

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.3
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Костукайло Кирилл Николаевич
1 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Проверил:
Кафедры инфокоммуникаций, старший
преподаватель
Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

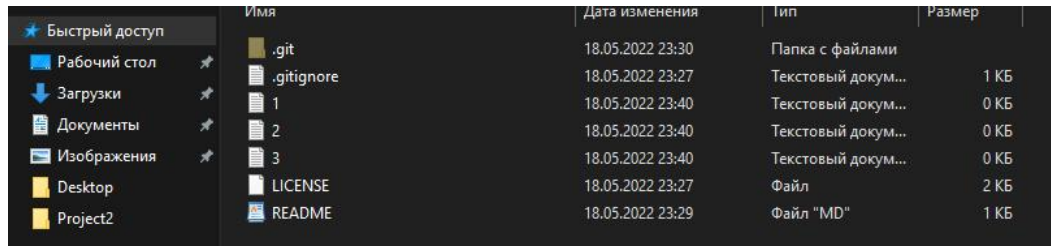
Ставрополь, 2022 г.

Тема: Основы ветвления Git

Цель работы: исследование базовых возможностей по работе с локальными и удалёнными ветками Git.

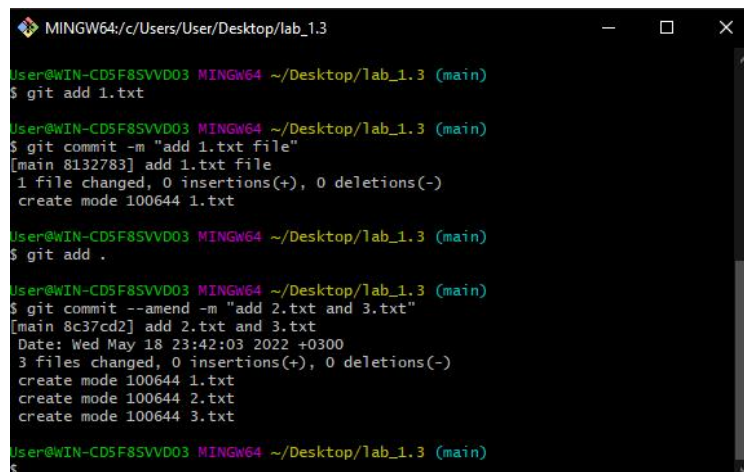
Порядок выполнения работы:

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT.
2. Создать три файла: 1.txt, 2.txt, 3.txt.



Имя	Дата изменения	Тип	Размер
.git	18.05.2022 23:30	Папка с файлами	
.gitignore	18.05.2022 23:27	Текстовый документ...	1 КБ
1	18.05.2022 23:40	Текстовый документ...	0 КБ
2	18.05.2022 23:40	Текстовый документ...	0 КБ
3	18.05.2022 23:40	Текстовый документ...	0 КБ
LICENSE	18.05.2022 23:27	Файл	2 КБ
README	18.05.2022 23:29	Файл "MD"	1 КБ

3. Проиндексировать первый файл и сделать коммит с комментарием "add 1.txt file".
4. Проиндексировать второй и третий файлы.
5. Перезаписать уже сделанный коммит с новым комментарием "add 2.txt and 3.txt".



```
MINGW64/c:/Users/User/Desktop/lab_1.3
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git add 1.txt
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git commit -m "add 1.txt file"
[main 8132783] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git add .
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git commit --amend -m "add 2.txt and 3.txt"
[main 8c37cd2] add 2.txt and 3.txt
Date: Wed May 18 23:42:03 2022 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$
```

6. Создать новую ветку my_first_branch.
7. Перейти на ветку и создать новый файл in_branch.txt, закоммитить изменения.

```
MINGW64/c/Users/User/Desktop/lab_1.3
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git branch my_first_branch

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git checkout my_first_branch
Switched to branch 'my_first_branch'

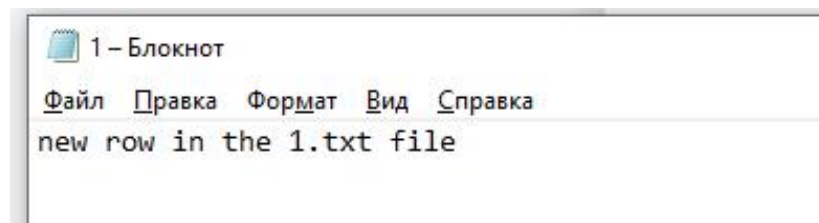
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (my_first_branch)
$ git branch
  main
* my_first_branch

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (my_first_branch)
$ git add .

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (my_first_branch)
$ git commit -m "add in_branch.txt file"
[my_first_branch fc0553e] add in_branch.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (my_first_branch)
$ |
```

8. Вернуться на ветку master.
9. Создать и сразу перейти на ветку new_branch.
10. Сделать изменения в файле 1.txt, добавить строчку “new row in the 1.txt file”, закоммитить изменения.



11. Перейти на ветку master и слить ветки master и my_first_branch, после чего слить ветки master и new_branch.
12. Удалить ветки my_first_branch и new_branch.

```
MINGW64/c/Users/User/Desktop/lab_1.3
commit a9570856abdc41e2ddd3cb692a1ee2379e0c94c
Author: MrGeleos <mr.geleos12@mail.ru>
Date: Wed May 18 23:30:13 2022 +0300

    Added information about yourself

commit 03a4b7daa1e61dac10312995a14ec114d17af7b7 (origin/main, origin/HEAD)
Author: Geleos Sapfirov <99472504+MrGeleos@users.noreply.github.com>
Date: Wed May 18 23:27:11 2022 +0300

    Initial commit

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main|MERGING)
$ git branch -d my_first_branch
Deleted branch my_first_branch (was fc0553e).

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main|MERGING)
$ git branch -d new_branch
error: The branch 'new_branch' is not fully merged.
If you are sure you want to delete it, run 'git branch -D new_branch'.

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main|MERGING)
$ git merge new_branch
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main|MERGING)
$ git commit -m "fix"
[main 6fd6693] fix

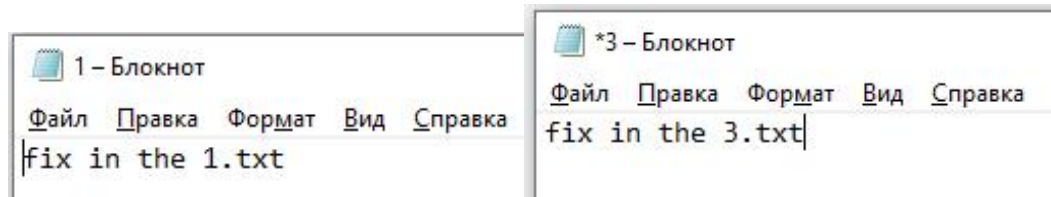
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git merge new_branch
Already up to date.

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git branch -d new_branch
Deleted branch new_branch (was dc28273).

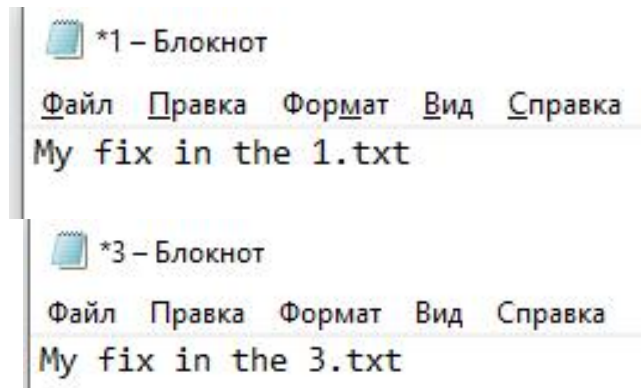
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ |
```

13. Создать ветки branch_1 и branch_2.

14. Перейти на ветку `branch_1` и изменить файл `1.txt`, удалить все содержимое и добавить текст “fix in the 1.txt”, изменить файл `3.txt`, удалить все содержимое и добавить текст “fix in the 3.txt”, закоммитить изменения.



15. Перейти на ветку `branch_2` и также изменить файл `1.txt`, удалить все содержимое и добавить текст “My fix in the 1.txt”, изменить файл `3.txt`, удалить все содержимое и добавить текст “My fix in the 3.txt”, закоммитить изменения.



16. Слить изменения ветки `branch_2` в ветку `branch_1`.

```
MINGW64/c/Users/User/Desktop/lab_13
On branch branch_2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   1.txt
    both modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
user@WIN-CD5F85VDD03 MINGW64 ~/Desktop/lab_1.3 (branch_2|MERGING)
$ git add 1.txt
user@WIN-CD5F85VDD03 MINGW64 ~/Desktop/lab_1.3 (branch_2|MERGING)
$ git commit -m "Fix 1.txt"
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
J      3.txt
user@WIN-CD5F85VDD03 MINGW64 ~/Desktop/lab_1.3 (branch_2|MERGING)
$ git status
On branch branch_2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   1.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   3.txt
user@WIN-CD5F85VDD03 MINGW64 ~/Desktop/lab_1.3 (branch_2|MERGING)
```

17. Решить конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду `git mergetool` с помощью одной из доступных утилит, например Meld.
18. Отправить ветку `branch_1` на GitHub.

```
MINGW64/c/Users/User/Desktop/lab_1.3
commit fc0553e8b7cc3e30ce4f5eae42e469a66a47a347
Author: MrGeleos <mr.geleos12@mail.ru>
Date: Wed May 18 23:52:00 2022 +0300

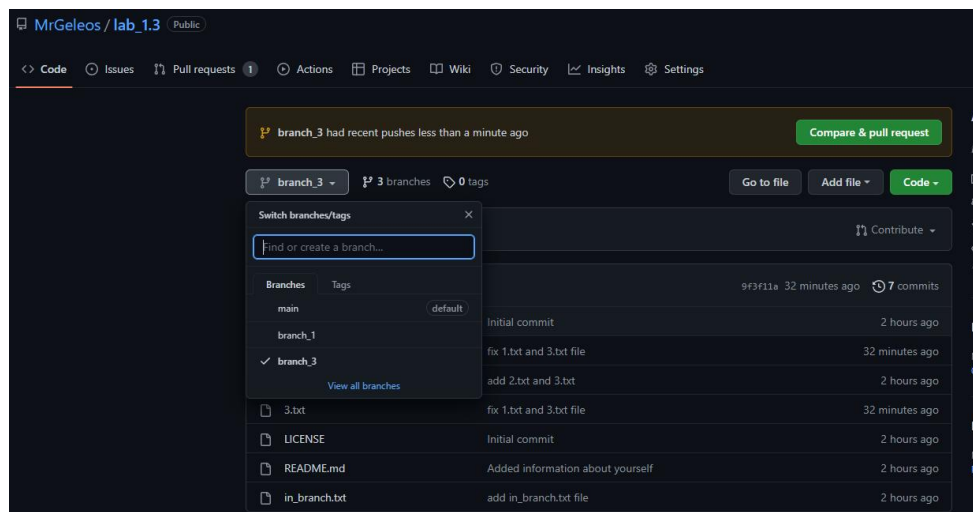
    add in_branch.txt file

commit 8c37cd27d80ad64cc687d6980b658540fa1d3115
Author: MrGeleos <mr.geleos12@mail.ru>
Date: Wed May 18 23:42:03 2022 +0300

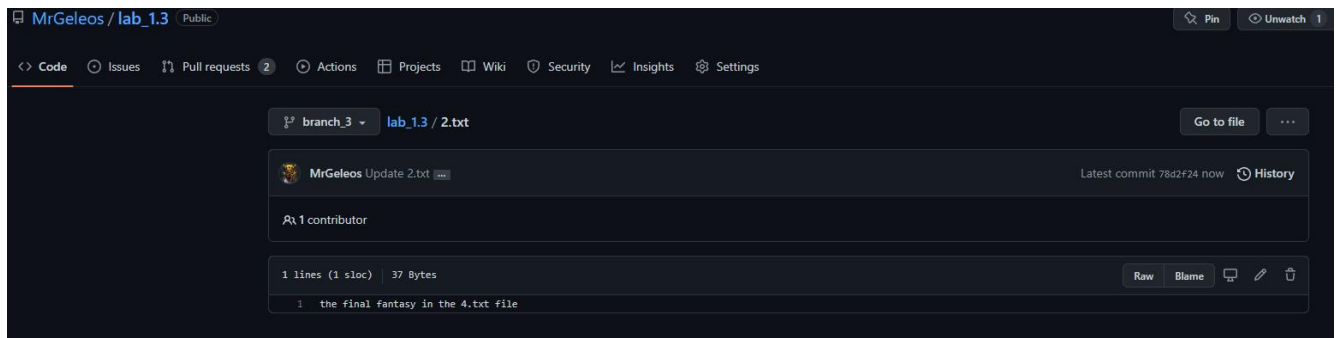
    add 2.txt and 3.txt

commit a9570856abdc41e2ddd3cb692a1eee2379e0c94c
User@WIN-CD5F85VVD03 MINGW64 ~/Desktop/lab_1.3 (branch_1)
$ git push --set-upstream origin branch_1
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (17/17), 1.43 KiB | 731.00 KiB/s, done.
Total 17 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/MrGeleos/lab_1.3/pull/new/branch_1
remote:
To https://github.com/MrGeleos/lab_1.3.git
 * [new branch]      branch_1 -> branch_1
branch 'branch_1' set up to track 'origin/branch_1'.
User@WIN-CD5F85VVD03 MINGW64 ~/Desktop/lab_1.3 (branch_1)
```

19. Создать средствами GitHub удаленную ветку `branch_3`.



20. Создать в локальном репозитории ветку отслеживания удаленной ветки `branch_3`.
21. Перейти на ветку `branch_3` и добавить файл файл 2.txt строку "the final fantasy in the 4.txt file".



22. Выполнить перемещение ветки master на ветку branch_2.

23. Отправить изменения веток master и branch_2 на GitHub.

```
MINGW64~/c/Users/User/Desktop/lab_1.3
push
User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (branch_2)
$ git push --set-upstream origin branch_2
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 732 bytes | 732.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/MrGeleos/lab_1.3/pull/new/branch_2
remote:
To https://github.com/MrGeleos/lab_1.3.git
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (branch_2)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MrGeleos/lab_1.3.git
 03a4b7d..6fd6693  main -> main

User@WIN-CD5F8SVVD03 MINGW64 ~/Desktop/lab_1.3 (main)
$
```

Вопросы для защиты работы

1. Что такое ветка?

Это последовательность коммитов, отклоняющаяся от основной линии разработки.

2. Что такое HEAD?

Это указатель, задача которого ссылаться на определённый коммит в репозитории, если точнее: на коммит, который станет родителем для следующего коммита.

3. Способы создания веток.

Новую ветку можно создать командой `git branch <название_ветки>` или на удаленном репозитории `git hub`

4. Как узнать текущую ветку?

При помощи команды `git branch`.

5. Как переключаться между ветками?

При помощи команды `git checkout <название_ветки>`.

6. Что такое удалённая ветка?

Это ветка, находящаяся на удаленном репозитории. Или ссылка на состояние ветки на удаленном репозитории.

7. Что такое ветка отслеживания?

Это ветка в локальном репозитории, которая напрямую связана с удаленной веткой на удаленном репозитории.

8. Как создать ветку отслеживания?

Командой `git checkout --track origin/<название_ветки>`.

9. Как отправить изменения из локальной ветки в удаленную ветку?

Командой `git push origin <название_ветки>`.

10. В чем отличие команд `git fetch` и `git pull` ?

`Git pull` – это сочетание команд `git fetch` (получение изменений с удаленного репозитория) и `git merge` (объединение веток).

11. Как удалить локальную и удаленную ветки?

Используя команду `git branch -d <название_ветки>`. Для удаление удаленной ветки существует команда `git push origin -d <название_ветки>`.

Вывод: освоил базовые возможности по работе с локальными и удалёнными ветками `Git`.