

Отчёт по лабораторной работе 2

Архитектура компьютера

Исмаил Хамза НКАбд-03-24

Содержание

1	Цель работы	5
2	Задания	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Подготовка GitHub репозитория	9
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Шаблонный репозиторий	9
4.2	Создание репозитория	10
4.3	Мой репозиторий	10
4.4	Параметры git	11
4.5	Генерация ключа	11
4.6	Добавляю ключ в аккаунт	12
4.7	Добавляю ключ в аккаунт	12
4.8	Клонирую репозиторий	13
4.9	Создание папок курса	13
4.10	Загрузка	14
4.11	Загрузка	14

Список таблиц

1 Цель работы

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

2 Задания

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 изменения, сделанные разными участниками, 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. 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. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией. 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). Затем можно вносить изменения в локальное дерево и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

4 Выполнение лабораторной работы

4.1 Подготовка GitHub репозитория

Регистрирую учетную запись на GitHub Приступаю к созданию репозитория на основе шаблона. (рис. 4.1, 4.2, 4.3)

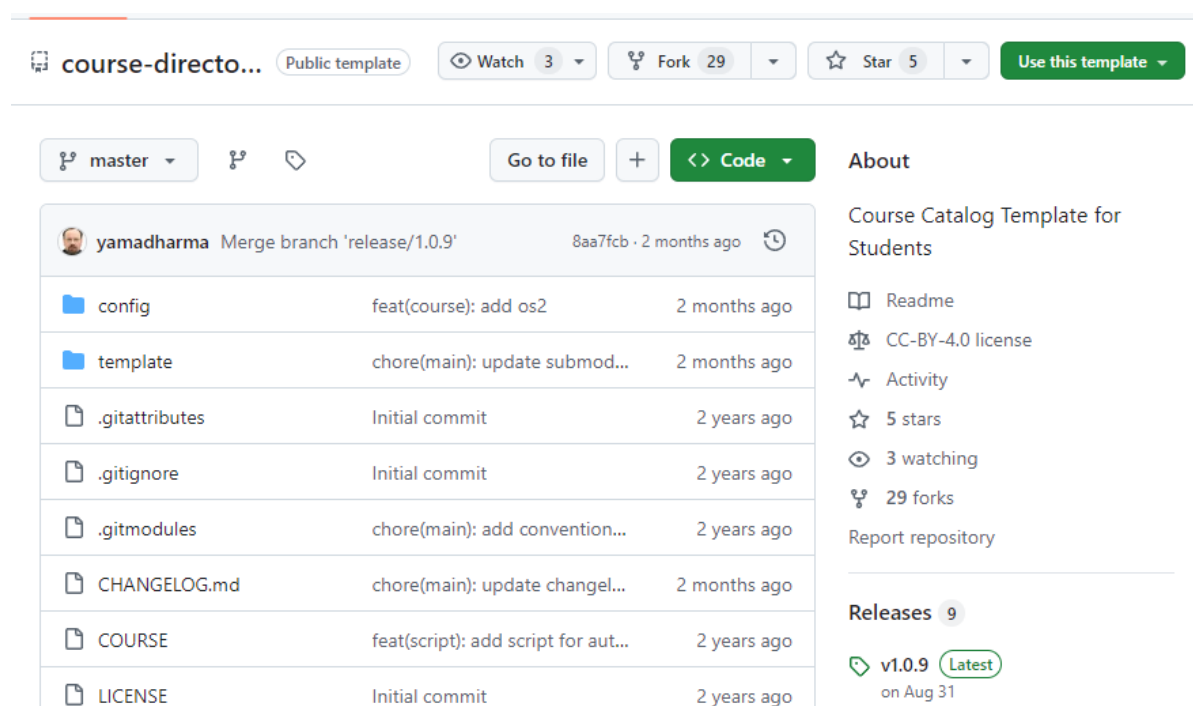


Рис. 4.1: Шаблонный репозиторий

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 (*).

Repository template


 yamadharm/course-directory-student-template

Start your repository with a template repository's contents.

☐ Include all branches

Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

Owner *

 hamzaismail398

Repository name *


arch-pd

arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-dollop](#) ?

Description (optional)

Рис. 4.2: Создание репозитория



 arch-pc

Public









[Pin](#) [Unwatch](#) 1 [Fork](#) 0 [Star](#) 0

generated from [yamadharm/course-directory-student-template](#)

master

[Go to file](#) [+](#) [Code](#)

 hamzaismail398	Initial commit	4add6bc · now
 config	Initial commit	now
 template	Initial commit	now
 .gitattributes	Initial commit	now
 .gitignore	Initial commit	now
 .gitmodules	Initial commit	now
 CHANGELOG.md	Initial commit	now
 COURSE	Initial commit	now

About

No description, website, or topics provided.

[Readme](#) [CC-BY-4.0 license](#) [Activity](#) [0 stars](#) [1 watching](#) [0 forks](#)

Releases

No releases published [Create a new release](#)

Рис. 4.3: Мой репозиторий

Теперь подключимся к репозиторию из системы линукс. Для этого задаем параметры. (рис. 4.4)

```
kismail@Ubuntu:~$  
kismail@Ubuntu:~$ git config --global user.name "hamzaismail398"  
kismail@Ubuntu:~$ git config --global user.email "hazaze190@gmail.com"  
kismail@Ubuntu:~$ git config --global core.quotepath false  
kismail@Ubuntu:~$ git config --global init.defaultBranch master  
kismail@Ubuntu:~$ git config --global core.autocrlf input  
kismail@Ubuntu:~$ git config --global core.safecrlf warn  
kismail@Ubuntu:~$
```


Рис. 4.4: Параметры git

SSH ключ нужен для авторизации пользователя. Создаем его (рис. 4.5)

```
kismail@Ubuntu:~$  
kismail@Ubuntu:~$ ssh-keygen -C "hamzaismail398 hazaze190@gmail.com"  
Generating public/private rsa key pair.  
  
Enter file in which to save the key (/home/kismail/.ssh/id_rsa): Created directory '/home  
/kismail/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/kismail/.ssh/id_rsa  
Your public key has been saved in /home/kismail/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:/SB9haoHK+jaGkG2a6gS/Iqv2UoxArOU0wKtw4T5PQY hamzaismail398 hazaze190@gmail.com  
The key's randomart image is:  
+---[RSA 3072]-----+  
|oo  
|+.E      .  
|=B0+     . .  
|=*+.+    o . .  
|++o. .  S = .  
|oooo .   = +  
|.o= . . o . .  
|=+ =     . .  
|B=*oo  
+---[SHA256]-----+  
kismail@Ubuntu:~$
```

Рис. 4.5: Генерация ключа

Теперь данные ключа нужно добавить в профиль на гитхабе. Тогда гитхаб будет узнавать нас по ключу. (рис. 4.6, 4.7)



hamzaismail398 (hamzaismail398)

Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Add new SSH Key

Title

kismail

Key type


Authentication Key

Key

AAAAAB3NzaC1yc2EAAAADAQABAAQDAgQDp0sniZ/1s/vmCtuqT0rTt0avsn1zqXdpWYNlrZBD8dczmAfVWX+/+x+XqkR/35oRxOIFN4zN4VqQZjYygh7h7lqHWhcQPKttGF8xWBXHR/mwaGLzZugh84M+dx5H+QohzXky1zNytMgyynTPMU6PDBC9mrZENjuEPWzh1mHGQkHsch2niCm0ia4/49v11Gxrd9QaLsy8S7OLYJWPaWciXQ+GVp8IIW78V9NNLklwvDZsQhAaRiKhFC1+tyQeOdqSjl3dwPH0lsc2Q0e72fLfpFI9Ty5IH91eHumth2gHfUJRUCUCZuDvfxWoJMnJ7Mr2rcrgWppkmF9KfsuubpEGHghCIA7mZQuHHc5siWsfmr9OWCApPCeDlv5bzJsLawyMPYclY9W3NQh7RbgITD4Vm2KDVGxAPSYk86RIGRWRP/1fl/fRrtxUxljZURr8X/w3i0xLoBmJS57qsB4LSX7ofn7I+QFG69/Evb/WxlPpx5hfh7uKmnNshQcvemGd0= hamzaismail398hazaze190@gmail.com

Add SSH key

Рис. 4.6: Добавляю ключ в аккаунт



hamzaismail398 (hamzaismail398)

Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions


SSH keys

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



kismail

SHA256: /SB9haoHK+jaGkG2a6gS/Iqv2UoxAr0U0wKtw4T5PQY

Added on Oct 12, 2024

Never used — Read/write

Delete

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

Рис. 4.7: Добавляю ключ в аккаунт

12

Создаем папку на компьютере и клонируем в нее содержимое репозитория, т.е. шаблон. (рис. 4.8)]

```
rkdown-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/kismail/work/study/2024-2025/Архитектура компьютера/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 | 1.11 MiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/kismail/work/study/2024-2025/Архитектура компьютера/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 | 1.88 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера$
```

Рис. 4.8: Клонирование репозитория

Оформили курс по шаблону и загрузили в сетевой репозиторий (рис. 4.9, 4.10)

```
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-2025/"
Архитектура компьютера"/arch-pc
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COUR
SE
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config        labs    Makefile  presentation  README.git-flow.md  template
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.9: Создание папок курса

```

create mode 100644 presentation/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 presentation/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_fignos.py
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
kismail@Ubuntu:~/work/study/2024-2025/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), 341.27 KiB | 2.34 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:hamzaismail398/arch-pc.git
4add6bc..6980337 master -> master
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 4.10: Загрузка

Также загрузили в сетевой репозиторий отчеты по сделанным работам (рис. 4.11)

```

kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -m 'add lab01'
[master 114030a] add lab01
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/lab1-_ismail_khamza.docx
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Warning: Permanently added the ECDSA host key for IP address '140.82.121.4' to the list of known hosts.
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 6 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.49 MiB | 2.17 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:hamzaismail398/arch-pc.git
6980337..114030a master -> master
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
kismail@Ubuntu:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 4.11: Загрузка

5 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git.

Список литературы

1. Архитектура ЭВМ
2. Git - gitattributes Документация