

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютера

Студент: Курбанов Рахман

Группа:НКАбд-06-24

МОСКВА

2024г.

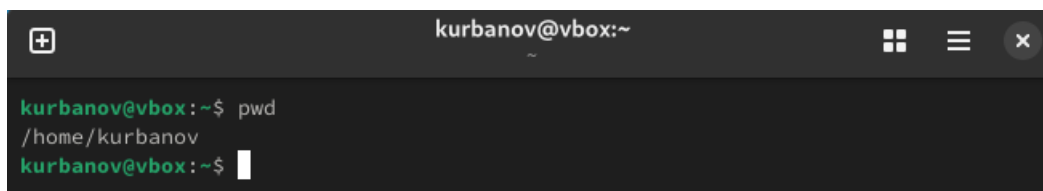
1.1 Цель лабораторной работы:

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

1.2 выполнения заданий лабораторной работы и результаты:

1.4.1. Создание учетной записи на GitHub:

Описание задания: Для начала необходимо создать учетную запись на сайте **GitHub**. Это позволит получить доступ к управлению репозиториями для работы над проектами.



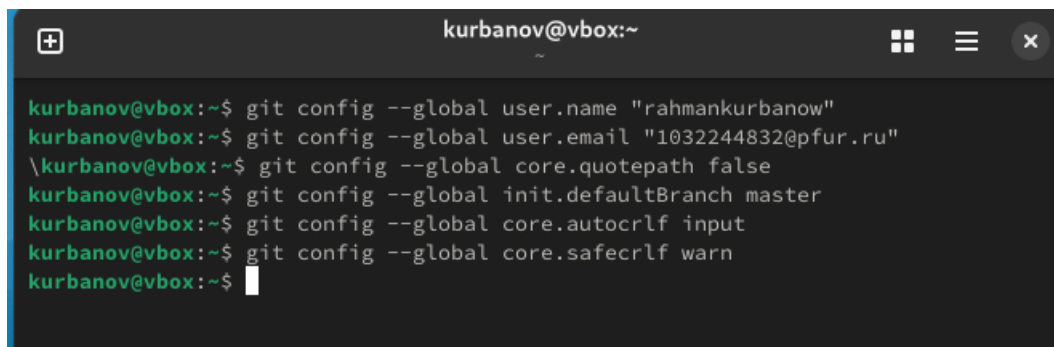
```
kurbanov@vbox:~  
kurbanov@vbox:~$ pwd  
/home/kurbanov  
kurbanov@vbox:~$
```

в моем случае учетная запись уже существует, поэтому этот шаг был пропущен.

1.4.2. Базовая настройка Git:

1. Указание имени пользователя и email для создания коммитов в репозиториях.

Эти данные будут добавляться к каждому коммиту, чтобы идентифицировать автора изменений.:



```
kurbanov@vbox:~  
kurbanov@vbox:~$ git config --global user.name "rahmankurbanow"  
kurbanov@vbox:~$ git config --global user.email "1032244832@pfur.ru"  
\kurbanov@vbox:~$ git config --global core.quotepath false  
kurbanov@vbox:~$ git config --global init.defaultBranch master  
kurbanov@vbox:~$ git config --global core.autocrlf input  
kurbanov@vbox:~$ git config --global core.safecrlf warn  
kurbanov@vbox:~$
```

Эта последовательность команд настраивает имя пользователя, email, кодировку UTF8, задает начальную ветку master, устанавливает обработку концов строк (autocrlf), и включает предупреждения о несоответствиях концов строк (safecrlf). Выполнение всех команд в одной строке позволяет быстрее закончить настройку и сразу перейти к работе с Git. Все параметры будут применены глобально для всех будущих репозиториев

1.4.3. Создание SSH-ключа:

Описание выполняемого задания:

Для безопасного подключения к репозиториям на GitHub, нужно сгенерировать SSH-ключи (публичный и приватный). Это позволит вам работать с репозиториями, не вводя каждый раз логин и пароль.

1. Генерация SSH-ключа:

```
kurbanov@vbox:~$ ssh-keygen -C "Kurbanow Rahman 1032244832@pfur.ru"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/kurbanov/.ssh/id_ed25519):
Created directory '/home/kurbanov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kurbanov/.ssh/id_ed25519
Your public key has been saved in /home/kurbanov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:+29LdqFEG9uPvRrU+N5PLy6+NovFC2pWpGnW1lf705M Kurbanow Rahman 1032244832@pfur.ru
The key's randomart image is:
+--[ED25519 256]--+
|
|
|      o
|    .. =o .
|   S= .+ooo.
|  =.+oo.o*
| o.o. *oo.B
| oo *=o+E*
```

Каталог `~/.ssh/` — это стандартное место для хранения SSH-ключей. Не изменяйте путь, если вы не хотите использовать другое место для хранения.

Копирование публичного ключа в буфер обмена:

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

Команда `cat ~/.ssh/id_rsa.pub` выводит содержимое публичного ключа, а команда `xclip -sel clip` копирует это содержимое в буфер обмена

Add new SSH Key

Title

SSH for Архитектура лабы

Key type

Authentication Key ▾

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKxyu6hFurtX04x8B04NiAzRpC4j2rorKhzFW48dsDR  
Kurbanow Rahman 1032244832@pfur.ru
```

Add SSH key

вставить ключ на сайт GitHub.

1.4.4. Создание рабочего пространства и репозитория курса на основе шаблона:

Для правильной организации рабочих файлов и проектов в рамках курса необходимо создать рабочее пространство по определенной структуре. В этом шаге будет выполнено создание директории для предмета «Архитектура компьютера», а также будет продемонстрировано, как структурировать папки для лабораторных работ.

Описание выполняемого задания:


Создание структуры рабочего пространства :

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

Эта команда создаст нужные каталоги по иерархии. Опция -p создает промежуточные каталоги, если они еще не существуют

1.4.5. Создание репозитория курса на основе шаблона:

Owner *

 A1zen666 ▾


 /

study_2024-2024_arch-pc


✔ study_2024-2024_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about [supreme-engine](#) ?


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

Эти действия создают новый репозиторий на основе предоставленного шаблона, который содержит структуру и необходимые файлы для работы по курсу.


Клонирование репозитория на локальный компьютер :



```
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера$ git clone --recursive git@github.com:A1zen666/study_2024-2025_arch-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 | 18.81 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/kurbanov/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 | 3.10 MiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/kurbanov/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 | 954.00 KiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера$
```



Команда `git clone --recursive` позволяет загрузить все файлы из удаленного репозитория в папку `arch-pc`.


 **study_2024-2025_arch-pc** Public


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














 master ▾

 Code ▾

 **A1zen666** Initial commit 8897d75 · 40 minutes ago 🕒 1 Commit

 config	Initial commit	40 minutes ago
 template	Initial commit	40 minutes ago
 .gitattributes	Initial commit	40 minutes ago
 .gitignore	Initial commit	40 minutes ago
 .gitmodules	Initial commit	40 minutes ago
 CHANGELOG.md	Initial commit	40 minutes ago
 COURSE	Initial commit	40 minutes ago
 LICENSE	Initial commit	40 minutes ago
 Makefile	Initial commit	40 minutes ago
 README.en.md	Initial commit	40 minutes ago
 README.git-flow.md	Initial commit	40 minutes ago
 README.md	Initial commit	40 minutes ago
 package.json	Initial commit	40 minutes ago

Настройка каталога курса :

```
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера$ cd /home/kurbanov/work/study/2024-2025/'архитектура компьютера'/arch-pc
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$
```

Удаление ненужного файла package.json :

```
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ rm package.json
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  Makefile  README.git-flow.md  template
config        LICENSE  README.en.md  README.md
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$
```

Удаление лишних файлов помогает избежать путаницы и оставить только необходимые для курса файлы.

Создание файла COURSE с названием курса:

```
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ echo arch-pc > COURSE
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ make prepare
make: 'prepare' is up to date.
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE      Makefile      README.en.md  template
config         labs       prepare      README.git-flow.md
course         LICENSE    presentation  README.md
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ cd labs
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc/labs$ ls
lab01 lab03 lab05 lab07 lab09 lab11 README.ru.md
lab02 lab04 lab06 lab08 lab10 README.md
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc/labs$
```

Отправка изменений на сервер :

```
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ git add .
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$ git commit -m 'feat(main): make course structure'
[master 6925f34] feat(main): make course structure
224 files changed, 53682 insertions(+), 14 deletions(-)
```

```
kurbanov@vbox:~/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 13 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.41 KiB | 968.00 KiB/s, done.
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:A1zen666/study_2024-2025_arch-pc.git
8897d75..6925f34 master -> master
kurbanov@vbox:~/work/study/2024-2025/архитектура компьютера/arch-pc$
```











Эти команды добавляют изменения в локальный репозиторий, создают коммит с описанием и отправляют изменения в удаленный репозиторий на GitHub.

1.5 Описание результатов выполнения заданий для самостоятельной работы:

1. Создание отчета в соответствующем каталоге рабочего пространства:

Описание задания:

Данное задание включает в себя создание отчета о выполнении лабораторной работы, копирование предыдущих отчетов и загрузку файлов на GitHub.

 A1zen666 Add files via upload a859230 · 5 minutes ago  History		
Name	Last commit message	Last commit date
 ..		
 bib	feat(main): make course structure	26 minutes ago
 image	feat(main): make course structure	26 minutes ago
 pandoc	feat(main): make course structure	26 minutes ago
 Makefile	feat(main): make course structure	26 minutes ago
 report.md	feat(main): make course structure	26 minutes ago
 ЛО1_Курбанов Рахман_отчет....	Add files via upload	5 minutes ago
 ЛО1_Курбанов Рахман_отчет....	Add files via upload	6 minutes ago

1.7 Вопросы для самопроверки:

1. **Что такое системы контроля версий (VCS) и для чего они предназначены?**

Системы контроля версий (VCS) — это инструменты для управления изменениями в коде и файлах проекта, позволяющие отслеживать историю изменений, работать с разными версиями и сотрудничать в команде.

2. **Объясните понятия VCS: хранилище, commit, история, рабочая копия**

- Хранилище (repository) место, где хранятся все версии проекта.
- Commit зафиксированные изменения с комментарием, представляющие версию проекта.
- История последовательность всех commit
- Рабочая копия (working copy) локальная версия проекта, с которой работает разработчик.

3. Чем отличаются централизованные и децентрализованные VCS? Примеры.

- Централизованные VCS (например, SVN): одно центральное хранилище, доступное для всех.
- Децентрализованные VCS (например, Git): каждый пользователь имеет своё полное хранилище, с возможностью синхронизации

4. Действия с VCS при единоличной работе с хранилищем.

- Инициализация репозитория.
- Добавление файлов и создание commit
- Проверка состояния (git status), просмотр истории (git log).

5. Порядок работы с общим хранилищем VCS.

- Клонирование репозитория.
- Создание ветки, внесение изменений.
- Commit изменений.
- Слияние с основной веткой через pull request.

6. Основные задачи, решаемые Git.

Управление версиями, создание веток, слияние изменений, разрешение конфликтов, работа с удалёнными репозиториями.

7. Команды Git: краткая характеристика.

- `git init`: создание репозитория.
- `git clone`: клонирование удалённого репозитория.
- `git add`: добавление файлов для отслеживания.
- `git commit`: фиксация изменений.
- `git push`: отправка изменений в удалённый репозиторий.
- `git pull`: получение изменений из удалённого репозитория.

8. Примеры работы с локальными и удалёнными репозиториями.

- Локальные : инициализация репозитория, создание commit , проверка состояния.
- Удалённые: клонирование, отправка изменений (git push), получение обновлений (git pull).

Комментарии и выводы по результатам выполнения заданий:

Выполнение задания помогло закрепить навыки работы с системами контроля версий, организацией файлов в рабочем пространстве, а также загрузкой проектов на GitHub. Эти навыки важны для поддержания эффективной командной работы, обеспечения безопасности данных и удобства доступа к прошлым версиям проектов.

