

Лабораторная работа №2. Система контроля версий Git

Павленко Сергей

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Задание для самостоятельной работы	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	1	8
3.2	17	10
3.3	18	11
3.4	19	11
4.1	20	12

Список таблиц

1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

2 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. Демидова А. В. 14 Архитектура ЭВМ 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

1. Создайте учётную запись на github.com

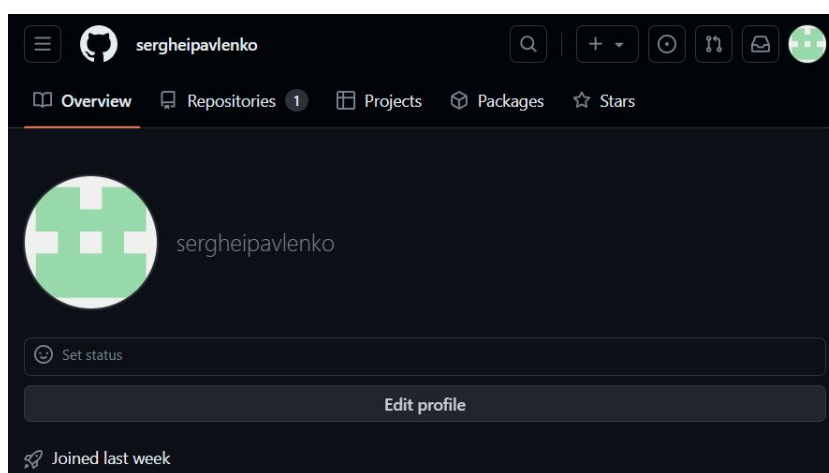


Рис. 3.1: 1

2. Базовая настройка git

Сделаем предварительную конфигурацию git.

```
[spavlenko@localhost ~]$ git config --global user.name "Serghei Pavlenko"
[spavlenko@localhost ~]$ git config --global user.email pavlenko.serghei.a
[spavlenko@localhost ~]$
```

Настроим utf-8 в выводе сообщений git: [spavlenko@localhost ~]\$ git config --global core.quotePath false

Зададим имя начальной ветки (будем называть её master): [spavlenko@localhost ~]\$ git config --global

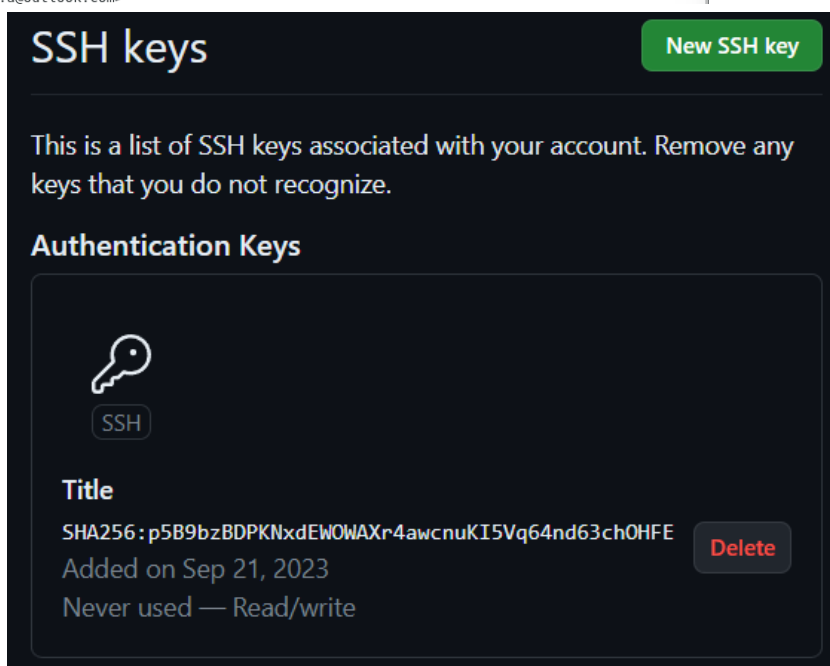
Параметр autocrlf: [spavlenko@localhost ~]\$ git config --global core.autocrlf input

Параметр safecrlf: [spavlenko@localhost ~]\$ git config --global core.safecrlf warn

Создание SSH ключа Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и


```
[spavlenko@localhost ~]$ ssh-keygen -C "Sergei Pavlenko <pavlenko.sergei.a@outlook.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/spavlenko/.ssh/id_rsa):
Created directory '/home/spavlenko/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/spavlenko/.ssh/id_rsa.
Your public key has been saved in /home/spavlenko/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:p5B9bzBDPKNxdEWOWAXr4awcnuKI5Vq64nd63ch0HFE Sergei Pavlenko <pavlenko.sergei.a@outlook.com>
The key's randomart image is:
+---[RSA 2048]---+
|      E  +=+      |
|      +  +  +     |
|      o B + .     |
|      o * * .     |
|      o S B +     |
|      o B O       |
|      oo=O= o     |
|      . .o++...   |
|      ..o**o.o    |
+---[SHA256]-----+
[spavlenko@localhost ~]$ ^C
[spavlenko@localhost ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
bash: xclip: команда не найдена...
[spavlenko@localhost ~]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaClyc2EAAAADAQABAAQDDNN2ZfXRSADXLjcs6oLuS3dU1b1P1eg20fefuf8dSUC/Nhpoi/0RgpxRbz/4m
KP1AtvjD8MeE0UD8QYH2HpurU8CSlptSL53P5j17gHe0W8vqP8S2AvcLD7wI4DTDYV8kZYBwPvdMh0+DUY/mqJLZ+z+mz0u5Fxc9
qIId+EzzTbBL0dPomyh53028XndwaD2yI3CfQ8mxs3j0gIq2PdFutH0uJNuUdbWdWYghc/Kd1hBXD+j/E6dnV0LYaKgljueo2mJT
Ggc00DLuPq5BBdPNKkCLJRar+Np1aImcxoeACWMoCFrATDeRkQKjGGYRSIZb11LabPha0GaAHFJZ0k0p Sergei Pavlenko <p
avlenko.sergei.a@outlook.com>
```

открытый):



Создали SSH ключ

Создание рабочего пространства и репозитория курса на основе шаблона

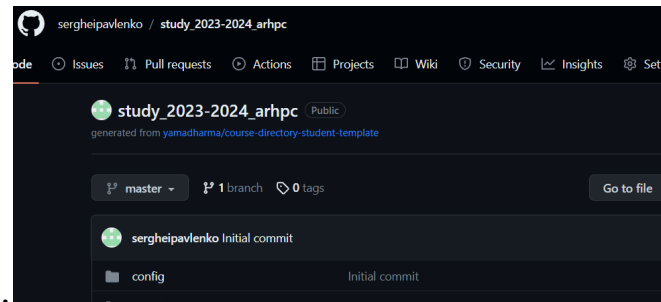
Откройте терминал и создайте каталог для предмета «Архитектура компьютера»:

```
[spavlenko@localhost ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
[spavlenko@localhost ~]$
```

Создали каталог для предмета «Архитектура компьютера»

Создание репозитория курса на основе шаблона Репозиторий на основе шаблона можно создать через web-интерфейс github. Перейдём на страницу репозитория с шаблоном курса <https://github.com/yamadharm/cour-se-directory->

student-template. Далее выберите Use this template.



Создали репозиторий по шаблону В открывшемся окне задали имя репозитория study_2023–2024_arhpc и создали репозиторий (from template).

Откроем терминал и перейдём в каталог курса:

```
[spavlenko@localhost ~]$ cd ~/work/study/2023-2024/"Архитектура к  
[spavlenko@localhost Архитектура компьютера]$
```

```
[spavlenko@localhost Архитектура компьютера]$ git clone --recursive git@github.com:sergheipavlenko/stu  
2023-2024_arhpc.git  
Cloning into 'study_2023-2024_arhpc'...  
The authenticity of host 'github.com (140.82.121.4)' can't be established.  
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeI0ttrVc98/R1BUFWu3/LiyKgUfQM.  
ECDSA key fingerprint is MD5:7b:99:81:1e:4c:91:a5:0d:5a:2e:2e:80:13:3f:24:ca.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'github.com,140.82.121.4' (ECDSA) to the list of known hosts.  
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  
Receiving objects: 100% (27/27), 16.94 KiB | 0 bytes/s, done.  
Resolving deltas: 100% (1/1), done.  
Submodule 'template/presentation' (https://github.com/yamadharm/academic-presentation-markdown-templa  
.git) registered for path 'template/presentation'  
Submodule 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) re  
stered for path 'template/report'  
Cloning into '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  
Unpacking objects: 100% (82/82), done.
```

Клонируем созданный репозиторий:

Настройка каталога курса Перейдём в каталог курса:

```
[spavlenko@localhost Архитектура компьютера]$ cd ~/work/study/2023-2024/Арх  
23-2024_arhpc/
```

Удалим лишние файлы:

```
[spavlenko@localhost study_2023-2024_arhpc]$ rm package.json
```

```
[spavlenko@localhost study_2023-2024_arhpc]$ echo arch-pc > COURSE  
[spavlenko@localhost study_2023-2024_arhpc]$ make  
[spavlenko@localhost study_2023-2024_arhpc]$
```

Создадим необходимые каталоги:

```
[spavlenko@localhost study_2023-2024_arhpc]$ git add .  
warning: You ran 'git add' with neither '-A (--all)' or '--ignore-removal',  
whose behaviour will change in Git 2.0 with respect to paths you removed.  
Paths like 'package.json' that are  
removed from your working tree are ignored with this version of Git.
```

Отправим файлы на сервер:

```
* 'git add --ignore-removal <paths>' which is the current default
```

```
[spavlenko@localhost study_2023-2024_arhpc]$ git commit -am 'feat(main): make course structure'  
[master fb7334a] feat(main): make course structure  
199 files changed, 54725 insertions(+), 14 deletions(-)  
create mode 100644 labs/README.md  
create mode 100644 labs/README.ru.md  
create mode 100644 labs/lab01/presentation/Makefile
```

Рис. 3.2: 17

```
[spavlenko@localhost study_2023-2024_arhpc]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 37, done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.14 KiB | 0 bytes/s, done.
Total 35 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
remote: To git@github.com:sergheipavlenko/study_2023-2024_arhpc.git
        41e4e13..fb7334a  master -> master
[spavlenko@localhost study_2023-2024_arhpc]$
```

Рис. 3.3: 18

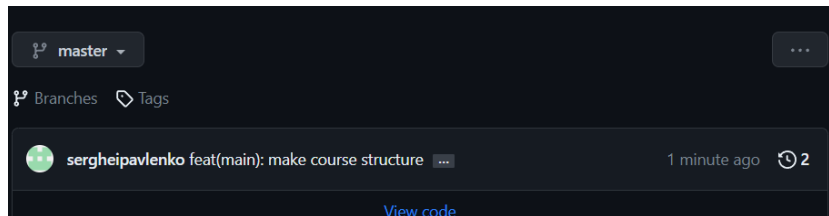


Рис. 3.4: 19

4 Задание для самостоятельной работы

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге ра

```
[spavlenko@localhost lab02]$ cd ~/work/study/2023-2024/Архитектура\ компьютера/s
tudy_2023-2024_arch-pc/labs/lab02/report/
[spavlenko@localhost report]$ git add Л02_Павленко_Сергей_отчет.docx
[spavlenko@localhost report]$ git commit -am 'feat(main): make course structure'
[master 6c75304] feat(main): make course structure
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename labs/lab02/{ => report}/Л02_Павленко_Сергей_отчет.docx (100%)
[spavlenko@localhost report]$ git commit -am 'feat(main): make course structure'
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   ../../lab01/lab1.docx
nothing added to commit but untracked files present (use "git add" to track)
[spavlenko@localhost report]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)
Counting objects: 9, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 536 bytes | 0 bytes/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote: To git@github.com:sergheipavlenko/study_2023-2024_arhpc.git
       c53ab55..6c75304 master -> master
[spavlenko@localhost report]$
```

Рис. 4.1: 20

2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства и загрузите

```
[spavlenko@localhost ~]$ cd ~/work/study/2023-2024/Архитектура\ компьютера/study
2023-2024_arch-pc/labs/lab01/report/
[spavlenko@localhost report]$ git add lab1.docx
[spavlenko@localhost report]$ git commit -am 'feat(main): make course structure'
[master e390c5b] feat(main): make course structure
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/lab1.docx
[spavlenko@localhost report]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 10, done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 582.12 KiB | 0 bytes/s, done.
Total 6 (delta 3), reused 0 (delta 0)
```

файлы на github. ■

Отчёты по лабораторным работам, находящийся по адресу ~/labs/lab02/report и ~/labs/lab01/report, отправили в репозиторий и сохранили их в соответствующим им директориям

5 Выводы

В самостоятельной работе мы выполнили работу по сохранению отчетов в репозиторий `github.com`, с помощью соответствующих команд в Terminal

Список литературы