

Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

Воронов Александр Валерьевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	9
3.1	Настройка Github	9
4	Выводы	16
5	Список литературы	17

Список иллюстраций

3.1	Предварительная конфигурация git	9
3.2	Настраиваю utf-8	9
3.3	Имя начальной ветки	9
3.4	Использование autocrlf	9
3.5	Использование safecrlf	10
3.6	Генерирование ключей SSH	10
3.7	Копирование SSH-ключа	10
3.8	Вставка SSH-ключа	11
3.9	Создание репозитория	11
3.10	Выбор шаблона	11
3.11	Создание репозитория	12
3.12	Переход в каталог курса	12
3.13	Клонирование репозитория	12
3.14	Удаление лишних файлов в каталоге курса	13
3.15	Создание нужных каталогов и отправка на сервер	13
3.16	Проверка проделанной работы	13
3.17	Создание отчета	13
3.18	Переход в директорию labs/lab01/report	14
3.19	Проверка расположения файла	14
3.20	Копирование файла	14
3.21	Комментарии к изменениям	15
3.22	Отправление на сервер	15
3.23	Проверка на нахождения файла	15

Список таблиц

1 Цель работы

Целью работы является применение средств контроля версий. А также очень важно приобрести практические навыки по работе с системой git.

2 Задание

- Настройка GitHub.
- Базовая настройка Git.
- Создание SSH-ключа.
- Создание рабочего пространства и репозитория курса на основе шаблона.
- Создание репозитория курса на основе шаблона.
- Настройка каталога курса.
- Выполнение заданий для самостоятельной работы.

#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 локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

3 Выполнение лабораторной работы

3.1 Настройка Github

Для выполнения лабораторной работы создаю учетную запись на <https://github.com/>

##Базовая настройка git

Делаю предварительную конфигурацию git. Захожу в терминал и ввожу команды, указывая свое имя и email (рис. -fig. 3.1)

```
avvoronov@fedora:~$ git config --global user.name "<Alexander Voronov>"
avvoronov@fedora:~$ git config --global user.email "<1132243812@pfur.ru>"
avvoronov@fedora:~$
```

Рис. 3.1: Предварительная конфигурация git

Настраиваю utf-8 в выходе сообщений git.(рис. -fig. 3.2)

```
avvoronov@fedora:~$ git config --global core.quotepath false
avvoronov@fedora:~$
```

Рис. 3.2: Настраиваю utf-8

Задаю имя начальной ветки, которую буду называть master (рис. -fig. 3.3)

```
avvoronov@fedora:~$ git config --global init.defaultBranch master
avvoronov@fedora:~$
```

Рис. 3.3: Имя начальной ветки

Ввожу autocrlf и safecrlf (рис. -fig. 3.4)

```
avvoronov@fedora:~$ git config --global core.autocrlf input
avvoronov@fedora:~$
```

Рис. 3.4: Использование autocrlf

(рис. -fig. 3.5)

```
avvoronov@fedora:~$ git config --global core.safecrlf warn
avvoronov@fedora:~$
```

Рис. 3.5: Использование safecrlf

##Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория генерирую пару ключей (приватный и открытый). (рис. -fig. 3.6)

```
avvoronov@fedora:~$ ssh-keygen -C "Alexander Voronov <1132243812@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/avvoronov/.ssh/id_ed25519):
Created directory '/home/avvoronov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/avvoronov/.ssh/id_ed25519
Your public key has been saved in /home/avvoronov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:VbSn0/ghALLYIvcXNEkMT3MZtcZ09FKgwS3vo5Dz/HI Alexander Voronov <1132243812@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|      .O+ O=+++.  |
|      +.+.OO*+.  |
|      . *  *OO O  |
|    o + o o .+   |
|  . + o . S. ..  |
| o o .+. .o.    |
|      . . =o.+   |
|      .  =oEo    |
|      +o        |
+-----[SHA256]-----+
avvoronov@fedora:~$
```

Рис. 3.6: Генерирование ключей SSH

Чтобы скопировать из локальной консоли ключ в буфер обмена, устанавливаю команду xclip. Затем воспользуюсь командой xclip (рис. -fig. 3.7)

```
avvoronov@fedora:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
avvoronov@fedora:~$
```

Рис. 3.7: Копирование SSH-ключа

Вставляю ключ в появившееся на сайте поле, указываю его имя.(рис. -fig. 3.8)

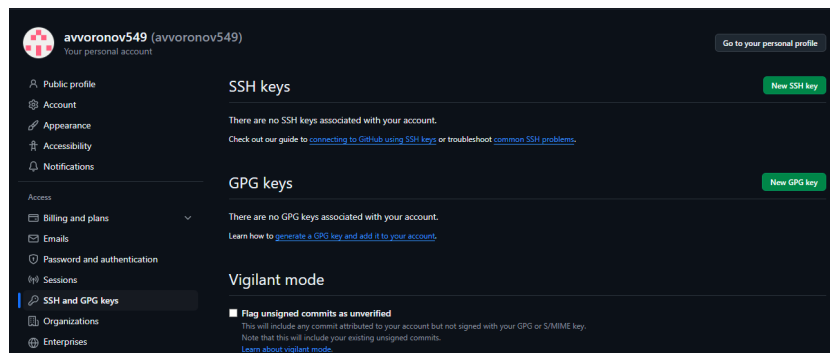


Рис. 3.8: Вставка SSH-ключа

##Создание рабочего пространства на основе шаблона

Открываю терминал и создаю репозиторий для предмета «Архитектура компьютеров». (рис. -fig. 3.9)

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

Рис. 3.9: Создание репозитория

##Создание репозитория курса на основе шаблона

Захожу на страницу репозитория с шаблоном курса, выбираю его в качестве своего нового. (рис. -fig. 3.10)

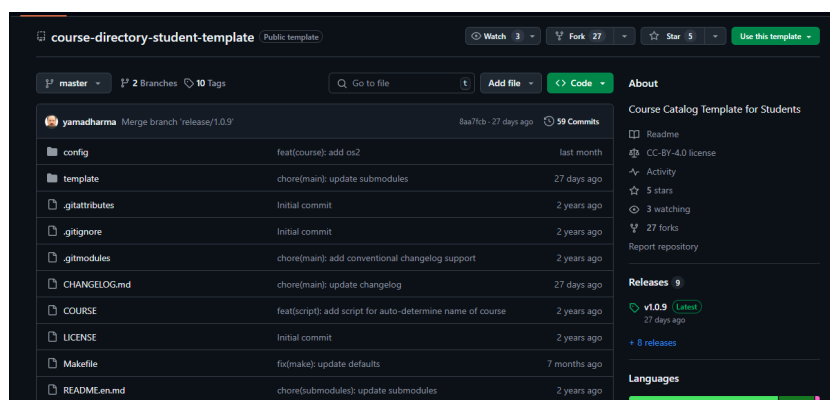


Рис. 3.10: Выбор шаблона

Затем создаю его, дав ему название. (рис. -fig. 3.11)

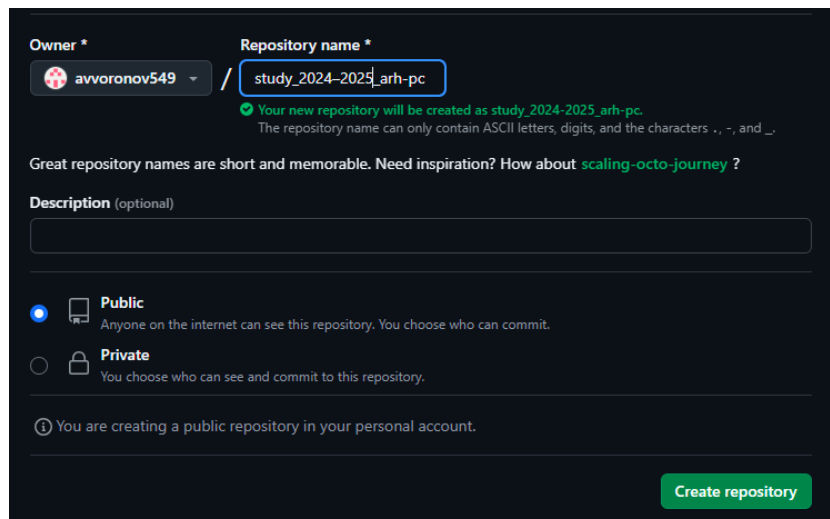


Рис. 3.11: Создание репозитория

Открываю терминал и перехожу в каталог курса. (рис. -fig. 3.12)

```
avvoronov@fedora: ~$ cd ~/work/study/2024-2025/"Архитектура компьютеров"
avvoronov@fedora: ~/work/study/2024-2025/Архитектура компьютеров$
```

Рис. 3.12: Переход в каталог курса

Клонирую созданный репозиторий (рис. -fig. 3.13)

```
avvoronov@fedora: ~/work/study/2024-2025/Архитектура компьютеров$ git clone --recursive git@github.com:avvoronov549/study_2024-2025_arh-pc.git arch-pc
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 | 1.71 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/avvoronov/work/study/2024-2025/Архитектура компьютеров/arch-pc/'...
```

Рис. 3.13: Клонирование репозитория

##Настройка каталога курса

Перехожу в каталог курса и удаляю лишние файлы (рис. -fig. ??)

[Переход в каталог курса] (image/017.PNG){#fig:014 width=70%}

(рис. -fig. 3.14)

```
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ rm package.json
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$
```

Рис. 3.14: Удаление лишних файлов в каталоге курса

Создаю необходимые каталоги, отправляю файлы на сервер (рис. -fig. 3.15)

```
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ make prepare
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ git add .
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ git commit -m 'feat(main): make course structure'
[master d37c057] 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
```

Рис. 3.15: Создание нужных каталогов и отправка на сервер

В репозитории проверяю результат выполненной работы. (рис. -fig. 3.16)

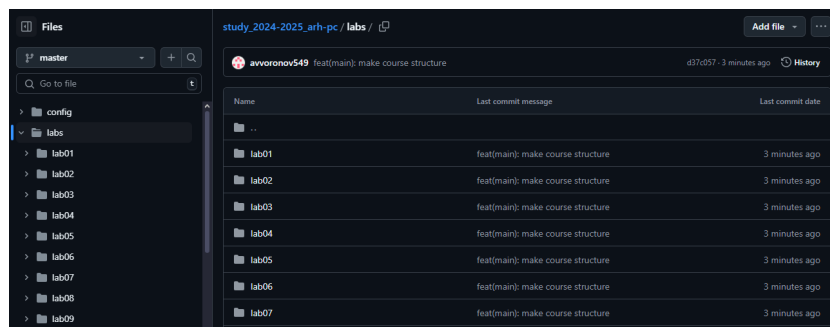


Рис. 3.16: Проверка проделанной работы

##Выполнение заданий для самостоятельной работы

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch, после чего перехожу в директорию labs/lab01/report (рис. -fig. 3.17)

```
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ cd ~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab02/report
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab02/report$ touch Л02_Воронов_А_отчет
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab02/report$ ls
bib image Makefile pandoc report.md Л02_Воронов_А_отчет
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab02/report$
```

Рис. 3.17: Создание отчета

(рис. -fig. 3.18)

```
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs$ cd lab01/report
```

Рис. 3.18: Переход в директорию labs/lab01/report

Проверяю местонахождение файла с отчетом по первой лабораторной работой. Он должен быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls. (рис. -fig. 3.19)

```
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ ls ~/Downloads
Л01_Воронов_отчет.pdf
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$
```

Рис. 3.19: Проверка расположения файла

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls. (рис. -fig. 3.20)

```
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ cp ~/Downloads/Л01_Воронов_отчет.pdf /home/avvoronov/work/study/2024-2025/"Архитектура компьютеров"/arch-pc/labs/lab01/report
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Л01_Воронов_отчет.pdf Л01_Воронов_А_отчет
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$
```

Рис. 3.20: Копирование файла

После чего добавляю файл в add Сохраняю изменения на сервере командой git commit -m "...", поясняя, что добавил файлы (рис. -fig. 3.21)

```

avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git add Л01_Воронов_отчет.pdf
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git commit -m "Add lab01/report"
[master c09444a] Add lab01/report
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Воронов_отчет.pdf
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      Л01_Воронов_А_отчет
      ../../lab02/report/Л02_Воронов_А_отчет

nothing added to commit but untracked files present (use "git add" to track)
avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$

```

Рис. 3.21: Комментарии к изменениям

Аналогично я поступаю с отчётом по этой лабораторной работе, после чего отправляю коммит на сервер.(рис. -fig. 3.22)

```

avvoronov@fedora:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 953.50 KiB | 5.64 MiB/s, done.
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:avvoronov549/study_2024-2025_arh-pc.git
   d37c057..c09444a  master -> master

```

Рис. 3.22: Отправление на сервер

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. -fig. 3.23)

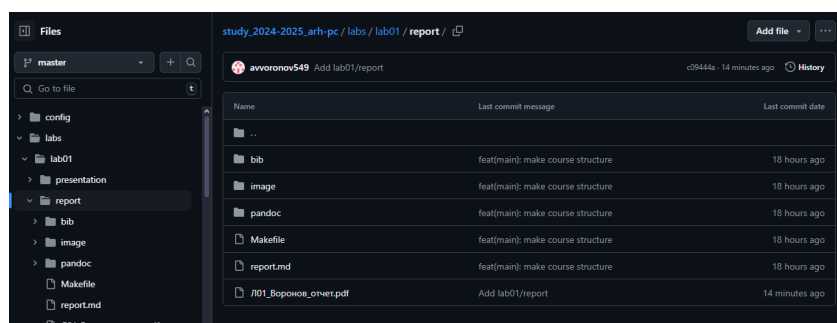


Рис. 3.23: Проверка на нахождения файла

4 Выводы

В заключение хочется отметить, что данная лабораторная работа позволила мне научиться работать с системой Git. Я практиковал свои навыки в работе с командной строкой, теперь уже связывая выполнимое с директориями GitHub.

5 Список литературы

1. Архитектура ЭВМ https://esystem.rudn.ru/pluginfile.php/2089082/mod_resource/content/0/Л.Бораторная%20работа%20№2.%20Система%20контроля%20версий%20Git.pdf
2. 30 команд Git, необходимых для освоения интерфейса командной строки Git / Хабр <https://habr.com/ru/companies/ruvds/articles/599929/>
3. Система контроля версий: определение, функции, популярные решения <https://gb.ru/blog/sistema-kontrolya-versij/>
4. Мой Github: https://github.com/avvoronov549/study_2024-2025_arh-pc