

Отчет по лабораторной работе №3

дисциплина: Архитектура компьютера

Кудинец Максим Антонович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Задания для самостоятельной работы	16
6	Выводы	17

Список иллюстраций

4.1	Базовая настройка Git	9
4.2	Создание SSH-ключа	10
4.3	Новый SSH-ключ	10
4.4	Создание директории	10
4.5	Страница создания репозитория	11
4.6	Копирование репозитория	11
4.7	Создание и удаление файлов и директорий	11
4.8	Команда make и создание коммита	12
4.9	Загрузка коммита на сервер	12
4.10	Список файлов в папке для лабораторных работ	12
4.11	Создание файла отчета	13
4.12	Копирование файлов	14
4.13	Коммит изменений	14
4.14	Загрузка данных на сервер	15
4.15	Лабораторная работа 1	15

Список таблиц

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий, а также приобретение практических навыков 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 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

1. Настройка Github На гитхабе у меня уже была учётная запись и поэтому создавать новую мне не пришлось.
2. Базовая настройка git Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду `git config --global "user.name"`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою. Настраиваю utf-8 в выводе сообщений git для корректного отображения символов. Задаю имя «master» для начальной ветки. Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах. CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах. Задаю параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость. При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
makudinets@fedora:~$ git config --global user.name "MaKudinets"
makudinets@fedora:~$ git config --global user.email "terracopter695@gmail.com"
makudinets@fedora:~$ git config --global core.quotepath false
makudinets@fedora:~$ git config --global init.defaultBranch master
makudinets@fedora:~$ git config --global core.autocrlf input
makudinets@fedora:~$ git config --global core.safecrlf warn
makudinets@fedora:~$
```

Рис. 4.1: Базовая настройка Git

3. Создание SSH-ключа Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия,`

work@email”, указывая имя владельца и электронную почту владельца.
Ключ автоматически сохранится в каталоге ~/.ssh/.

```
makudinets@fedora:~$ ssh-keygen -C "Кудинец Максим <terracofter695@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/makudinets/.ssh/id_ed25519):
Created directory '/home/makudinets/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/makudinets/.ssh/id_ed25519
Your public key has been saved in /home/makudinets/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:zQ7UKEHnQGhD+yFhBttV0WA9R1SdFIZH1puRqFFJ9Kk Кудинец Максим <terracofter695@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|  .o=*.B%Bo      |
|   *X.*0         |
|  o.=.O =o..     |
|   o + B o       |
|  . E + S o      |
|       o         |
|       .         |
+-----[SHA256]-----+
makudinets@fedora:~$
```

Рис. 4.2: Создание SSH-ключа

Копирую ключ из открытой директории в которой он был, перехожу на сайт Github и добавляю там новый SSH-ключ



Рис. 4.3: Новый SSH-ключ

4. Создание рабочего пространства и репозитория курса на основе шаблона
Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -p создаю все директории после домашней ~ /work/study/2022-2023/“Архитектура компьютера” рекурсивно.

```
makudinets@fedora:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
makudinets@fedora:~$
```

Рис. 4.4: Создание директории

5. Создание репозитория курса на основе шаблона В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория. В открывшемся окне задаю имя репозитория (Repository name): study_2022– 2023_arh- pc и создаю репозиторий, нажимаю на кнопку ‘Create repository from template’
- Репозиторий создан

```
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd labs/lab02/report
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ touch /02_Кудинец_отчет
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ cd ..
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02$ cd ..
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab01/report
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис. 4.5: Страница создания репозитория

Через терминал перехожу в созданный каталог курса и клонирую созданный репозиторий с помощью команды `git clone –recursive git@github.com:/study_2024–2025_arh-pc.git arch-pc`

```
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.6: Копирование репозитория

6. Настройка каталога курса Перехожу в каталог arch-pc с помощью утилиты `cd` Удаляю лишние файлы с помощью утилиты `rm` Создаю необходимые каталоги Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью `git add`, комментирую и сохраняю изменения на сервере как добавление курса с помощью `git commit`

```
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ ls ~/Загрузки
/01_Кудинец_отчет.pdf
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ █
```

Рис. 4.7: Создание и удаление файлов и директорий

```
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 341.40 КиБ | 2.97 МиБ/с, готово.
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:MaKudinets/study_2024-2025_arh-pc.git
   a9d3dba..b1fa4e4  master -> master
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.8: Команда make и создание коммита

С помощью команды push выгружаем всё на сервер

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
lab10	feat(main): make course structure	2 minutes ago
lab11	feat(main): make course structure	2 minutes ago
README.md	feat(main): make course structure	2 minutes ago
README.ru.md	feat(main): make course structure	2 minutes ago

Рис. 4.9: Загрузка коммита на сервер

Проверяю правильность выполнения работы сначала на самом сайте GitHub

```
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:MaKudinets/study_2024-2025_arh-pc.git arch-pc
Клонирование в arch-pc...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:DIY3wVv6UzJhbp21sf/zLD8zPHSVHokR4UvCOQu.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 34 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (34/34), 19.58 КиБ | 19.58 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Клонирование в /home/makudinets/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)
Получение объектов: 100% (111/111), 102.17 КиБ | 1.11 МиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в /home/makudinets/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)
Получение объектов: 100% (142/142), 341.89 КиБ | 2.35 МиБ/с, готово.
Определение изменений: 100% (68/68), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера$
```

Рис. 4.10: Список файлов в папке для лабораторных работ

7. Выполнение заданий для самостоятельной работы Перехожу в директорию labs/lab03/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch, после чего перехожу в директорию labs/lab01/report

```

makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submodules

makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ add .
bash: add: команда не найдена...
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master b1fa4e4] feat(main): make course structure
223 files changed, 53681 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/.projectile
create mode 100644 labs/lab02/presentation/.texlabroot
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg

```

Рис. 4.11: Создание файла отчета


Проверяю местонахождение файла с отчетом по первой лабораторной работой. Он должен быть в подкаталоге домашней директории «Загрузки», для проверки использую команду `ls`

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 * **Repository name ***

 MaKudinets ▾ / study_2024-2025_arh-pc


✓ study_2024-2025_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about **congenial-tribble** ?

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.

 You are creating a public repository in your personal account.

Create repository

Копирую первую

лабораторную с помощью утилиты `ср` и проверяю правильность выполнения команды `ср` с помощью `ls`

```
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ cp -r /usr/share/doc/Arhitektura_kompyutera/arch-pc/lab01/report/* .
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
ls: cannot access 'Makefile': No such file or directory
ls: cannot access 'report.ed': No such file or directory
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис. 4.12: Копирование файлов

После чего добавляю файл в `add` Сохраняю изменения на сервере командой `git commit -m "..."`, поясняя, что добавил файлы.

```
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git push
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 614.60 КБ | 4.49 МБ/с, готово.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:MaKudinets/study_2024-2025_arh-pc.git
 b1fa4e4..c84381d master -> master
makudinets@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис. 4.13: Коммит изменений

Аналогично я поступаю с отчётом по этой лабораторной работе, после чего отправляю коммит на сервер.

```

makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git add /01_Кудинец_отчет.pdf
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git commit -m "Add lab01/report"
[master c84381d] Add lab01/report
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/01_Кудинец_отчет.pdf
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git status
Текущая ветка: master
Ваша ветка опережает «origin/master» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)


Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
.../.. /lab02/report/02_Кудинец_отчет



индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
makudinets@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$

```

Рис. 4.14: Загрузка данных на сервер

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report

study_2024-2025_arch-pc / labs / lab01 / report /  Add file ...

 Makudinets Add lab01/report c84381d - 1 minute ago  History







Name	Last commit message	Last commit date
..		
 bib	feat(main): make course structure	28 minutes ago
 image	feat(main): make course structure	28 minutes ago
 pandoc	feat(main): make course structure	28 minutes ago
 Makefile	feat(main): make course structure	28 minutes ago
 report.md	feat(main): make course structure	28 minutes ago
 01_Кудинец_отчет.pdf	Add lab01/report	1 minute ago

Рис. 4.15: Лабораторная работа 1

5 Задания для самостоятельной работы

6 Выводы

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