

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: Горбунова Яна Сергеевна

Группа: НКАбд-02-24

МОСКВА

2024 г.

Содержание

1. Цель работы...
2. Теоретическое введение...
3. Задание...
4. Задание для самостоятельной работы...
5. Выводы...
6. Список литературы...

Цель работы

Целью работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git

Теоретическое введение

1. Системы контроля версий. Общие понятия

Системы контроля версий (VCS) используются для совместной работы над проектами. Обычно основное дерево проекта хранится в локальном или удалённом репозитории с доступом для участников. VCS фиксирует изменения, объединяет их и позволяет откатываться к предыдущим версиям.

В классических системах применяется централизованная модель с единственным репозиторием, где сервер управляет версиями. Пользователь получает нужную версию файлов, вносит изменения и загружает новую версию, при этом старые версии сохраняются. Сервер может использовать дельта-компрессию, сохраняя только изменения между версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий предлагают гибкие функции, такие как поддержка нескольких версий одного файла с сохранением общей истории изменений и индивидуальных ветвей. Они фиксируют информацию о том, кто и когда вносил изменения, что хранится в журнале, доступ к которому можно ограничить. В распределённых системах центральный репозиторий не обязателен. Из классических VCS известны CVS и Subversion, а среди распределённых — Git, Bazaar и Mercurial. Принципы их работы схожи, различия в синтаксисе команд.

2. Система контроля версий Git

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.

3. Основные команды git

Наиболее часто используемые команды git представлены в таблице

Команда	Описание
<code>git init</code>	создание основного дерева репозитория
<code>git pull</code>	получение обновлений (изменений) текущего дерева из центрального репозитория
<code>git push</code>	отправка всех произведённых изменений локального дерева в центральный репозиторий
<code>git status</code>	просмотр списка изменённых файлов в текущей директории
<code>git diff</code>	просмотр текущих изменения
<code>git add .</code>	добавить все изменённые и/или созданные файлы и/или каталоги
<code>git add имена_файлов</code>	добавить конкретные изменённые и/или созданные файлы и/или каталоги
<code>git rm имена_файлов</code>	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
<code>git commit -am 'Описание коммита'</code>	сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b имя_ветки</code>	создание новой ветки, базирующейся на текущей
<code>git checkout имя_ветки</code>	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin имя_ветки</code>	отправка изменений конкретной ветки в центральный репозиторий
<code>git merge --no-ff имя_ветки</code>	слияние ветки с текущим деревом
<code>git branch -d имя_ветки</code>	удаление локальной уже слитой с основным деревом ветки
<code>git branch -D имя_ветки</code>	принудительное удаление локальной ветки
<code>git push origin :имя_ветки</code>	удаление ветки с центрального репозитория

Задание

1. Базовая настройка git

Существует несколько доступных серверов репозитория с возможностью бесплатного размещения данных. Для выполнения лабораторных работ я буду использовать Github.

Создаем учётную запись на сайте <https://github.com/> и заполняем основные данные (рис.1)

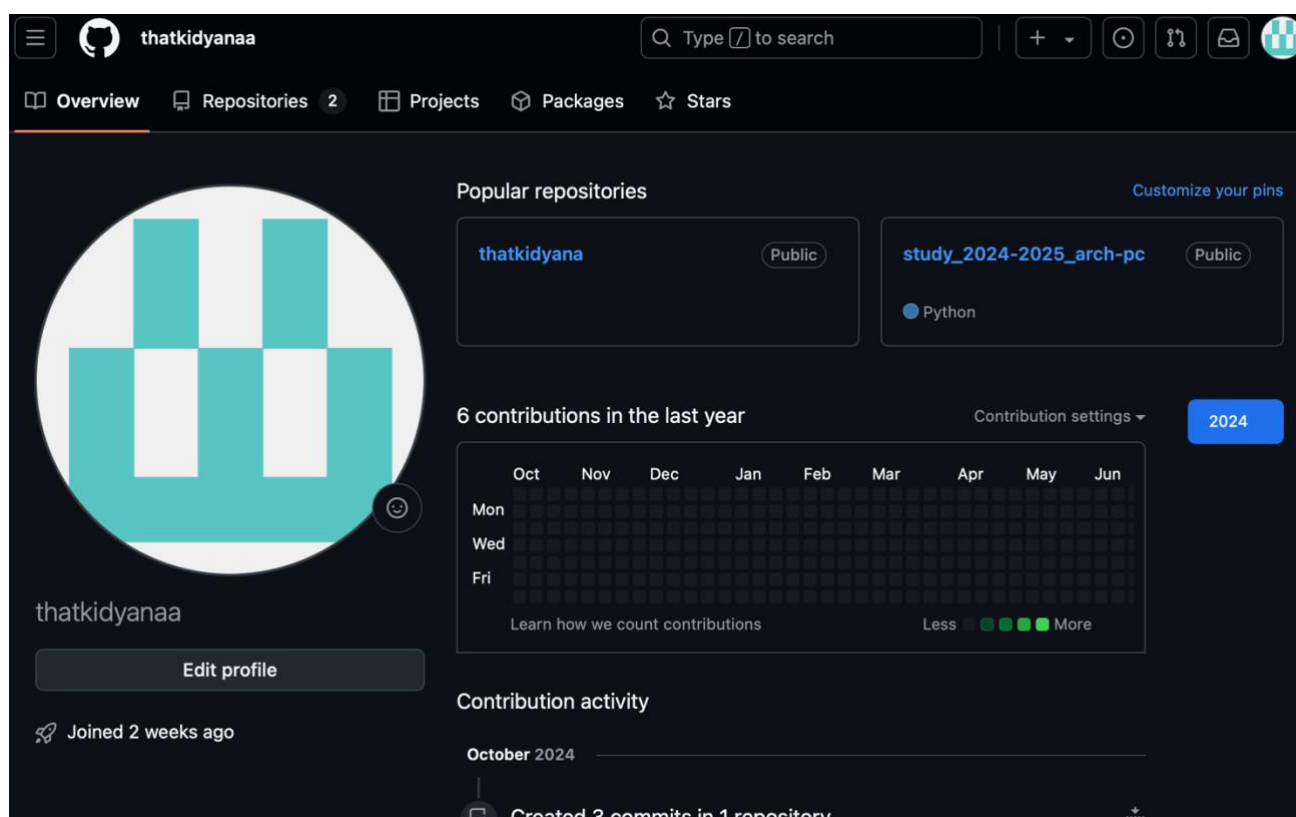


рис.1

Делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав имя и email владельца репозитория (рис.2)

```
ysgorbunova@dk1n22 ~ $ git config --global user.name "<Yana Gorbunova>"
ysgorbunova@dk1n22 ~ $ git config --global user.email "<flowjely@gmail.com>"
ysgorbunova@dk1n22 ~ $ git config --global core.quotepath false
ysgorbunova@dk1n22 ~ $ git config --global init.defaultBranch master
ysgorbunova@dk1n22 ~ $ git config --global core.autocrlf input
ysgorbunova@dk1n22 ~ $ git config --global core.safecrlf warn
```

рис.2

Настроим utf-8 в выводе сообщений git (рис.2)

Задаю имя начальной ветки (будем называть её master) (рис.2)

Параметр autocrlf и safecrlf (рис.2)

2. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый) (рис.3)

```
ysgorbunova@dk1n22 ~ $ ssh-keygen -C "Yana Gorbunova <flowjely@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/y/s/ysgorbunova/.ssh/id_ed25519):
/afs/.dk.sci.pfu.edu.ru/home/y/s/ysgorbunova/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/y/s/ysgorbunova/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/y/s/ysgorbunova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Ho5Y1V+aDpAXhoNrVpJHwsEA0EMgbttqyCEN5z/a59M Yana Gorbunova <flowjely@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
| . o=o..*+oo      |
| .. o +o*+ .      |
| .o. ..=+.o .      |
| .=o +. o . +      |
| .oo. o. S . +      |
| + o. o + . o      |
| .+ + ..o .      |
| . o .o E          |
| . .o..           |
+-----[SHA256]-----+
```

рис.3

Ключи сохранятся в каталоге ~/.ssh/.

Далее необходимо загрузить сгенерённый открытый ключ. Для этого захожу на сайт <http://github.org/> под своей учётной записью и переходим в меню Settings. После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key, скопировав

из локальной консоли ключ в буфер обмена и вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title) (рис.4)

```
ysgorbunova@dk1n22 ~ $ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
```

Рис.4

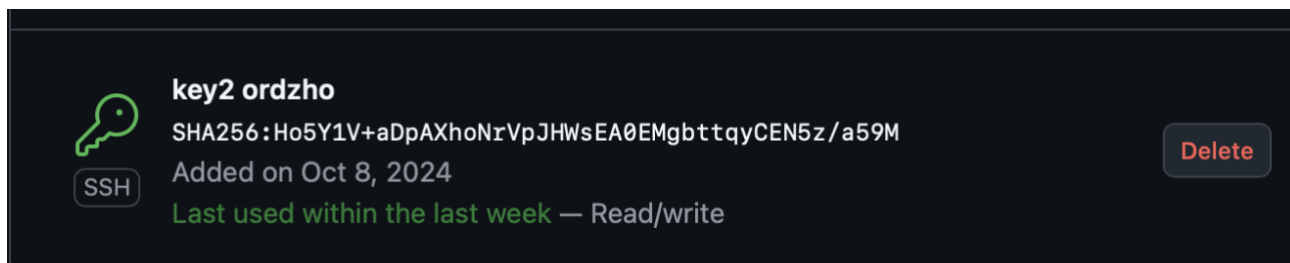


рис.5

SSH-ключ отразился в меню github (рис.5)

3. Создание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Рабочее пространство по предмету располагается в следующей иерархии:

~/work/study/

└── <учебный год>/

└── <название предмета>/

└── <код предмета>/

Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) структура каталогов примет следующий вид: ~/work/study/

└── 2023–2024/

└── Архитектура компьютера/

└── arch-pc/

└── labs/

└── lab01/

└── lab02/ ───

lab03...

- Каталог для лабораторных работ имеет вид labs.
- Каталоги для лабораторных работ имеют вид lab<номер>, например: lab01, lab02 и т.д.

Название проекта на хостинге git имеет вид: study_<учебный год>_<код предмета>

Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) название проекта примет следующий вид: study_2023–2024_arch-pc

Открываем терминал и создаем каталог для предмета «Архитектура

компьютера»:mkdir -p ~/work/study/2024-2025/"Архитектура компьютера" (рис.6)

```
ysgorbunova@dk1n22 ~ $ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
ysgorbunova@dk1n22 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
ysgorbunova@dk1n22 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
```

рис.6

Репозиторий на основе шаблона можно создать через web-интерфейс github.

Переходим на страницу репозитория с шаблоном курса (рис.10)

<https://github.com/yamadharma/course-directory-student-template>

<> Code Issues Pull requests Actions Projects Security Insights

cour... Public template Watch 3 Fork 28 Star 5 Use this template

ma... Go to file + <> Code About

yamadharma Merge branch 'relea... 8aa7fcb · last month

config	feat(course): ad...	last month
template	chore(main): up...	last month
.gitattributes	Initial commit	2 years ago
.gitignore	Initial commit	2 years ago
.gitmodules	chore(main): ad...	2 years ago
CHANGELOG.md	chore(main): up...	last month
COURSE	feat(script): add ...	2 years ago
LICENSE	Initial commit	2 years ago
Makefile	fix(make): updat...	7 months ago

Course Catalog Template for Students

Readme CC-BY-4.0 license Activity 5 stars 3 watching 28 forks Report repository

Releases 9

v1.0.9 Latest last month

+ 8 releases

рис.7 Далее выбираем Use this template.

В открывшемся окне задаем имя репозитория (Repository name) study_2024–2025_arch-pc и создаем репозиторий (кнопка Create repository from template).

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 *

 sdsamoylova ▾

Repository name *

/ study_2024-2025_arch-pc

✔ Your new repository will be created as study_2024-2025_arch-pc.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about **friendly-spoon** ?

Description (optional)

рис.8

Открываем терминал, переходим в каталог курса и клонируем созданный репозиторий (рис.9)

```
ysgorbunova@dk1n22 ~ $ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
ysgorbunova@dk1n22 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
ysgorbunova@dk1n22 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:thatkiyanaa/study_2024-2025_arch-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:+DiY3wvV6TuJJhpZisF/zLDA0zPMSvHdkr4UvC0qU.
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: 33, done.
remote: Counting objects: 100% (33/33) done
```

рис.9

4. Настройка каталога курса

Переходим в каталог курса и удаляем лишние файлы (рис.15)

```
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера $ cd arch-pc/
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ ls
CHANGELOG.md  config  COURSE  LICENSE  Makefile  README.en.md  README.git-flow.md  README.md  template
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo arch-pc > COURSE
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make
```

рис.10

Создаем необходимые каталоги (рис.10)

```
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make prepare
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config        labs    Makefile  presentation  README.git-flow.md  template
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .
ysgorbunova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am 'feat(main):
make course structure'
[master ce81f99] 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
```

Отправляем файл на сервер (рис.11)

рис.11

Проверяем правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.

Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	8 minutes ago
lab02	feat(main): make course structure	8 minutes ago
lab03	feat(main): make course structure	8 minutes ago
lab04	feat(main): make course structure	8 minutes ago
lab05	feat(main): make course structure	8 minutes ago
lab06	feat(main): make course structure	8 minutes ago
lab07	feat(main): make course structure	8 minutes ago
lab08	feat(main): make course structure	8 minutes ago
lab09	feat(main): make course structure	8 minutes ago
lab10	feat(main): make course structure	8 minutes ago
lab11	feat(main): make course structure	8 minutes ago
README.md	feat(main): make course structure	8 minutes ago
README.ru.md	feat(main): make course structure	8 minutes ago

рис.12

Задание 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. Я добавила отчет по предыдущей лабораторной работе в соответствующий каталог (рис.13)









Name	Last commit message	Last commit date
 ..		
 bib	feat(main): make course structure	2 hours ago
 image	feat(main): make course structure	2 hours ago
 pandoc	feat(main): make course structure	2 hours ago
 Makefile	feat(main): make course structure	2 hours ago
 report.md	feat(main): make course structure	2 hours ago
 Л01_Самойлова_отчет.doc	Add files via upload	now
 Л01_Самойлова_отчет.pdf	Add files via upload	now

рис.13

Выводы

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

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnightcommander.org/>.
4. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
5. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
6. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
7. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.