

Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Арбатова Варвара Петровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выполнение заданий для самостоятельной работы	14
6	Выводы	16

Список таблиц

Список иллюстраций

4.1	Рис1. Регистрация в github	9
4.2	Рис2. Базовая настройка git	9
4.3	Рис3. Генерирую ключи	10
4.4	Рис4. Перехожу в свой аккаунт на github	10
4.5	Рис7.	10
4.6	Рис8.	11
4.7	Рис9. Репозиторий создан	11
4.8	Рис10. Переход в домашний каталог	11
4.9	Рис11. Отправка созданных каталогов	12
4.10	Рис12. Удаление лишних файлов	12
4.11	Рис13. Создание каталогов	12
4.12	Рис14. Выгрузка изменений на сервер	12
4.13	Рис15. Страница репозитория	13
5.1	Рис16. Создание файла	14
5.2	Рис17. Перемещение и проверка файлов	14
5.3	Рис18. Добавление 2 лабораторной	14
5.4	Рис19. Добавление 1 лабораторной	15
5.5	Рис20. Проверка файлов	15

1 Цель работы

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

2 Задание

- 1) Создать аккаунт в github
- 2) Базовая настройка github
- 3) Создать SSH-ключ
- 4) Настроить каталог курса на основе шаблона
- 5) Выполнить задания 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 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. 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

2.4.1) Настройка github Регистрируюсь в github с корпоративной почтой и заполняю основные данные

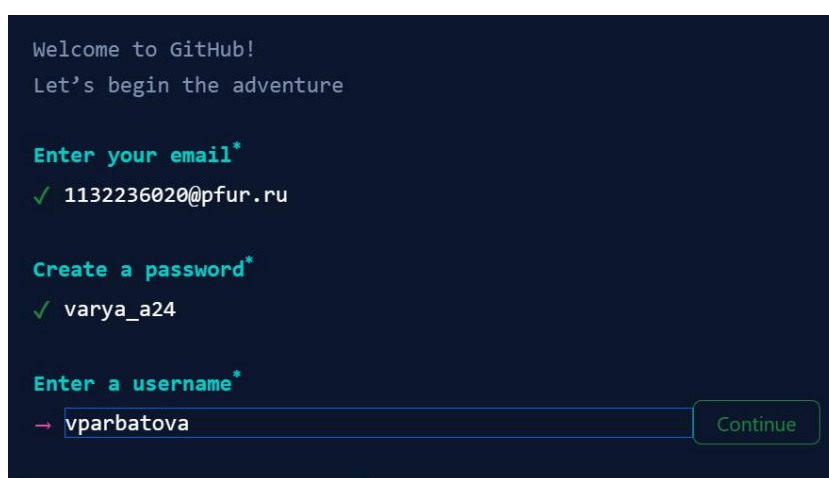


Рис. 4.1: Рис1. Регистрация в github

2.4.2) Базовая настройка git Сначала делаю предварительную конфигурацию git. Настраиваю utf-8 в выводе сообщений git, задаю имя начальной ветки, параметры autocrlf и safecrlf

```
vparbatova@Varvarishe:~$ git config --global user.name "<Varvara Arbatova>"
vparbatova@Varvarishe:~$ git config --global user.email "<1132236020@pfur.ru>"
vparbatova@Varvarishe:~$ git config --global core.quotepath false
vparbatova@Varvarishe:~$ git config --global init.defaultBranch master
vparbatova@Varvarishe:~$ git config --global core.autocrlf input
vparbatova@Varvarishe:~$ git config --global core.safecrlf warn
vparbatova@Varvarishe:~$
```

Рис. 4.2: Рис2. Базовая настройка git

2.4.3) Создание SSH-ключа Генерирую пару ключей.

```
vparbatova@Varvarishe:~$ ssh-keygen -C "Varvara Arbatova <1132236020@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vparbatova/.ssh/id_rsa):
/home/vparbatova/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vparbatova/.ssh/id_rsa
Your public key has been saved in /home/vparbatova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:I5poGgyxFQibjffVoghHjVN5QQPZJ+uiV12X0+xoW8A Varvara Arbatova <1132236020@pfur.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|+oo+o*+|
|.=+..+..|
|+ o.o.+ |
|o+ + .. .o|
|+. ...S.Eo|
|o. o.oo...+|
|..o.oo o..|
|+. . o..|
|. . ...|
+---[SHA256]-----+
```

Рис. 4.3: Рис3. Генерирую ключи

Перехожу в аккаунт на github

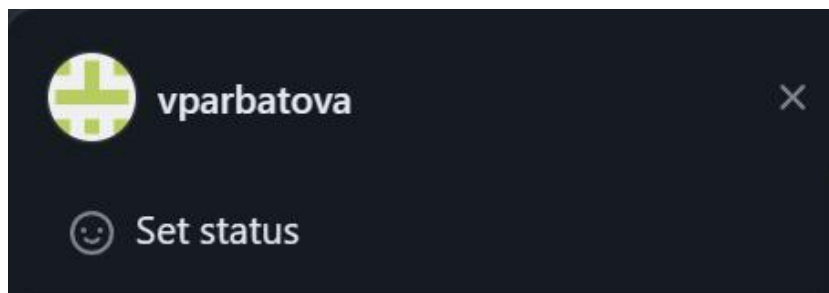


Рис. 4.4: Рис4. Перехожу в свой аккаунт на github

Копирую из локальной консоли ключ в буфер обмена

```
vparbatova@Varvarishe:~$ cat ~/.ssh/id_rsa.pub |
vparbatova@Varvarishe:~$
```

2.4.4) Сознание рабочего пространства и репозитория курса на основе шаблона С помощью команды mkdir -p создаю вложенные друг папки, проверяю корректность создания

```
vparbatova@Varvarishe:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
vparbatova@Varvarishe:~$ ls
work
vparbatova@Varvarishe:~$
```

Рис. 4.5: Рис7.

Создаю репозиторий на основе шаблона через web-интерфейс github.

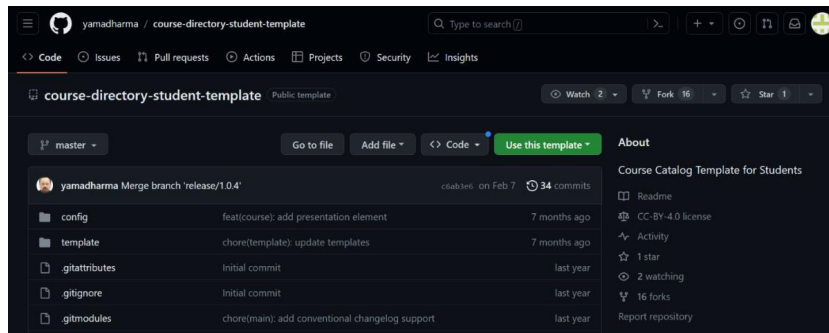


Рис. 4.6: Рис8.

Репозиторий создан

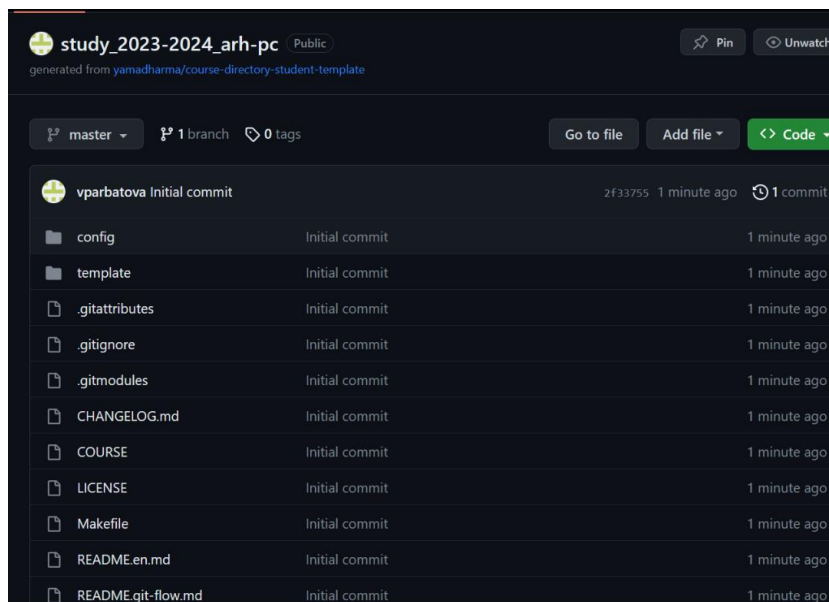


Рис. 4.7: Рис9. Репозиторий создан

Перехожу в нужный мне каталог с помощью cd

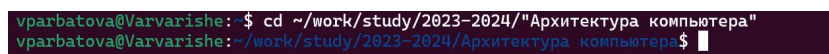


Рис. 4.8: Рис10. Переход в домашний каталог

Клонирую созданный репозиторий с помощью команды git clone –recursive

```
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:vparbatova/study_2023-2024_arh-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (27/27), 16.93 KiB | 361.00 KiB/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/vparbatova/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation'...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Receiving objects: 100% (82/82), 92.90 KiB | 827.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into '/home/vparbatova/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report'...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Receiving objects: 100% (101/101), 327.25 KiB | 444.00 KiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
```

Рис. 4.9: Рис11. Отправка созданных каталогов

С помощью rm удаляю лишние файлы из каталога

```
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера$ cd arch-pc
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера/arch-pc$ rm package.json
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 4.10: Рис12. Удаление лишних файлов

Создаю необходимые каталоги

```
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера/arch-pc$ make
```

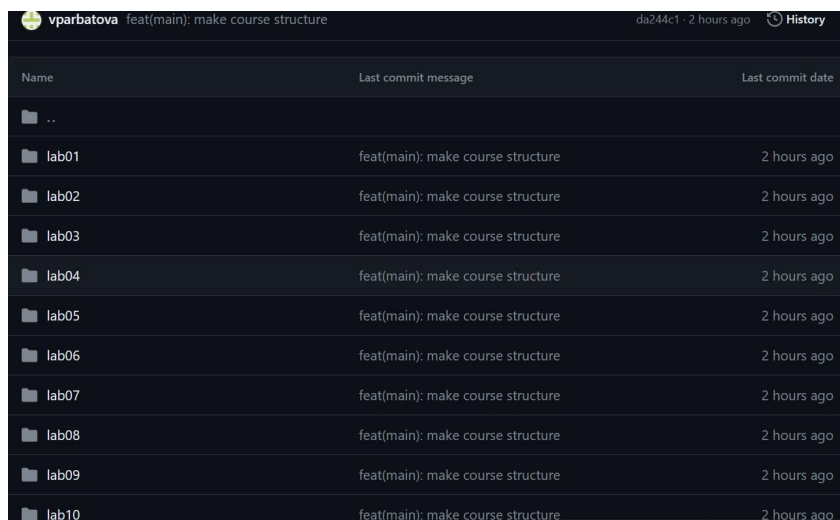
Рис. 4.11: Рис13. Создание каталогов

С помощью git add добавляю все созданные каталоги, затем комментирую и сохраняю изменения на сервере с помощью git commit, и отправляю все на сервер с помощью git push

```
vparbatova@Varvarishe: /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 12 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.13 KiB | 3.20 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:vparbatova/study_2023-2024_arh-pc.git
2f33755..da244c1 master -> master
vparbatova@Varvarishe: /work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 4.12: Рис14. Выгрузка изменений на сервер

Проверяю правильность выполнения всех команд



The screenshot shows a repository page for user 'vparbatova'. The header includes the username, the current branch 'feat(main): make course structure', the commit hash 'da244c1', and the time '2 hours ago'. A 'History' button is visible. Below the header is a table with three columns: 'Name', 'Last commit message', and 'Last commit date'. The table lists ten commits, each with a folder icon and the name 'lab01' through 'lab10'. All commit messages are 'feat(main): make course structure' and all dates are '2 hours ago'.

Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	2 hours ago
lab02	feat(main): make course structure	2 hours ago
lab03	feat(main): make course structure	2 hours ago
lab04	feat(main): make course structure	2 hours ago
lab05	feat(main): make course structure	2 hours ago
lab06	feat(main): make course structure	2 hours ago
lab07	feat(main): make course structure	2 hours ago
lab08	feat(main): make course structure	2 hours ago
lab09	feat(main): make course structure	2 hours ago
lab10	feat(main): make course structure	2 hours ago

Рис. 4.13: Рис15. Страница репозитория

5 Выполнение заданий для самостоятельной работы

Создаю файл в папке

```
vparbatova@Varvarishe: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd labs/lab03/report
vparbatova@Varvarishe: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03/report$ touch Л02_Арбатов
a_отчет
```

Рис. 5.1: Рис16. Создание файла

Перемещаю файл с ранее выполненной работой 1 и проверяю корректность выполнения

```
vparbatova@Varvarishe: $ mv ~/Л01_Арбатова_отчет.pdf ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/lab
s/lab01/report
vparbatova@Varvarishe: ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
Makefile lib image pandoc report.md Л01_Арбатова_отчет.pdf
vparbatova@Varvarishe: $
```

Рис. 5.2: Рис17. Перемещение и проверка файлов

Выгружаю 2 работу на сайт

```
vparbatova@Varvarishe: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
vparbatova@Varvarishe: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am "Добавление файла"
[master c456a5d] Добавление файла
1 file changed, 0 insertions(+), 0 deletions(-)
vparbatova@Varvarishe: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 524.87 KiB | 5.64 MiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:vparbatova/study_2023-2024_arh-pc.git
736250a..c456a5d master -> master
vparbatova@Varvarishe: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ █
```


Рис. 5.3: Рис18. Добавление 2 лабораторной


Выгружаю 1 работу на сайт аналогично

```
vparbatova@Varvarishe: $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
vparbatova@Varvarishe:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $ git add л01_Арбатова_отчет.pdf
```

Рис. 5.4: Рис19. Добавление 1 лабораторной

Проверяю выгрузились ли файлы

study_2023-2024_arh-pc / labs / lab02 / report /  Add file ▾ ⋮

 **vparbatova** · Добавление файла c456a5d · 3 hours ago History









Name	Last commit message	Last commit date
 ..		
 bib	feat(main): make course structure	last week
 image	feat(main): make course structure	last week
 pandoc	feat(main): make course structure	last week
 Makefile	feat(main): make course structure	last week
 report.md	feat(main): make course structure	last week
 ~\$2_Арбатова_отчет	Добавление файла	2 days ago
 л02_Арбатова_отчет	Добавление файла	3 hours ago

Рис. 5.5: Рис20. Проверка файлов

6 Выводы

Я создала репозиторий в github, познакомилась с работой в системе git, выгрузила к себе в аккаунт свои лабораторные работы # Список литературы{unnumbered}