

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2
Система контроля версий Git

РИДОЙ ХАЙАТУННАБИ

ГРУППА: НКАБД-05-2023

Цель работы: научиться управлять версиями проекта с помощью системы Git; освоить работу с репозиториями GitHub.

Ход выполнения работы

Устанавливаем git для Windows.

Производим инициализацию себя как пользователя.

```
git config --global user.name "Вашеимя"
```

```
git config --global user.email ваш@mail.ru
```

Создаем папку site6.local в корне локального сервера. Путь до рабочей папки выглядит следующим образом: C:/OpenServer/domains/site6.local.

Открываем git-консоль и перейдите в рабочую папку:

```
cd C:/OpenServer/domains/site6.local
```

```
MINGW64:/c/OpenServer/domains/site6.local
```

```
RIDOY+admin@RIDOY MINGW64 ~ (master)
```

```
$ cd C:/
```

```
RIDOY+admin@RIDOY MINGW64 /c
```

```
$ git config --global user.name Hayatunnabi
```

```
RIDOY+admin@RIDOY MINGW64 /c
```

```
$ git config --global user.email hayatunnabi@mail.com
```

```
RIDOY+admin@RIDOY MINGW64 /c
```

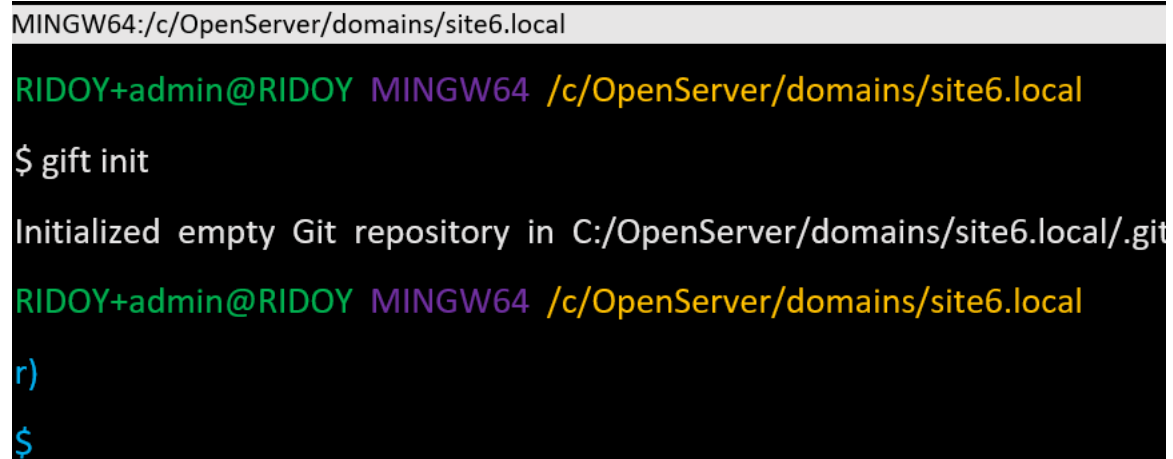
```
$ cd C:/OpenServer/domain/site6.local
```

```
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local
```

```
$ |
```

Рисунок 1 – Переход в рабочую папку

Производим инициализацию git-репозитория с помощью команды `git init`



```
MINGW64:/c/OpenServer/domains/site6.local
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local
$ git init
Initialized empty Git repository in C:/OpenServer/domains/site6.local/.git
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local
r)
$
```

Рисунок 2 – Инициализация репозитория

Создаем обычный текстовый файл `test1.txt` в указанной папке. Вводим в git-консоль команду:

`git status`

В ответ мы получим сообщение, представленное на рисунке 3.

```
MINGW64:/c/OpenServer/domains/site6.local
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git status
On branch master
NO commits yet

Untracked files:
  (use "git Add <file>..." to include in what will be committed)
    test1.txt

nothing added to commit but untracked files present (use "git add" to track)
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 3 – Git-статус

Пропишем в терминале git-команду добавления файла *test1.txt*:

git add.

Проверим статус командой:

git status

Теперь сообщение выглядит иначе (рисунок 4).

```
MINGW64:/c/OpenServer/domains/site6.local

RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git add .

RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git status

On branch master

NO commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   test1.txt

RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 4 – Измененный Git-статус

При всех последующих изменениях файла test1.txt каждый раз необходимо добавлять файл заново.

После добавления файла командой *git add* необходимо произвести фиксацию изменений командой:

```
git commit -m "сообщение"
```

Для просмотра истории коммитов достаточно ввести команду:

```
git log
```

В результате получаем ответ, представленный на рисунке 5.

```
MINGW64:/c/OpenServer/domains/site6.local

RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git commit -m "БНЕСЛИ ТАКИЕ-ТО КАКИЕ ИЗМЕНЕНИЯ"
[master (root-commit) 4ac3cf1] БНЕСЛИ ТАКИЕ-ТО КАКИЕ ИЗМЕНЕНИЯ
1 file changed, 3 insertions(+)
create mode 100064 test1.txt

RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git log
commit 4ac3cf1d4203b9a261091811daf205f75e59 (HEAD -> master)
Author: RIDOY ridoy@gmail.com
Date: Sun Sep 24 11:42:46 2023 +0300
    БНЕСЛИ ТАКИЕ-ТО КАКИЕ ИЗМЕНЕНИЯ
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ |
```

Рисунок 5 – Git-лог

С каждым последующим изменением в файле или добавлением новых мы должны запомнить последовательность действий:

git add (добавляем файлы) -> git commit -m "" (Фиксируем состояние).

Git reset – Откат изменений

В содержимое текстового файла test1.txt добавим: hello world!!!

После успешного сохранения документа вводим в консоль:

git add.

git commit -m "Написали привет мир!"

git log

```

MINGW64:/c:/OpenServer/domains/site6.local
RIDOY+admin@RIDOY MINGW64 /c:/OpenServer/domains/site6.local (master)
$ git add .

RIDOY+admin@RIDOY MINGW64 /c:/OpenServer/domains/site6.local (master)
$ git commit -m "НАПИСАЛИ ПРИБЕТ МИР"
[master ab5d886 ] НАПИСАЛИ ПРИБЕТ МИР
1 file changed, 3 insertions(+), 1 deletion(-)

RIDOY+admin@RIDOY MINGW64 /c:/OpenServer/domains/site6.local (master)
$ git log

commit 4ac3cf1d4203b9a261091811daf205f75e59 (HEAD -> master)
Author: RIDOY ridoy@gmail.com
Date: Sun Sep 24 11:42:46 2023 +0300

    НАПИСАЛИ ПРИБЕТ МИР

commit 4ac3cf1d4203b9a261091811daf205f75e59
Author: RIDOY ridoy@gmail.com
Date: Sun Sep 24 11:42:46 2023 +0300

    БНЕСЛИ ТАКИЕ-ТО КАКИЕ ИЗМЕНЕНИЯ

RIDOY+admin@RIDOY MINGW64 /c:/OpenServer/domains/site6.local (master)
$|

```

Рисунок 6 – Изменение документа

Скопируем полностью хэш самого первого состояния в формате:

4ac3cf1d4203b9a261091811aa86daf205f75e59.

Для отката изменений к конкретному коммиту выполним команду:

git reset --soft 4ac3cf1d4203b9a261091811aa86daf205f75e59

Откат изменений с ключом *--soft* не удаляет состояния, а лишь смещает индекс.

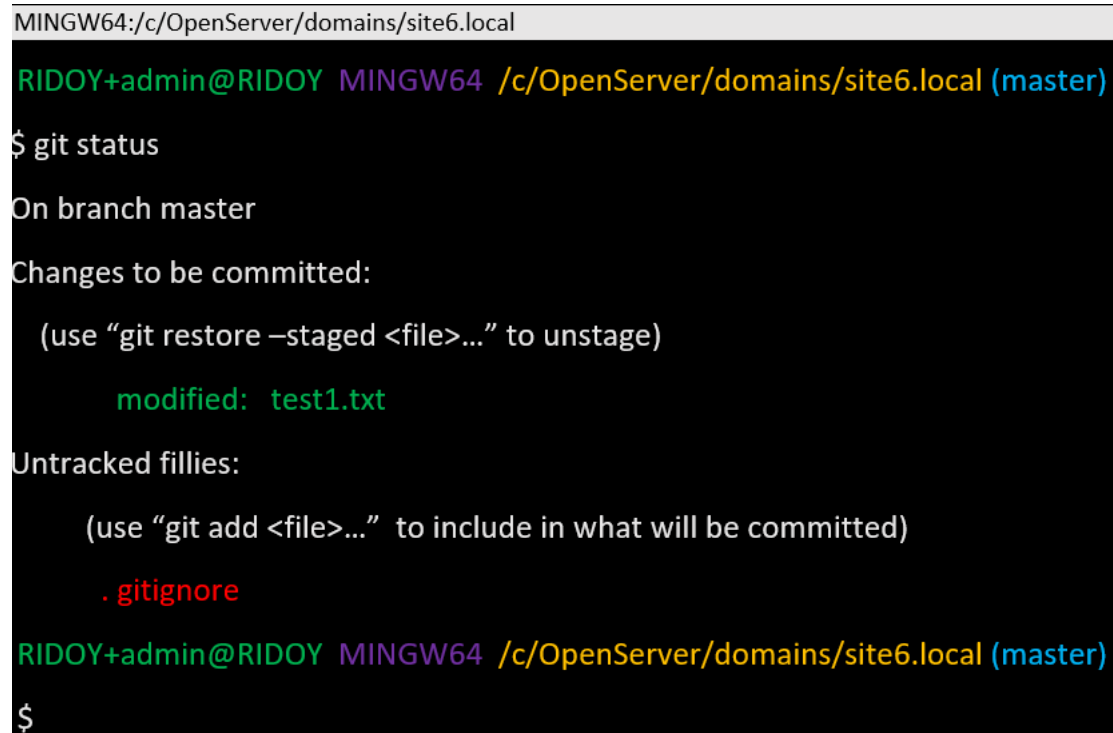
Откат изменений с ключом *--hard* удаляет полностью все состояния и индексы.

Gitignore – игнорирование файлов и папок

Создадим папку с названием `ignore_files`. Внутри папки создадим текстовый файл `file.txt`. Если по каким-либо причинам требуется игнорировать данную папку и не добавлять ее в `git`-репозиторий, необходимо в корне проекта создать `gitignore`-файл.

Если система ругается на создание файла с именем *.gitignore, создайте простой текстовый документ и пересохраните его с именем *.gitignore.

Откроем файл *.gitignore в [текстовом редакторе](#) и добавим название папки, которую хотим исключить: ignore_files (рисунок 7).



```
MINGW64:/c/OpenServer/domains/site6.local
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test1.txt
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 7 - Проверка статуса после выполнения команды gitignore
Все сделали правильно, в ответ не увидели папки которую исключили.

Работа с удаленными git-репозиториями

Перед подключением и отправкой данных в git-репозиторий, создадим учетную запись на github.com.

Создадим новый репозиторий на github.

1. Задаем имя.
2. Описание (по желанию).

3. Оставляем репозиторий в публичном доступе.

4. Создаем репозиторий.

5. Копируем ссылку на репозиторий.

Переходим в рабочую папку локального репозитория и вводим в консоль команду:

```
git remote add origin https://github.com/Maksion2020/git_lab2.git
```

Теперь локальный git-репозиторий успешно связан с удаленным.

Отправляем данные в удаленный git-репозиторий

После успешного подключения к репозиторию на GitHub отправляем данные командой:

```
git push origin master,
```

где

origin – имя репозитория по умолчанию;

master – ветка, в которую мы хотим отправить данные (по умолчанию master – главная ветка).

```
MINGW64:/c/OpenServer/domains/site6.local
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects :100% (2/2), done
Writing objects :100% (3/3), 324 bytes | 32.00 KiB/s, done
Total 3 (delta 0), reused 0 (delta 0), pack-refused 0
To https://github.com/RIDOY2023/git_lab2.git
    * [new branch]    master -> master
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 8 – Отправка данных на удаленный репозиторий

Принимаем данные из удаленного git-репозитория

Поскольку изменений в удаленный репозиторий никто не вносил, принимать нам нечего. Внесем правки в уже имеющийся файл, а затем введем команду:

git pull origin master

```
MINGW64:/c/OpenServer/domains/site6.local

RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git pull origin master
From https://github.com/RIDOY2023/git_lab2.git
    * branch      master      -> FETCH_HEAD
Auto-merging test1.txt
CONFLICT (content): Merge conflict in test1.txt
Automatic merge failed; fix conflicts and then commit the result.
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
(MERGING)
$
```

Рисунок 9 – прием данных из удаленного репозитория

Git clone – Клонирование репозитория

Для скачивания git-репозитория из GitHub достаточно ввести команду:

git clone https://github.com/Ridoy2020/git_lab2.git

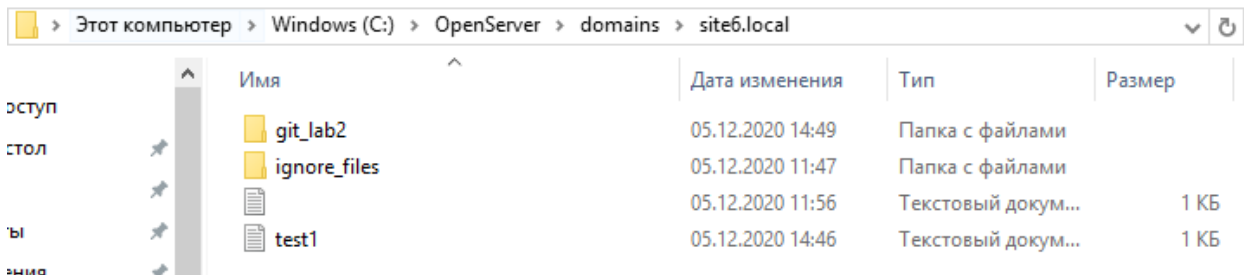


Рисунок 10 – скачивания git-репозитория из GitHub

Создание git-ветки

В терминале введем команду на создание новой git-ветки:

git branch 1.0

Для просмотра всех имеющихся веток вводим команду:

git branch

В ответ получаем имена веток где *звездочкой указана текущая активная ветвь.

Для перехода на ветвь с именем 1.0 необходимо ввести команду:

git checkout 1.0



```
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git branch
1.0
# master
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git checkout 1.0
Switched to branch '1.0'
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 11 – создание новой git-ветки

Теперь создадим в корне проекта папку 1.0, внутри которой текстовый файл с именем 1.0 и внесем в него любое содержимое. На этом этапе мы должны находиться на ветке с именем 1.0.

Добавим файлы:

git add .

Зафиксируем изменения:

git commit -m "Создали папку и файл с именем 1.0 в новой git ветке 1.0"

Переключимся на ветку master командой:

git checkout master

```
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (1.0)
$ git branch
# 1.0
master
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (1.0)
$ git add .
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (1.0)
$ git commit -m "Создади папку и файл с именем 1.0 в новой git ветке 1.0
[1.0 67c1741] Создади папку и файл с именем 1.0 в новой git ветке 1.0
2 files changed, 2 insertions(+)
create mode 100644 1.0/1.0.txt
create mode 160000 git_lab2
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (1.0)
$ git checkout master
Warning: unable to rmdir 'git_lab2' : Directory not empty
Switched to branch 'master'
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 12 – Создание новой ветки

Выполняя [переходы с одной ветки на другую](#), в корне проекта исчезает и появляется папка 1.0.

Слияние и удаление git-веток

Чтобы произвести слияние ветки master и 1.0 нужно переключиться на ветку master:

git checkout master

Командой git merge производим слияние двух git-веток:

git merge 1.0

Затем git-ветку с именем 1.0 можно удалить командой:

git branch 1.0 -D

```
MINGW64:/c/OpenServer/domains/site6.local
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git merge 1.0
Updating ccc32e7..67c1743
Fast-forward
 1.0/1.0.txt |1 +
 git_lab2    |1 +
2 files changed, 2 insertions (+)
create mode 10064 1.0/1.0.txt
create mode 16000 git_lab2
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$ git merge 1.0 -D
Deleted branch 1.0 (was 67c1743) .
RIDOY+admin@RIDOY MINGW64 /c/OpenServer/domains/site6.local (master)
$
```

Рисунок 13 – Слияние веток и удаление ветки 1.0

Вывод

В ходе выполнения лабораторной работы были получены знания и умения управления версиями проекта с помощью системы Git и освоена работа с репозиториями GitHub.

