

Лабораторная работа №2

Дисциплина: Архитектура компьютера

Баранов Георгий Павлович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Техническое обеспечение	9
4.2	Базовая настройка Git	9
4.3	Создание рабочего пространства и репозитория курса на основе шаблона	10
4.4	Создание репозитория курса на основе шаблона	11
4.5	Задания для самостоятельной работы	12
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Предварительная конфигурация Git	9
4.2	Настройка параметров Git	9
4.3	Создание пары ssh ключей	10
4.4	Установка пары ключей	10
4.5	Создание рабочей директории	10
4.6	Создание репозитория на GitHub	11
4.7	Копирование репозитория на рабочий компьютер	11
4.8	Настройка каталога	12
4.9	Заключительные действия	12

Список таблиц

1 Цель работы

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

2 Задание

На основе методических указаний провести работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить GitHub.

3 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 зави-

симости 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.

4 Выполнение лабораторной работы

4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

4.2 Базовая настройка Git

Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды на (рис. -fig. 4.1).

```
baranov_georgy_1132246760@vbox:~$ git config --global user.email "1132246760@pfur.ru"
baranov_georgy_1132246760@vbox:~$ git config --global user.name "Георгий"
baranov_georgy_1132246760@vbox:~$
```

Рис. 4.1: Предварительная конфигурация Git

Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf. (рис. -fig. 4.2)

```
baranov_georgy_1132246760@vbox:~$ git config --global user.email "1132246760@pfur.ru"
baranov_georgy_1132246760@vbox:~$ git config --global user.name "Георгий"
baranov_georgy_1132246760@vbox:~$ git config --global core.quotepath false
baranov_georgy_1132246760@vbox:~$ git config --global init.defaultBranch master
baranov_georgy_1132246760@vbox:~$ git config --global core.autocrlf input
baranov_georgy_1132246760@vbox:~$ git config --global core.safecrlf warn
baranov_georgy_1132246760@vbox:~$
```

Рис. 4.2: Настройка параметров Git

Далее создаю пару ssh ключей для интеграции с платформой GitHub. (рис. -fig. 4.3)

```

baranov_georgiy_1132246760@vbox:~$ ssh-keygen -C "Георгий 1132246760@pfur.ru"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/baranov_georgiy_1132246760/.ssh/id_ed25519):
Created directory '/home/baranov_georgiy_1132246760/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/baranov_georgiy_1132246760/.ssh/id_ed25519
Your public key has been saved in /home/baranov_georgiy_1132246760/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:rQruZwEczSV1z01pA6qu5YpNSzCHHSy/UikeP9Jzkvo Георгий 1132246760@pfur.ru
The key's randomart image is:
+--[ED25519 256]--+
|      .  o  +  .  .  |
|      . oo . +  .  |
|      . o  o  .  |
| o .oo .  +  |
| B .oo $  .  |
| .o*o+ .  |
| ..**....  |
| ++B=+ .  |
| o*EB .  |
+----[SHA256]-----+
baranov_georgiy_1132246760@vbox:~$

```

Рис. 4.3: Создание пары ssh ключей

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. -fig. 4.4)

Рис. 4.4: Установка пары ключей

4.3 Создание рабочего пространства и репозитория курса на основе шаблона

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своем рабочем компьютере. (рис. -fig. 4.5)

```

baranov_georgiy_1132246760@vbox:~$ mkdir -p ~/work/study/2024-2025/"Arhitectura pc"
baranov_georgiy_1132246760@vbox:~$

```

Рис. 4.5: Создание рабочей директории

4.4 Создание репозитория курса на основе шаблона

Создаю репозиторий на основе имеющего шаблона (рис. -fig. 4.6) через функционал клонирования интерфейса GitHub.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

Electro20001 / icoNA7r7l29CEIjsl Георгий

✓ Your new repository will be created as ssh-ed25519-
AAAC3NzaC1lZDI1NTE5AAAAICbm9d4cQSkP7ExyTA3E6qGC7JKhcoNA7r7l29CEIjsl-
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [curly-invention](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

Рис. 4.6: Создание репозитория на GitHub

Перехожу в каталог и клонирую репозиторий. (рис. -fig. 4.7)

```
baranov_georgy_1132246760@vbox:~/work/study/2024-2025/Arhetictura pc$ cd ~/work/study/2024-2025/"Arhetictura pc"
baranov_georgy_1132246760@vbox:~/work/study/2024-2025/Arhetictura pc$ git clone --recursive git@
fatal: репозиторий «git@» не существует
baranov_georgy_1132246760@vbox:~/work/study/2024-2025/Arhetictura pc$ git clone --recursive git@github.com:Electro20001/study2024-2025_arh-pc.git arch-pc
Клонирование в «arch-pc»...
```

Рис. 4.7: Копирование репозитория на рабочий компьютер

Создание необходимых каталогов курса (рис. -fig. 4.8)

```

baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc$ git clone --recursive git@
github.com:Electro20001/study2024-2025_arh-pc.git arch-pc
Cloning into 'arch-pc'...
warning: You appear to have cloned an empty repository.
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc$ cd ~/work/study/2024-2025/
"Arhetictura pc"/arch-pc
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ rm package.json
rm: cannot remove 'package.json': No such file or directory
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ echo arch-pc > COU
RSE

```

Рис. 4.8: Настройка каталога

4.5 Задания для самостоятельной работы

Через терминал отправляю предыдущий отчет по лабораторной работе на свой удаленный репозиторий в GitHub

Отслеживание файлов, их отправление в репозиторий и проверка выполнения кода (рис. -fig. 4.9)

```

Installed:
  gc-8.2.2-6.fc40.x86_64  guile30-3.0.7-12.fc40.x86_64  make-1:4.4.1-6.fc40.x86_64

Complete!
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ make
make: *** No targets specified and no makefile found. Stop.
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ git add .
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ git commit -am "fe
at(main): make course structure"
[main (root-commit) 307be4f] feat(main): make course structure
 1 file changed, 1 insertion(+)
 create mode 100644 COURSE
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Electro20001/study2024-2025_arh-pc.git arch-pc
 * [new branch]      main -> main
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ ls ~/work/study/20
24-2025/"Arhetictura pc"/arch-pc/labs
ls: cannot access '/Electro20001/work/study/2024-2025/Arhetictura pc/arch-pc/labs': No s
uch file or directory
baranov_georgiy_1132246760@vbox:/study/2024-2025/Arhetictura pc/arch-pc$ s

```

Рис. 4.9: Заключительные действия

5 Выводы

Мы познакомились с системой контроля git, выучили команды для работы с ним, создали свой репозиторий на платформе github, где в последующем будут храниться все будущие отчеты по лабораторным работам.

Список литературы

1. Лабораторная работа
2. Курс на ТУИС
3. Методические указания