

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технологический университет «СТАНКИН»

(ФГБОУ ВО МГТУ «СТАНКИН»)

Институт цифровых интеллектуальных систем

Кафедра «Компьютерные системы управления»

Образовательная программа 15.03.04

«Автоматизация технологических процессов и производств»

Дисциплина «Основы системного программного обеспечения»

Отчёт по лабораторной работе №1

«Работа с системами контроля версий на примере GitHub»

|  |
| --- |
|  |
| (дата) |

|  |
| --- |
|  |
| (подпись) |

Выполнил: Рудометов Н.В.

студент группы: АДБ 21-07

|  |
| --- |
|  |
| (дата) |

|  |
| --- |
|  |
| (подпись) |

Проверил: Ковалев И.А.

к.т.н., доцент

Москва 2024 г.

Оглавление

[***Цель работы:*** 3](#_Toc168510306)

[***Краткая теория:*** 3](#_Toc168510307)

[Основные задачи работы: 5](#_Toc168510308)

[Ход работы: 5](#_Toc168510309)

[Вывод: 22](#_Toc168510310)

[Список использованных литературы: 23](#_Toc168510311)

***Лабораторная работа №1***

**«Работа с системами контроля версий на примере GitHub»**

***Цель работы:***

Освоение навыков работы с системами контроля версий на примере GitHub.

***Краткая теория:***

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов. Для примеров будем использовать исходные коды программ, но на самом деле под версионный контроль можно поместить файлы практически любого типа.

Главное отличие Git'а от любых других СКВ (например, Subversion и ей подобных) — это то, как Git смотрит на свои данные. В принципе, большинство других систем хранит информацию как список изменений (патчей) для файлов. Эти системы (CVS, Subversion, Perforce, Bazaar и другие) относятся к хранимым данным как к набору файлов и изменений, сделанных для каждого из этих файлов во времени. Git считает хранимые данные набором слепков небольшой файловой системы. Каждый раз, когда вы фиксируете текущую версию проекта, Git, по сути, сохраняет слепок того, как выглядят все файлы проекта на текущий момент. Ради эффективности, если файл не менялся, Git не сохраняет файл снова, а делает ссылку на ранее сохранённый файл.

Это важное отличие Git'а от практически всех других систем контроля версий. Из-за него Git вынужден пересмотреть практически все аспекты контроля версий, которые другие системы переняли от своих предшественниц. Git больше похож на небольшую файловую систему с невероятно мощными инструментами, работающими поверх неё, чем на просто СКВ.

Для совершения большинства операций в Git'е необходимы только локальные файлы и ресурсы, т.е. обычно информация с других компьютеров в сети не нужна. Если вы пользовались централизованными системами, где практически на каждую операцию накладывается сетевая задержка, вы, возможно, подумаете, что боги наделили Git неземной силой. Поскольку вся история проекта хранится локально у вас на диске, большинство операций кажутся практически мгновенными.

К примеру, чтобы показать историю проекта, Git'у не нужно скачивать её с сервера, он просто читает её прямо из вашего локального репозитория. Поэтому историю вы увидите практически мгновенно. Если вам нужно просмотреть изменения между текущей версией файла и версией, сделанной месяц назад, Git может взять файл месячной давности и вычислить разницу на месте, вместо того чтобы запрашивать разницу у СКВ-сервера или качать с него старую версию файла и делать локальное сравнение.

Перед сохранением любого файла Git вычисляет контрольную сумму, и она становится индексом этого файла. Поэтому невозможно изменить содержимое файла или каталога так, чтобы Git не узнал об этом. Эта функциональность встроена в сам фундамент Git'а и является важной составляющей его философии. Если информация потеряется при передаче или повредится на диске, Git всегда это выявит.

Практически все действия, которые вы совершаете в Git'е, только добавляют данные в базу. Очень сложно заставить систему удалить данные или сделать что-то неотменяемое. Можно, как и в любой другой СКВ, потерять данные, которые вы ещё не сохранили, но как только они зафиксированы, их очень сложно потерять, особенно если вы регулярно отправляете изменения в другой репозиторий.

# Основные задачи работы:

1. Создать учетную запись на github.com

2. Создать локальный репозиторий

3. Зафиксировать изменения в области заготовленных файлов

4. Переслать локальный коммит на сервер

5. Сделать слияние веток

6. Просмотреть изменения и разрешение конфликтов

7. Удалить ветку на сервере и вернуть к предыдущему состоянию

8. Исправить коммит.

# Ход работы:

1. Создаем учетную запись на github.com или заходим в уже имеющийся личный кабинет, как в нашем случае и создаем новый репозиторий.

2. Создали файлик test.txt и написал наши фамилии. Открыли командную строку «cmd» и перешли с помощью команды «cd» в созданный нами каталог. Настроим самые важные опции и параметры: наше пользователя и адрес электронной почты.

3. Переходим в созданную нами папку, используя команду cd. Теперь необходимо проинициализировать эту папку как git репозиторий. Вводим команду: git init. Командная строка возвращает нам сообщение, что проинициализирована пустая git директория. Теперь используя проводник windows создаем в своей папке текстовый файл и пишем в него наши имена. Вводим команду: git stаtus и нам отображается сообщение, что есть новый файл, но он не отслеживается. Чтобы исправить ситуацию используем команды add . и commit.

Microsoft Windows [Version 10.0.19045.4412]

(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Admin>cd C:\Users\Admin\Documents\Rudometovv

C:\Users\Admin\Documents\Rudometovv>git init

Initialized empty Git repository in C:/Users/Admin/Documents/Rudometovv/.git/

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

C:\Users\Admin\Documents\Rudometovv>git add

Nothing specified, nothing added.

hint: Maybe you wanted to say 'git add .'?

hint: Disable this message with "git config advice.addEmptyPathspec false"

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "first commit"

[master (root-commit) d4e0944] first commit

1 file changed, 1 insertion(+)

create mode 100644 test.txt

4. Чтобы связать наш локальный репозиторий с репозиторием на GitHub, выполним следующую команду в терминале

C:\Users\Admin\Documents\Rudometovv>git remote add origin <https://github.com/NikitosRudometov/Lab1>

C:\Users\Admin\Documents\Rudometovv>git remote -v

origin https://github.com/NikitosRudometov/Lab1 (fetch)

origin https://github.com/NikitosRudometov/Lab1 (push)

5. Запросим изменения с сервера

Мы создали изменения в нашем репозитории, создав файл readme и другие пользователи могут скачать изменения при помощи команды pull

C:\Users\Admin\Documents\Rudometovv>git pull origin master

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 856 bytes | 38.00 KiB/s, done.

From https://github.com/NikitosRudometov/Lab1

\* branch master -> FETCH\_HEAD

\* [new branch] master -> origin/master

Successfully rebased and updated refs/heads/master.

6. Теперь перешлем локальный коммит на сервер

C:\Users\Admin\Documents\Rudometovv>git push origin master

info: please complete authentication in your browser...

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 16 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 296 bytes | 148.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/NikitosRudometov/Lab1

485e220..3819981 master -> master

Посмотрим наши изменения с использованием команды

C:\Users\Admin\Documents\Rudometovv>git log

commit 381998182d5d31f920d3b59028bf24a9bf456c23 (HEAD -> master, origin/master)

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 02:58:32 2024 +0300

first commit

commit 485e2206edeecba91569480609bf825adfb7d7f6

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 02:47:14 2024 +0300

Initial commit

7. Создадим новую ветку second, посмотрим в какой находимся и перейдем на другую:

C:\Users\Admin\Documents\Rudometovv>git branch

\* master

second

C:\Users\Admin\Documents\Rudometovv>git checkout second

Switched to branch 'second'

8. Создаем новый файл в нашем локальном репозитории и пишем в нем свои фамилии, добавляем в область подготовленных файлов, коммитим и отправляем на сервер: так же сливаем ветки и удаляем second

C:\Users\Admin\Documents\Rudometovv>git add test2.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "second commit"

[second ec243c8] second commit

1 file changed, 1 insertion(+)

create mode 100644 test2.txt

C:\Users\Admin\Documents\Rudometovv>git push origin second

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Delta compression using up to 16 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (2/2), 277 bytes | 277.00 KiB/s, done.

Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'second' on GitHub by visiting:

remote: https://github.com/NikitosRudometov/Lab1/pull/new/second

remote:

To https://github.com/NikitosRudometov/Lab1

\* [new branch] second -> second

9. Слияние веток

C:\Users\Admin\Documents\Rudometovv>git checkout master

Switched to branch 'master'

C:\Users\Admin\Documents\Rudometovv>git merge second

Updating 3819981..ec243c8

Fast-forward

test2.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 test2.txt

C:\Users\Admin\Documents\Rudometovv>git branch -d second

Deleted branch second (was ec243c8).

C:\Users\Admin\Documents\Rudometovv>git branch newdev

C:\Users\Admin\Documents\Rudometovv>git checkout newdev

M test.txt

Switched to branch 'newdev'

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "third commit"

[newdev ffa6c5d] third commit

1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Admin\Documents\Rudometovv>git push origin newdev

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 16 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 328 bytes | 328.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'newdev' on GitHub by visiting:

remote: https://github.com/NikitosRudometov/Lab1/pull/new/newdev

remote:

To https://github.com/NikitosRudometov/Lab1

\* [new branch] newdev -> newdev

C:\Users\Admin\Documents\Rudometovv>git checkout master

Switched to branch 'master'

C:\Users\Admin\Documents\Rudometovv>git merge newdev

Updating ec243c8..ffa6c5d

Fast-forward

test.txt | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Admin\Documents\Rudometovv>git checkout newdev

Switched to branch 'newdev'

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch newdev

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "4"

[newdev e7530d7] 4

1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Admin\Documents\Rudometovv>git push origin newdev

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 16 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/NikitosRudometov/Lab1

ffa6c5d..e7530d7 newdev -> newdev

C:\Users\Admin\Documents\Rudometovv>git checkout master

Switched to branch 'master'

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "5"

[master b44b4e0] 5

1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Admin\Documents\Rudometovv>git push origin master

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 16 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/NikitosRudometov/Lab1

3819981..b44b4e0 master -> master

10. Просмотр изменений и разрешение конфликтов

C:\Users\Admin\Documents\Rudometovv>git merge newdev

Auto-merging test.txt

CONFLICT (content): Merge conflict in test.txt

Automatic merge failed; fix conflicts and then commit the result.

C:\Users\Admin\Documents\Rudometovv>git diff

diff --cc test.txt

index 26969f5,fba77b3..0000000

--- a/test.txt

+++ b/test.txt

@@@ -1,1 -1,1 +1,5 @@@

++<<<<<<< HEAD

+Nikita Valerievichhhh

++=======

+ Nikita Valerievi

++>>>>>>> newdev

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

All conflicts fixed but you are still merging.

(use "git commit" to conclude merge)

Changes to be committed:

modified: test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "test"

[master 3d81049] test

C:\Users\Admin\Documents\Rudometovv>git push origin master

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 16 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 375 bytes | 375.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/NikitosRudometov/Lab1

b44b4e0..3d81049 master -> master

11.Удаление веток на сервере

Удаляем ветки с github:

C:\Users\Admin\Documents\Rudometovv>git branch -d newdev

Deleted branch newdev (was e7530d7).

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

nothing to commit, working tree clean

C:\Users\Admin\Documents\Rudometovv>git branch

\* master

C:\Users\Admin\Documents\Rudometovv>git push origin --delete newdev

To https://github.com/NikitosRudometov/Lab1

- [deleted] newdev

C:\Users\Admin\Documents\Rudometovv>git push origin --delete second

To https://github.com/NikitosRudometov/Lab1

- [deleted] second

1. Возврат к предыдущему состоянию

Чтобы посмотреть все комиты, можно использовать команду git log:

C:\Users\Admin\Documents\Rudometovv>git log

commit 3d810491ab74450d2107a6f1bed3428e8fb1948c (HEAD -> master, origin/master)

Merge: b44b4e0 e7530d7

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:40:00 2024 +0300

test

commit b44b4e04185143f4f71259469084a10165b95433

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:27:34 2024 +0300

5

commit e7530d72b4814a0df590062a3e7b648e22744ce6

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:23:58 2024 +0300

4

commit ffa6c5dcf3caf41aa28b43fb766fb71d1f01aaa2

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:11:04 2024 +0300

third commit

commit ec243c8c9ac31e71db2aeb7866ec7a83c653ee48

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:05:21 2024 +0300

Выбираем коммит и откатываемся:

C:\Users\Admin\Documents\Rudometovv>git checkout ec243c8c9ac31e71db2aeb7866ec7a83c653ee48

Note: switching to 'ec243c8c9ac31e71db2aeb7866ec7a83c653ee48'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -c with the switch command. Example:

git switch -c <new-branch-name>

Or undo this operation with:

git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at ec243c8 second commit

C:\Users\Admin\Documents\Rudometovv>git branch

\* (HEAD detached at ec243c8)

Master

1. Исправление коммита

C:\Users\Admin\Documents\Rudometovv>git checkout master

Previous HEAD position was ec243c8 second commit

Switched to branch 'master'

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "test"

[master 7fe88e0] test

1 file changed, 1 insertion(+), 3 deletions(-)

C:\Users\Admin\Documents\Rudometovv>git push origin master

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 16 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 329 bytes | 329.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/NikitosRudometov/Lab1

3d81049..7fe88e0 master -> master

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: test.txt

C:\Users\Admin\Documents\Rudometovv>git reset HEAD

Unstaged changes after reset:

M test.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: test.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Admin\Documents\Rudometovv>git add test.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "6"

[master 6e5171d] 6

1 file changed, 1 insertion(+)

C:\Users\Admin\Documents\Rudometovv>git reset HEAD

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

nothing to commit, working tree clean

C:\Users\Admin\Documents\Rudometovv>git add .

C:\Users\Admin\Documents\Rudometovv>git commit --amend

[master 112e795] 6

Date: Mon May 27 03:59:49 2024 +0300

1 file changed, 1 insertion(+)

C:\Users\Admin\Documents\Rudometovv>git log

commit 112e7958a1609f3751914433c90249d752c61826 (HEAD -> master)

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:59:49 2024 +0300

6

commit 7fe88e019412e0725cc178328c96c381228033c9 (origin/master)

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:54:47 2024 +0300

test

commit 3d810491ab74450d2107a6f1bed3428e8fb1948c

Merge: b44b4e0 e7530d7

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:40:00 2024 +0300

test

commit b44b4e04185143f4f71259469084a10165b95433

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:27:34 2024 +0300

5

commit e7530d72b4814a0df590062a3e7b648e22744ce6

Author: NikitosRudometov <nik.rudometov.03@mail.ru>

Date: Mon May 27 03:23:58 2024 +0300

1. Отправка только нужных файлов на сервер

Создаем вручную файл под названием «.gitignore» и сохраняем его в

директорию проекта. Внутри файла перечисляем названия файлов/папок,

которые нужно игнорировать, каждый с новой строки. Файл «.gitignore»

добавляем, коммитим и отправляем на сервер.

C:\Users\Admin\Documents\Rudometovv>git add .gitignore.txt

C:\Users\Admin\Documents\Rudometovv>git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore.txt

C:\Users\Admin\Documents\Rudometovv>git commit -m "ignore"

[master cba0e75] ignore

1 file changed, 1 insertion(+)

create mode 100644 .gitignore.txt

C:\Users\Admin\Documents\Rudometovv>git push origin master

Enumerating objects: 8, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 16 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (6/6), 563 bytes | 563.00 KiB/s, done.

Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/NikitosRudometov/Lab1

7fe88e0..cba0e75 master -> master

C:\Users\Admin\Documents\Rudometovv>

# Вывод:

В лабораторной работе №1 *«Работа с системами контроля версий на примере GitHub»* мы освоили навыки работы с системами контроля версий на примере GitHub.

# Список использованных литературы:

1. «HTML, XHTML and CSS», Andy Harris. Изд.: Машиностроение, 2012.
2. «JavaScript», Дэвид Макфарланд. Изд.: Символ-Плюс, 2013.
3. «Большая книга CSS3», Дэвид Макфарланд. Изд.: Питер, 2016.
4. Интернет-ресурсы.