:07:42.  
зависимости разрешены.  
=====
```

Пакет	Архитектура	Версия	Репозиторий	Размер
Установка: gh	x86_64	2.65.0-1.fc40	updates	11 М

```
=====
```

Результат транзакции

Установка 1 Пакет

Объем загрузки: 11 М
Объем изменений: 50 М
Продолжить? [д/н]: y
Загрузка пакетов:
gh-2.65.0-1.fc40.x86_64.rpm 22 MB/s | 11 MB 00:00

Общий размер	11 MB/s 11 MB	00:00
Проверка транзакции		
Проверка транзакции успешно завершена.		
Готов проверка транзакции		
Тест транзакции проведен успешно.		
Выполнение транзакции		
Подготовка :		1/1
Установка : gh-2.65.0-1.fc40.x86_64		1/1
Запуск скрипглета: gh-2.65.0-1.fc40.x86_64		1/1

```
Установлено:  
gh-2.65.0-1.fc40.x86_64  
выполнено!
```

Рис. 4.2: установка

Базовая настройка git

Задаю имя и email для моего репозитория:

```
git config --global user.name "Emil Ahatov" git config --global user.email  
"ahatovemil48@gmail"
```

Настраиваю utf-8 в выводе сообщений git:

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

Настраиваю верификацию и подписание коммитов git

Задаю имя начальной ветки (будем называть её master):

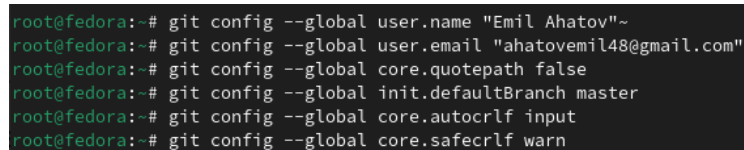
```
git config --global init.defaultBranch master
```

Параметр autocrlf:

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

Параметр safecrlf:

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

A screenshot of a terminal window showing a series of git configuration commands being executed. The prompt is 'root@fedora: #'. The commands are: 'git config --global user.name "Emil Ahatov"', 'git config --global user.email "ahatovemil48@gmail.com"', 'git config --global core.quotepath false', 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'.

```
root@fedora: # git config --global user.name "Emil Ahatov"  
root@fedora: # git config --global user.email "ahatovemil48@gmail.com"  
root@fedora: # git config --global core.quotepath false  
root@fedora: # git config --global init.defaultBranch master  
root@fedora: # git config --global core.autocrlf input  
root@fedora: # git config --global core.safecrlf warn
```

Рис. 4.3: базовая настройка git

Создаю ключи ssh по алгоритму rsa с ключём размером 4096 бит:

```
ssh-keygen -t rsa -b 4096
```

по алгоритму ed25519:

```
ssh-keygen -t ed25519
```

Создаю ключи pgp

Генерируем ключ gpg --full-generate-key

```

your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:yaynWYGUTEKRahKoi+xdL+Y6rYhNhUeEbb6bkP5IAK8 root@fedora
The key's randomart image is:
+---[RSA 4096]-----+
| .  +o. |
| o  ..++ . |
| o.  +.  + |
|+.oo..  + . |
|++o.o.. S |
|o+oo.. . . |
|E.+..ooo o |
|.=o+o+ * |
|..o+B.+ |
+-----[SHA256]-----+
root@fedora:~# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:lnVT8kHQn6EMsnGRYAkO/y5tQJIf8tfrskmilex0tXM root@fedora
The key's randomart image is:
+---[ED25519 256]---+
|      ...oo.=++ |
| . .  .+ + +.o |
| +      * = o.o |
| + +  + . + .. |
| * o S . |
| +o+ .. . |
| o== ..o E |
| o.=o. o |
| . o++ |
+-----[SHA256]-----+
root@fedora:~#

```

Рис. 4.4: создание ключей



Рис. 4.5: окно подтверждения

```
<n> = срок действия ключа - n дней
<nw> = срок действия ключа - n недель
<nm> = срок действия ключа - n месяцев
<ny> = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

gpg должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Ahatov Emil
Адрес электронной почты: ahatovemil48@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
"Ahatov Emil <ahatovemil48@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход?
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 (печать
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 (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /root/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/root/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/B3D469F27483AB5D8862A53077AEC3C990CC9885.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2025-03-02 [SC]
      B3D469F27483AB5D8862A53077AEC3C990CC9885
uid          Ahatov Emil <ahatovemil48@gmail.com>
sub   rsa4096 2025-03-02 [E]
```

Рис. 4.6: ключ

Выводим список ключей и копируем отпечаток приватного ключа. Скопировал сгенерированный PGP ключ в буфер обмена:

```
gpg --armor --export | xclip -sel clip
```

Перехожу в настройки GitHub, нажимаю на кнопку New GPG key и вставляю полученный ключ в поле ввода.

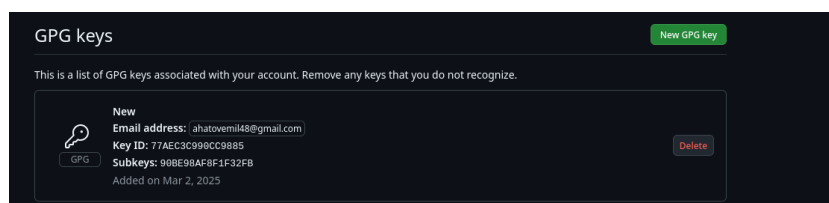


Рис. 4.7: копирование ключа

```
root@fedora:~# git config --global user.signingkey 77AEC3C990CC9885
root@fedora:~# git config --global commit.gpgsign true
root@fedora:~# git config --global gpg.program $(which gpg2)
```

Рис. 4.8: добавление ключа

Авторизуюсь через браузер

```

emil@fedora: ~/work/study/2024-2025/Операционные системы$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/emil/.ssh/id_ed25519.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ SSH key already existed on your GitHub account: /home/emil/.ssh/id_ed25519.pub
✓ Logged in as AhatovEmil

```

Рис. 4.9: авторизация

создание репозитория курса на основе шаблона прописываю комнды для создания репозитория:

```

mkdir -p ~/work/study/2022-2023/“Операционные системы” cd ~/work/study/2022-2023/“Операционные системы” gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public git clone --recursive git@github.com:/study_2024-2025_os-intro.git os-intro

```

```

emil@fedora: ~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:AhatovEmil/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (36/36), 19.37 Киб | 6.46 Мб/с, готово.
Определение изменений: 100% (1/1), 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/emil/work/study/2024-2025/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Получение объектов: 100% (111/111), 102.17 Киб | 1.11 Мб/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/emil/work/study/2024-2025/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Получение объектов: 100% (142/142), 341.09 Киб | 2.16 Мб/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fcaid4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
emil@fedora: ~/work/study/2024-2025/Операционные системы$

```

Рис. 4.10: создание репозитория

Настройка каталога курса

Перехожу в каталог курса:

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

Удаляю лишние файлы:

```
rm package.json
```

Создаю необходимые каталоги:

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

Отправляю файлы на сервер:

git add . git commit -am 'feat(main): make course structure' git push

```
emil@fedora: ~/work/study/2024-2025/Операционные системы$ cd ~/work/study/2024-2025/"Операционные системы"/os-intro
emil@fedora: ~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json
emil@fedora: ~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
emil@fedora: ~/work/study/2024-2025/Операционные системы/os-intro$ make
Usage:
  make <target>

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

emil@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ git add.
git: «add.» не является командой git. Смотрите «git --help».

Самые похожие команды:
  add

emil@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
emil@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 39de9ed] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
emil@fedora:~/work/study/2024-2025/Операционные системы/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.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:AhatovEmil/study_2024-2025_os-intro.git
9b23b5d..39de9ed master -> master
```

Рис. 4.11: отправка на сервер

5 Выводы

Я изучил идеологию и применение средств контроля версий. Освоил умения по работе с git.