

# Отчёт по лабораторной работе номер 2

## Архитектура компьютера

Титков Ярослав Максимович

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Теоретическое введение.....	1
4	Выполнение лабораторной работы.....	2
5	Выводы.....	7

## 1 Цель работы

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

## 2 Задание

1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.
7. Выполнение заданий для самостоятельной работы

## 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 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. Создал учётную запись github:

```
Cloning into '/home/liveuser/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arhpc/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 | 84.00 KiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02b
d2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5
c748'
liveuser@localhost-live:~/work/study/2024-2025/Архитектура Компьютера$
```

Сначала сделал предварительную конфигурацию git. Открыл терминал и ввёл все данные:

```
liveuser@localhost-live:~$ git config --global user.name "<Titkov Yaroslav>"
liveuser@localhost-live:~$ git config --global user.email "<mandusrek@gmail.com>"
```

Настроил utf-8 в выводе сообщений git:

```
liveuser@localhost-live:~$ git config --global core.quotepath false
```

Задаю

имя начальной ветки Master, параметр autocrlf со значением input, чтобы конвертировать CRLF в L, параметр safecrlf со значением warn

```
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
```

Для

последующей идентификации пользователя на сервере репозитория сгенерировал пару ключей (приватный и открытый):

```
liveuser@localhost-live:~$ ssh-keygen -C "Titkov Yaroslav <mandusrek@gmail.com>"
>
> clear
> "
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:MYQ5owhiZb+Re4/2pK0w+7RLyfxd2z2x/hP1RTQuYPw Titkov Yaroslav <mandusrek@gmail.com>

clear

The key's randomart image is:
+--[ED25519 256]--+
|  .o  o.  .o  o. |
|o..  *..  ...  o|
|o.  = oo   .. o |
|  .  + o   E. o |
|    o .S      +|
```

Захожу

на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key». Вставляю скопированный ключ в поле «Key»

```

Installed:
  xclip-0.13-21.git11cba61.fc40.x86_64

Complete!
liveuser@localhost-live:~$ cat ~/.ssh/id_ed25519.pub
bash: cat ~/.ssh/id_ed25519.pub: No such file or directory
liveuser@localhost-live:~$ ls ~/.ssh/
id_ed25519  id_ed25519.pub
liveuser@localhost-live:~$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHTNgiaJvxGdd++SCTbGmC/Gmyrt0uy4pCcaMzgkRwDM Titkov Yaroslav <mandusrek@gmail.com>

clear

```

## Add new SSH Key

Title

Yaroslav

Key type

Authentication Key

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHTNgiaJvxGdd++SCTbGmC/Gmyrt0uy4pCcaMzgkRwDM Titkov Yaroslav
<mandusrek@gmail.com>
```

clear

|

Открыл

терминал и создал каталог для предмета «Архитектура компьютера»



```
liveuser@localhost-live:~$ mkdir -p ~/work/study/2024-2025/"Архитекутра Компьюетра"
liveuser@localhost-live:~$
```











Репозиторий на основе шаблона создал через web-интерфейс github. Перешёл на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выбрал Use this template:

The image shows two GitHub repository pages. The top page is for the repository `course-directory-student-template` by user `yamadharm`. It is a public template repository with 2 branches, 10 tags, and 59 commits. The file tree shows folders `config` and `template`, and files `.gitattributes`, `.gitignore`, and `.gitmodules`. The bottom page is for the repository `study_2024-2025_arhpc` by user `TitkovYaroslav`. It is a public repository with 1 branch and 0 tags. The file tree shows the same structure as the top repository, with folders `config` and `template`, and files `.gitattributes`, `.gitignore`, and `.gitmodules`.

Клонирую созданный репозиторий с помощью команды `git clone --recursive git@github.com:/study_2024-2025_arh-pc.git` arch-pc: клонирую репрезиторий  
Настроил каталог курса, удалил лишние файлы, затем создал необходимые каталоги и загрузил все на сервер: настроил каталог курса

```
create mode 100755 presentation/report/pandoc/filters/pandoc_secnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/core.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/main.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 presentation/report/report.md
veuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 6 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.40 KiB | 2.44 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:TitkovYaroslav/study_2023-2024_arch-pc.git
  7e98393..6c8c67e  master -> master
veuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

 TitkovYaroslav feat(main): make course structure 6c8c67e · 2 minutes ago  History

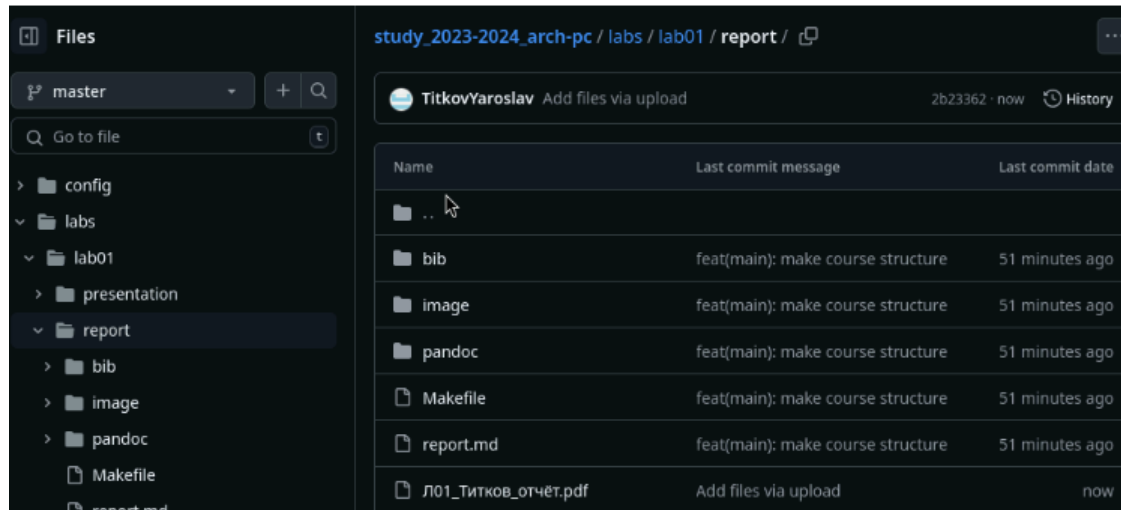
Name	Last commit message	Last commit date
 ..		
 lab01	feat(main): make course structure	2 minutes ago
 lab02	feat(main): make course structure	2 minutes ago
 lab03	feat(main): make course structure	2 minutes ago
 lab04	feat(main): make course structure	2 minutes ago
 lab05	feat(main): make course structure	2 minutes ago
 lab06	feat(main): make course structure	2 minutes ago
 lab07	feat(main): make course structure	2 minutes ago
 lab08	feat(main): make course structure	2 minutes ago
 lab09	feat(main): make course structure	2 minutes ago

Задания для самостоятельной работы:

Я создал отчёт по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs > lab02 > report). Скопировал отчёты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего

пространства. Загрузил файлы на GitHub.

```
localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03/report$ touch L03_Titkov_Yaroslav
localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03/report$ cd ..
localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03$ cd ..
localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$ cd lab
b02/ lab03/ lab04/ lab05/ lab06/ lab07/ lab08/ lab09/ lab10/ lab11/
localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$ cd lab01
localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01$ cd report
```



The screenshot shows the GitHub web interface for the repository 'study\_2023-2024\_arch-pc / labs / lab01 / report'. The left sidebar displays the file tree with the 'report' directory selected. The main area shows a table of files and folders in the repository.

Name	Last commit message	Last commit date
..		
bib	feat(main): make course structure	51 minutes ago
image	feat(main): make course structure	51 minutes ago
pandoc	feat(main): make course structure	51 minutes ago
Makefile	feat(main): make course structure	51 minutes ago
report.md	feat(main): make course structure	51 minutes ago
L01_Титков_отчет.pdf	Add files via upload	now

## 5 Выводы

В ходе выполнения лабораторной работы была достигнута её цель — изучены идеология и применение средств контроля версий, а также приобретены практические навыки по работе с системой Git.