

Лабораторная работа №2. Система контроля версий Git

Кривобоков Юрий Дмитриевич НКАбд-03-24

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

Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений):

```
git checkout master  
git pull  
git checkout -b имя_ветки
```

Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории. Для этого необходимо проверить, какие файлы изменились к текущему моменту:

```
git status
```

и при необходимости удаляем лишние файлы, которые не хотим отправлять в центральный репозиторий.
Затем полезно просмотреть текст изменений на предмет соответствия правилам ведения чистых коммитов:

```
git diff
```

Если какие-либо файлы не должны попасть в коммит, то помечаем только те файлы, изменения которых нужно сохранить. Для этого используем команды добавления и/или удаления с нужными опциями:

```
git add имена_файлов
```

```
git rm имена_файлов
```

Если нужно сохранить все изменения в текущем каталоге, то используем:

```
git add .
```

Затем сохраняем изменения, поясняя, что было сделано:

```
git commit -am "Some commit message"
```

и отправляем в центральный репозиторий:

git push origin имя_ветки или git push

2. Базовая настройка git

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория:

```
git config --global user.name "<Name Surname>" git config --global user.email "work@mail"
```

Настроим utf-8 в выводе сообщений git:

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

Зададим имя начальной ветки (будем называть её master):

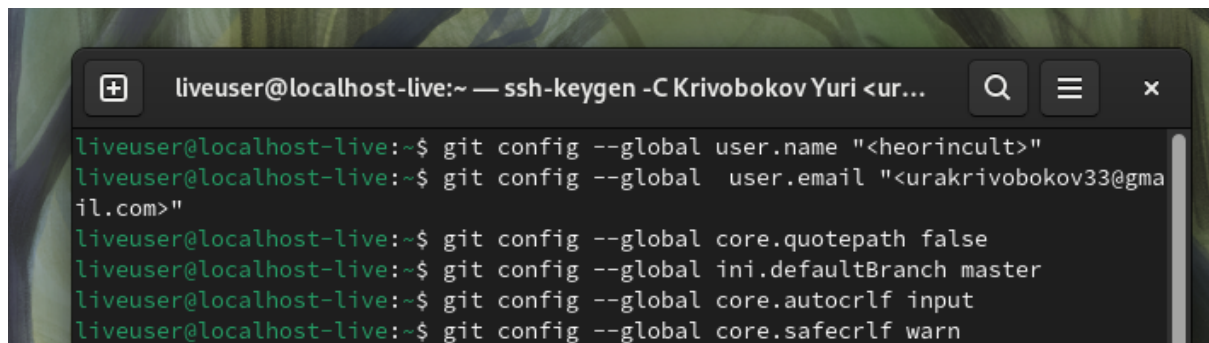
```
git config --global init.defaultBranch master
```

Параметр autocrlf:

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

Параметр safecrlf:

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

A screenshot of a terminal window with a dark background. The window title is "liveuser@localhost-live:~ — ssh-keygen -C Krivobokov Yuri <ur...". The terminal shows a series of git configuration commands being executed. The prompt is "liveuser@localhost-live:~\$". The commands and their outputs are: 1. "git config --global user.name "<heorincult>"", 2. "git config --global user.email "<urakrivobokov33@gmail.com>"", 3. "git config --global core.quotePath false", 4. "git config --global init.defaultBranch master", 5. "git config --global core.autocrlf input", and 6. "git config --global core.safecrlf warn".

```
liveuser@localhost-live:~$ git config --global user.name "<heorincult>"
liveuser@localhost-live:~$ git config --global user.email "<urakrivobokov33@gmail.com>"
liveuser@localhost-live:~$ git config --global core.quotePath false
liveuser@localhost-live:~$ git config --global init.defaultBranch master
liveuser@localhost-live:~$ git config --global core.autocrlf input
liveuser@localhost-live:~$ git config --global core.safecrlf warn
```

3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):

```
ssh-keygen -C "Имя Фамилия "
```

Ключи сохраняются в каталоге ~/.ssh/. Далее необходимо загрузить сгенерированный открытый ключ. Для этого зайти на сайт <http://github.org/> под своей учётной записью и перейти в меню Setting . После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key . Скопировав из локальной консоли ключ в буфер обмена

```
cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title).

```
liveuser@localhost-live:~$ ssh-keygen -C "Krivobokov Yuri <urakrivobokov33@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:CH2kB4Z/dcmXlIBZLHiVPxB2ZU84HNVRkLu3GE2PbYQ Krivobokov Yuri <urakrivobokov33@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|      .o .o++B=**|
|      .o ++ =.*.B=o|
|      ..o o+ o *E+o|
|      ..+. . B=.|
|      ..S   ooo=|
|              oo.|
|              . .|
|              |
+----[SHA256]-----+
liveuser@localhost-live:~$ cat ~/.ssh/id_rsa.pub
cat: /home/liveuser/.ssh/id_rsa.pub: No such file or directory
liveuser@localhost-live:~$ ^C
liveuser@localhost-live:~$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBZhrfokyBvG+TUGFSJj+QKvXuEhr0klSW2peAcL6Ee9 Krivobokov Yuri <urakrivobokov33@gmail.com>
```

heroincult (heroincult)
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

lab2
SHA256:2nYkks+g14k/w3bB4u/L+BiI+1b4andk6bJXSC3gV5U
Added on Sep 28, 2024
Never used — Read/write
Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

clear

4. Сознание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Рабочее пространство по предмету располагается в следующей иерархии:

Архитектура ЭВМ

~/work/study/

└─ <учебный год>/

└─ <название предмета>/

└─ <код предмета>/

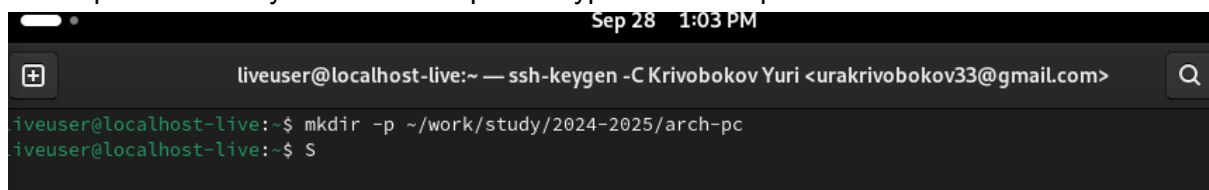
Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) структура каталогов примет следующий вид:

```
~/work/study/  
└── 2023–2024/  
    └── Архитектура компьютера/  
        └── arch-pc/  
            └── labs/  
                ├── lab01/  
                ├── lab02/  
                └── lab03/  
...
```

- Каталог для лабораторных работ имеет вид labs.
- Каталоги для лабораторных работ имеют вид lab<номер>, например: lab01, lab02 и т.д.

Название проекта на хостинге git имеет вид: study_<учебный год>_<код предмета>
Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) название проекта примет следующий вид: study_2023–2024_arch-pc
Откройте терминал и создайте каталог для предмета «Архитектура компьютера»:

```
mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
```



5. Создание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github.

Перейдите на страницу репозитория с шаблоном курса

<https://github.com/yamadharma/course-directory-student-template>

. Далее выберите Use this template.

В открывшемся окне задайте имя репозитория (Repository name)

study_2023–2024_arhpc и создайте репозиторий (кнопка Create repository from template).

Откройте терминал и перейдите в каталог курса:

```
cd ~/work/study/2023–2024/"Архитектура компьютера"
```

клонировать созданный репозиторий: `git clone --recursive`

```
git@github.com:/study_2023–2024_arh-pc.git ↩ arch-pc
```

Ссылку для клонирования можно скопировать на странице созданного репозитория Code -> SSH:

```
liveuser@localhost-live:~$ cd ~/work/study/2024-2025/arch-pc
liveuser@localhost-live:~/work/study/2024-2025/arch-pc$ git clone --recursive git@github.com:heroincult/study_2024-2025_arch-pc.git
Cloning into 'study_2024-2025_arch-pc'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (33/33), 18.81 KiB | 1.71 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/liveuser/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc/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)
Receiving objects: 100% (111/111), 102.17 KiB | 634.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/liveuser/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc/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)
Receiving objects: 100% (142/142), 341.09 KiB | 1.01 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
liveuser@localhost-live:~/work/study/2024-2025/arch-pc$
```

6. Настройка каталога курса

Перейдите в каталог курса:

```
cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
```

Удалите лишние файлы:

```
rm package.json
```

Создайте необходимые каталоги:

```
echo arch-pc > COURSE
```

```
make
```

Отправьте файлы на сервер:

```
git add .
```

```
git commit -am 'feat(main): make course structure'
```

```
git push
```

Проверьте правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.

```
Sep 28 1:38 PM
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc

liveuser@localhost-live:~$ cd ~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ rm package.json
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ echo arch-pc > COURSE
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ make
ash: make: command not found...
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ echo study_2024-2025_arch-pc > COURSE
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ make
ash: make: command not found...
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ git add .
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ git commit -am 'feat(main): make cours
structure'
[master 0d94d8b] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 151.00 KiB/s, done.
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:heroincult/study_2024-2025_arch-pc.git
   e3dc7cf..0d94d8b master -> master
liveuser@localhost-live:~/work/study/2024-2025/arch-pc/study_2024-2025_arch-pc$
```

Задания для самостоятельной работы

Не смог справиться с заданиями, т.к. они не точно сформулированы. Не понял какого формата необходимо сделать отчет и что он должен содержать.

Вывод: По ходу выполнения лабораторной работы, я научился работать с github. А также более уверенно пользоваться ОС Linux.