

Отчёт по лабораторной работе 2

Архитектура компьютеров

Махкамов Рауфджон НММбд-04-24

Содержание

1	Цель работы	5
2	Теоретическое введение	6
2.1	Системы контроля версий. Общие понятия	6
3	Выполнение лабораторной работы	8
3.1	Настройка github	8
3.2	Создание репозитория курса на основе шаблона	10
3.3	Настройка каталога курса	10
3.4	Задание для самостоятельной работы	12
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	настраиваю Git	8
3.2	Генерирую SSH-ключи	9
3.3	Добавляю ключ на GitHub	9
3.4	Репозиторий-шаблон	10
3.5	Копирование шаблона	10
3.6	Клонирование репозитория	11
3.7	Подготовка папок	11
3.8	Отправка изменений в гитхаб	12
3.9	Отправка изменений в гитхаб	12
3.10	Обзор репозитория	13

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

2.1 Системы контроля версий. Общие понятия

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.

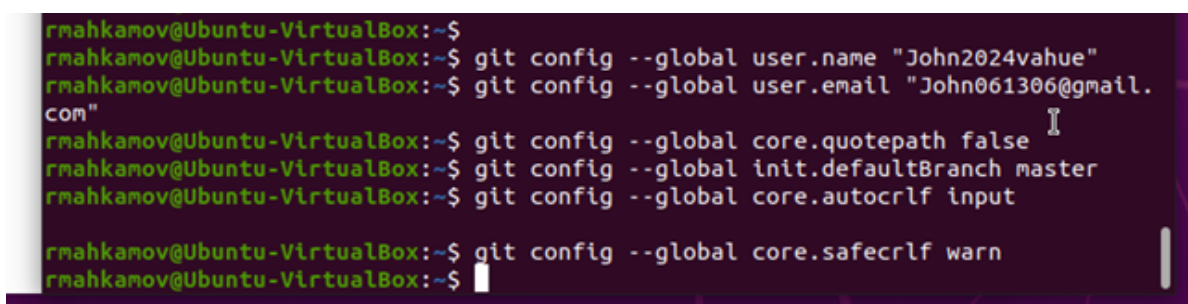
Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких

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.

3 Выполнение лабораторной работы

3.1 Настройка github

Сначала настраиваю Git на своем компьютере. Создаю пользователя в системе Git и указываю параметры — имя и email, чтобы мои действия были правильно подписаны. (рис. 3.1)

A screenshot of a terminal window with a dark background and light-colored text. The prompt is 'rmahkamov@Ubuntu-VirtualBox:~\$'. The user enters several 'git config' commands to set global user information and core settings. The commands are: 'git config --global user.name "John2024vahue"', 'git config --global user.email "John061306@gmail.com"', 'git config --global core.quotePath false', 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'. The terminal shows the output of each command, which is typically empty for these types of configurations.

```
rmahkamov@Ubuntu-VirtualBox:~$  
rmahkamov@Ubuntu-VirtualBox:~$ git config --global user.name "John2024vahue"  
rmahkamov@Ubuntu-VirtualBox:~$ git config --global user.email "John061306@gmail.  
com"  
rmahkamov@Ubuntu-VirtualBox:~$ git config --global core.quotePath false  
rmahkamov@Ubuntu-VirtualBox:~$ git config --global init.defaultBranch master  
rmahkamov@Ubuntu-VirtualBox:~$ git config --global core.autocrlf input  
rmahkamov@Ubuntu-VirtualBox:~$ git config --global core.safecrlf warn  
rmahkamov@Ubuntu-VirtualBox:~$
```

Рис. 3.1: настраиваю Git

Генерирую SSH-ключи, которые нужны для идентификации на GitHub. (рис. 3.2) Сохраняю их на компьютере и добавляю публичный ключ в свой профиль на GitHub для установления связи. (рис. 3.3)


```
rmahkamov@Ubuntu-VirtualBox: ~  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/rmahkamov/.ssh/id_rsa): Created direc  
tory '/home/rmahkamov/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/rmahkamov/.ssh/id_rsa  
Your public key has been saved in /home/rmahkamov/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:NdVYPbA9+AU/6M+H3nipAHZjv4Tg1y6QFXiLhypeaJc John2024vahue John061306@gmai  
l.com  
The key's randomart image is:  
+----[RSA 3072]-----+  
|          .oB=. |  
|          .o=.0+. |  
|          o++*. *+ |  
|          .. ++O+ + |  
|          Sooo.o + |  
|          + E+ . =. |  
|          o + o ..oo |  
|          .   o. +o |  
|          .+.. |  
+-----[SHA256]-----+  
rmahkamov@Ubuntu-VirtualBox:~$
```

Рис. 3.2: Генерирую SSH-ключи

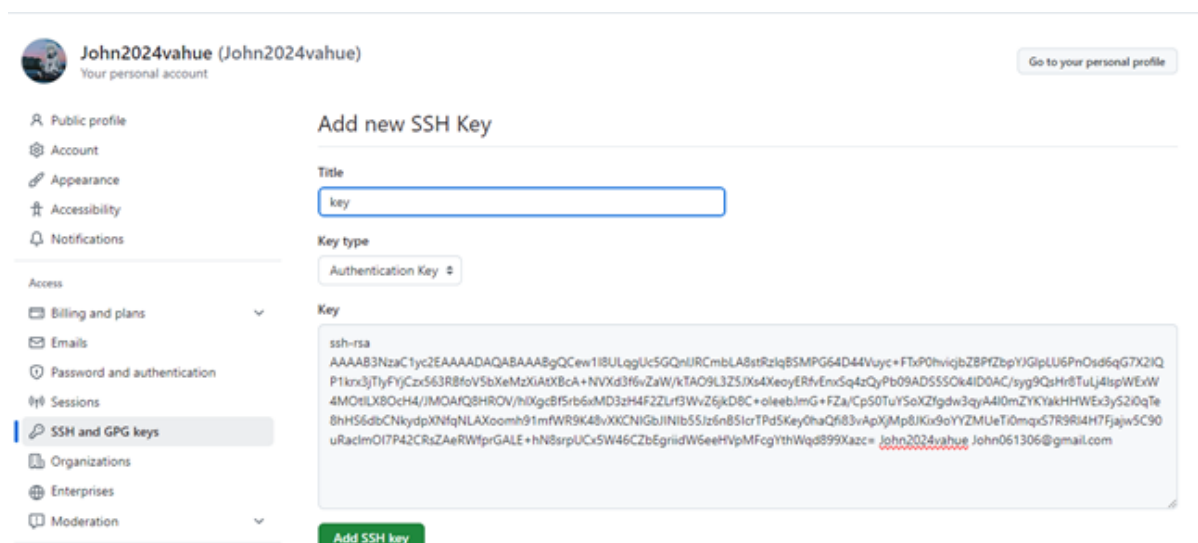


Рис. 3.3: Добавляю ключ на GitHub

3.2 Создание репозитория курса на основе шаблона

Нахожу нужный репозиторий-шаблон (рис. 3.4) и делаю из него копию для работы (рис. 3.5).

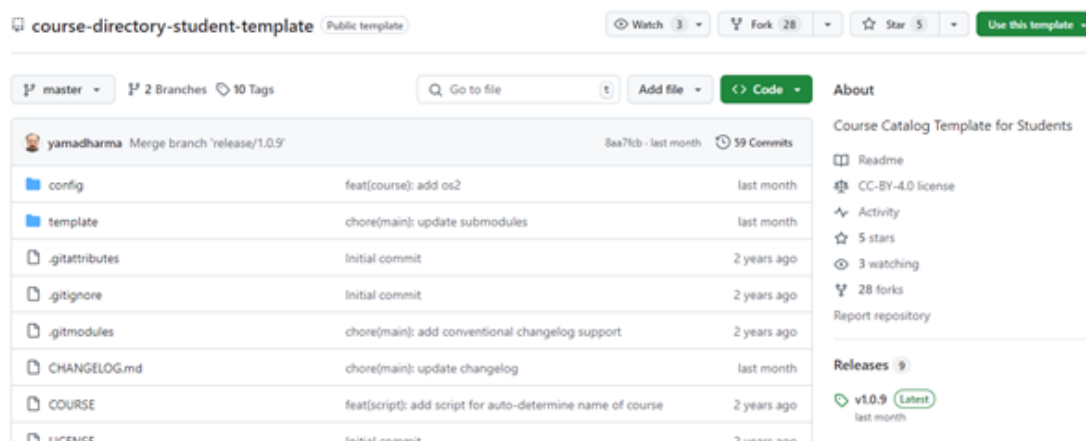


Рис. 3.4: Репозиторий-шаблон

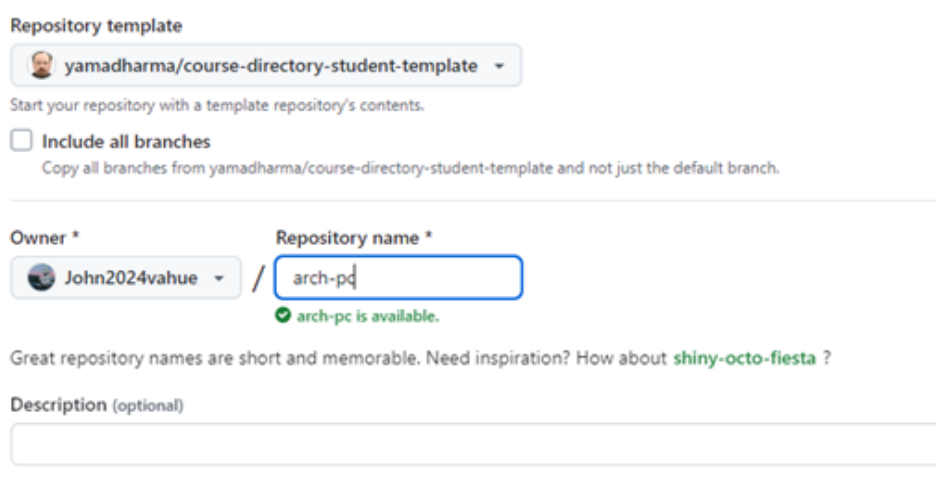


Рис. 3.5: Копирование шаблона

3.3 Настройка каталога курса

Создаю рабочую директорию на своем компьютере для хранения файлов проекта. Клонировать репозиторий с GitHub в эту директорию, чтобы работать с

файлами локально. (рис. 3.6) (рис. 3.7)

```
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:John2024vahue/arch-pc.git
Cloning into '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 | 3.13 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/rmahkamov/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 | 1.02 MiB/s, done.
```

Рис. 3.6: Клонирование репозитория

```
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ ls
arch-pc
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ cd arch-pc/
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md
config        LICENSE package.json README.git-flow.md  template
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
rmahkamov@Ubuntu-VirtualBox:~/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
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 3.7: Подготовка папок

После подготовки структуры добавляю изменения в локальный репозиторий и отправляю их на GitHub командой push. (рис. 3.8)

```
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
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Warning: Permanently added the ECDSA host key for IP address '140.82.121.3' to the list of known hosts.
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 6 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (34/34), 341.23 KiB | 2.53 MiB/s, done.
Total 34 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:John2024vahue/arch-pc.git
   22b907a..562416b  master -> master
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```


Рис. 3.8: Отправка изменений в гитхаб

3.4 Задание для самостоятельной работы

Загружаю отчёты по выполненным работам в соответствующие папки на GitHub, обновляя репозиторий по мере необходимости. (рис. 3.9) (рис. 3.10)

```
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'upload lab01'
[master 79227f9] upload lab01
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/отчет_лаб_1.docx
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 800.28 KiB | 6.20 MiB/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:John2024vahue/arch-pc.git
   562416b..79227f9  master -> master
rmahkamov@Ubuntu-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 3.9: Отправка изменений в гитхаб

 arch-pc Public

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

[Pin](#) [Unwatch](#) 1


master

1 Branch

0 Tags

Add file

Code

 John2024vahue

upload lab01

79227f9 · now

3 Commits

config	Initial commit	2 minutes ago
labs	upload lab01	now
presentation	feat(main): make course structure	1 minute ago
template	Initial commit	2 minutes ago
.gitattributes	Initial commit	2 minutes ago
.gitignore	Initial commit	2 minutes ago
.gitmodules	Initial commit	2 minutes ago

Рис. 3.10: Обзор репозитория

4 Выводы

В ходе выполнения работы изучили работу с GitHub.

Список литературы

1. Архитектура ЭВМ - Материалы курса
2. Markdown Документация
3. GitHub Документация