

Отчёт по лабораторной работе №1

Управление версиями

Арам Грачьяевич Саргсян

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	11
	Список литературы	12

Список иллюстраций

4.1	Репозиторий на github	9
4.2	Работа с git в консоли	9
4.3	ssh ключ	10
4.4	pgr ключ	10

Список таблиц

1 Цель работы

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

2 Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий по предмету.

3 Теоретическое введение

Системы контроля версий 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 при необходимости.

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 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 дельтакомпрессию — 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

1. Я создал репозиторий для работы с курсом в своем github (рис. 4.1).

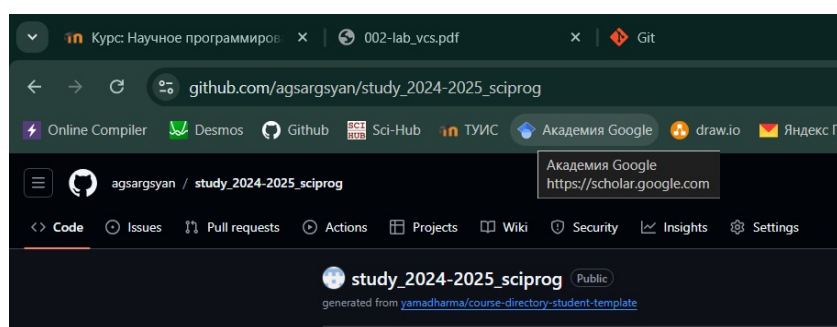


Рис. 4.1: Репозиторий на github

2. Установил рабочее пространство для работы с git (рис. 4.2).

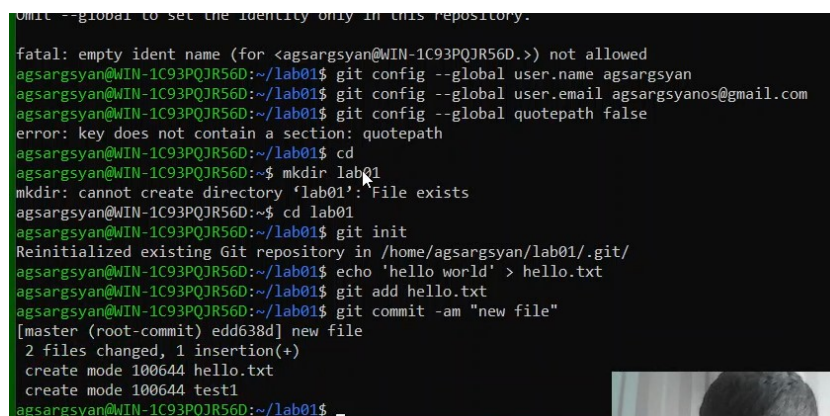


Рис. 4.2: Работа с git в консоли

3. Создал ssh и pgr ключи (рис. 4.3, 4.4).

```

Enter file in which to save the key (/home/agsargsyan/.ssh/id_rsa): C
agsargsyan@WIN-1C93PQJR56D:~/lab01$ ssh-keygen -C "Aram Sargsyan <agsargsyanos@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/agsargsyan/.ssh/id_rsa):
Created directory '/home/agsargsyan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/agsargsyan/.ssh/id_rsa
Your public key has been saved in /home/agsargsyan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XyAyqtNlRgOSGtD0anv35/Ie7MrCU1LxoBFEDfy8W78 Aram Sargsyan <agsargsyanos@gmail.com>
The key's randomart image is:
+--[RSA 3072]-----+
|o=..Bo|
|.=..B|
|. =..+B.|
|oo.o*+|.
|o.ooS..|
|. + + o..o|
|. + + o..o|
|. +.o..E|
|ooB=|
+-----[SHA256]-----+
agsargsyan@WIN-1C93PQJR56D:~/lab01$

```

Рис. 4.3: ssh ключ

```

agsargsyan@WIN-1C93PQJR56D:~/lab01$ gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/agsargsyan/.gnupg' created
gpg: keybox '/home/agsargsyan/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Aram
Name must be at least 5 characters long
Real name: agsargsyan
Email address: agsargsyanos@gmail.com
Comment: pgp
You selected this USER-ID:
    "agsargsyan (pgp) <agsargsyanos@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?

```

Рис. 4.4: pgr ключ

5 Выводы

Я установил рабочее пространство и получил навыки для работы с git и markdown.

Список литературы

1. Система контроля версиями
2. Репозиторий курса