**ОБЛАСТНОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КУРСКИЙ ТЕХНИКУМ СВЯЗИ»**

**Лабораторные работы**

**С GIT**

**Выполнил: студент 3 курса группы ИСП 3.2**

**Шмалюк Н.М.**

Лабораторная работа №1 «Первоначальная настройка git»

**Тема:** Первоначальная настройка git. Инициализация каталога. Состояния фалов в git. Первый коммит. Цель работы: провести первоначальную настройку системы контроля версии git, после установки инициализировать каталог для работы, разобраться с существующими состояниями файлов в git, сделать первый коммит.

**Порядок выполнения работы**

1. Изучить теоретическую часть работы.

2. Зайти в папку T://{Номер группы} и в ней создать папку соответствующую инициалам студента на английском языке. Например, для студента Иванов Петр Петрович, папка будет иметь имя IPP.

3. Провести инициализацию репозитория в созданной папке. Для этого, открыть программу Git Bash, перейти в созданную папку (для перемещения используется команда cd T://{Номер группы}/{Инициалы}).

4. Установить настройки имени и e-mail’а, не используя опцию --global.

5. Создать в папке файл my\_first\_file.txt и проиндексировать его.

6. Сделать первый коммит.

7. Открыть файл my\_first\_file.txt и добавить в него строчку “test row”. Проиндексировать изменения.

8. Создать новый файл my\_second\_file.txt. Проиндексировать изменения.

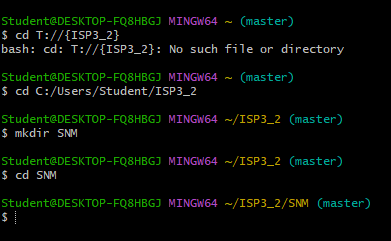
9. Сделать второй коммит.

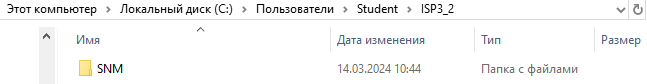
10. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

**Ход работы**

2. Зайти в папку T://{Номер группы} и в ней создать папку соответствующую инициалам студента на английском языке. Например, для студента Иванов Петр Петрович, папка будет иметь имя IPP.

C:\Users\Student\ISP3\_2

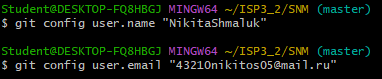




3.Провести инициализацию репозитория в созданной папке. Для этого, открыть программу Git Bash, перейти в созданную папку (для перемещения используется команда cd T://{Номер группы}/{Инициалы}).



4. Установить настройки имени и e-mail’а, не используя опцию --global.

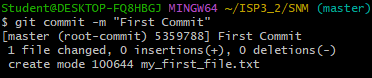


5. Создать в папке файл my\_first\_file.txt и проиндексировать его.





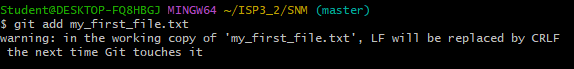
6. Сделать первый коммит.



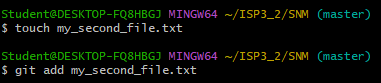
7. Открыть файл my\_first\_file.txt и добавить в него строчку “test row”. Проиндексировать изменения.



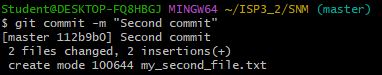




8.Создать новый файл my\_second\_file.txt. Проиндексировать изменения.



9. Сделать второй коммит.



**Лабораторная работа № 2**

**«Игнорирование, сравнение, удаление и перемещение файлов»**

**Тема:** Игнорирование файлов. Сравнение изменений. Удаление и перемещение файлов. Цель работы: научиться исключать файлы, которые нет необходимости вести в системе контроля версий.

**Порядок выполнения работы**

1. Изучить теоретическую часть работы.

2. Продолжить работу с созданным репозиторием на первой лабораторной работе.

3. Создать папку temp в своем репозитории.

4. Создать папку log и добавить в нее 2 файла: main.html и some.tmp.

5. Создать файл .gitignore и добавить в игнорирование папку temp и файлы с расширением .tmp из папки log.

6. Закоммитить добавление файла .gitignore.

7. Внести изменения в файл my\_first\_file.txt, добавив строчку “row to index”, проиндексировать данные изменения. Еще раз внести изменения в файл, добавив строчку “row no index”.

8. Посмотреть индексированные и неиндексированные изменения используя команду git diff.

9. Удалить файл my\_first\_file.txt, зафиксировать данное удаление.

10. Переименовать файл my\_second\_file.txt в my\_first\_file.txt, зафиксировать изменение.

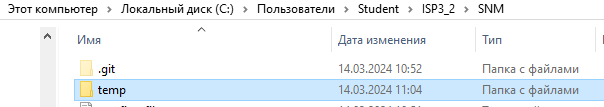
11. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы

**Ход работы:**

2. Продолжить работу с созданным репозиторием на первой лабораторной работе.

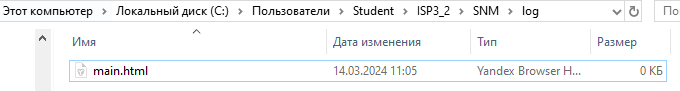


3.Создать папку temp в своем репозитории.

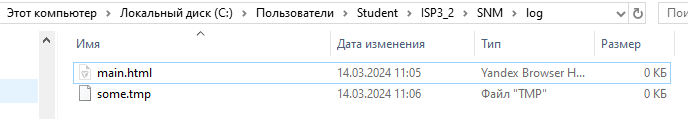


4. Создать папку log и добавить в нее 2 файла: main.html и some.tmp.



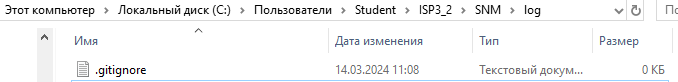


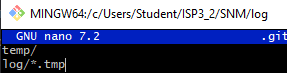




5. Создать файл .gitignore и добавить в игнорирование папку temp и файлы с расширением .tmp из папки log.



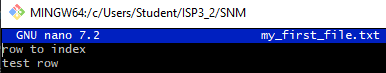


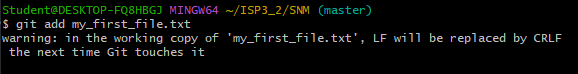


6. Закоммитить добавление файла .gitignore.

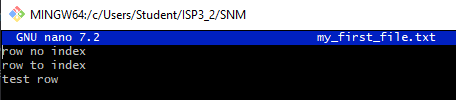


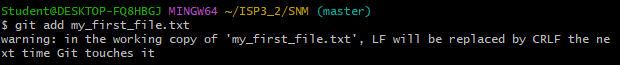
7. Внести изменения в файл my\_first\_file.txt, добавив строчку “row to index”, проиндексировать данные изменения.



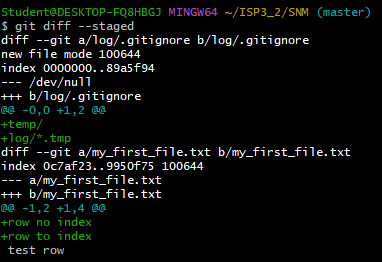


Еще раз внести изменения в файл, добавив строчку “row no index”.

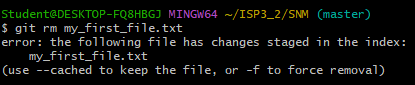


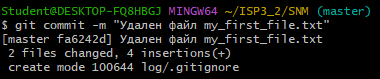


8. Посмотреть индексированные и неиндексированные изменения используя команду git diff.



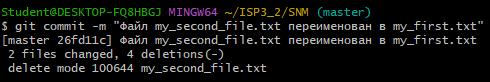
9. Удалить файл my\_first\_file.txt, зафиксировать данное удаление.

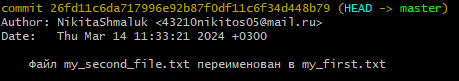




10. Переименовать файл my\_second\_file.txt в my\_first\_file.txt, зафиксировать изменение.







**Лабораторная работа №3 «Просмотр истории коммитов»**

**Тема:** Просмотр истории коммитов, команда git log.

**Порядок выполнения работы**

1. Изучить теоретическую часть работы.

2. Продолжить работу с созданным репозиторием.

3. Изучить возможности команды git log, выполнить различные варианты вывода информации и ее отбора.

4. Выполнить задание согласно варианту.

5. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

1 Вывести коммиты, автором которых являетесь Вы, за последний месяц.

2 Вывести все коммиты в формате: короткий хеш, автор, комментарий.

3 Вывести все коммиты, в сообщении которых присутствует слово my.

4 Вывести все коммиты за текущий месяц с информацией о том, какие файлы были изменены.

5 Вывести информацию о первом коммите в системе, с выводом дельты 7 (diff).

6 Вывести коммиты сделанные за последний месяц назад, но исключая последнюю неделю.

7 Вывести информацию о коммитах в формате: автор, дата, список измененных файлов.

8 Вывести коммиты, автором которых являетесь Вы, с выводом дельты (diff).

9 Вывести коммиты, в которых происходили изменения файла my\_first\_file.txt, за последние 2 недели.

10 Вывести последние 3 коммита в формате: автор, комментарий.

11 Вывести информацию о первом коммите в формате: дата, автор, комментарий, а также список измененных файлов.

12 Вывести коммиты, автором которых являетесь Вы, со списком измененных файлов.

13 Вывести все коммиты за текущий месяц в формате: сокращенный хеш, дата, комментарий.

14 Вывести все коммиты в формате: e-mail автора, дата коммита, хеши родительских коммитов.

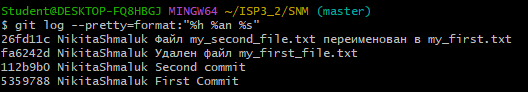
15 Вывести коммиты, в которых происходили изменения файла my\_first\_file.txt.

**Ход работы:**

**Мой вариант: Вариант 2**

2 Вывести все коммиты в формате: короткий хеш, автор, комментарий.





**Лабораторная работа №4**

**«Отмена изменений. Работа с метками»**

**Тема: Отмена внесенных изменений. Работа с метками.**

**Порядок выполнения работы**

1. Изучить теоретическую часть работы.

2. Продолжить работу с созданным репозиторием.

3. Создать три файла: 1.txt, 2.txt, 3.txt.

4. Проиндексировать первый файл и сделать коммит с комментарием “add 1.txt file”.

5. Проиндексировать второй и третий файлы.

6. Удалить из индекса второй файл.

7. Перезаписать уже сделанный коммит с новым комментарием “add 1.txt and 3.txt”

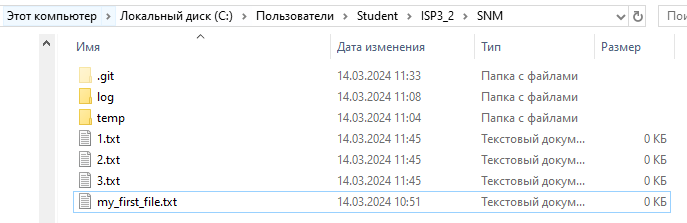
8. Создать аннотированную метку с названием v0.01.

9. Создать легковесную ветку указывающую на первый коммит в репозитории.

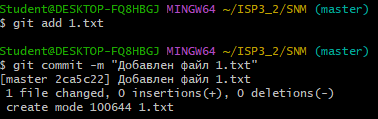
10. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

**Ход работы:**

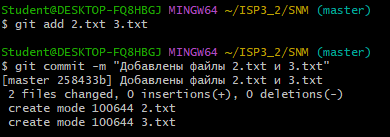
3. Создать три файла: 1.txt, 2.txt, 3.txt.



4. Проиндексировать первый файл и сделать коммит с комментарием “add 1.txt file”.



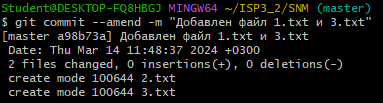
5. Проиндексировать второй и третий файлы.



6. Удалить из индекса второй файл.



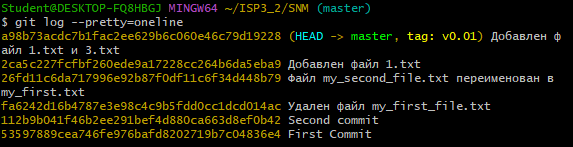
7. Перезаписать уже сделанный коммит с новым комментарием “add 1.txt and 3.txt”



8. Создать аннотированную метку с названием v0.01.



9. Создать легковесную ветку указывающую на первый коммит в репозитории.



**Лабораторная работа №5**

**«Ветвление. Конфликты»**

**Тема: Работа с ветками, решение конфликтов**

**Порядок выполнения работы**

1. Изучить теоретическую часть работы.

2. Продолжить работу с созданным репозиторием.

3. Создать новую ветку my\_first\_branch.

4. Перейти на ветку и создать новый файл in\_branch.txt, закоммитить изменения.

5. Вернуться на ветку master.

6. Создать и сразу перейти на ветку new\_branch.

7. Сделать изменения в файле 1.txt, добавить строчку “new row in 1.txt file”, закоммитить изменения.

8. Перейти на ветку master и слить ветки master и my\_first\_branch, после чего слить ветки master и new\_branch.

9. Удалить ветки my\_first\_branch и new\_branch.

10. Создать ветки branch\_1 и branch\_2.

11. Перейти на ветку branch\_1 и изменить файл 1.txt, удалить все содержимое и добавить текст “fix in 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “fix in 3.txt”, закоммитить изменения.

12. Перейти на ветку branch\_2 и также изменить файл 1.txt, удалить все содержимое и добавить текст “My fix in 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “My fix in 3.txt”, закоммитить изменения.

13. Слить изменения ветки branch\_2 в ветку branch\_1.

14. Решить конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду git mergetool c утилитой Meld.

15. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

**Ход работы:**

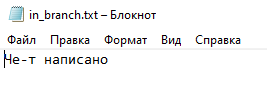
3. Создать новую ветку my\_first\_branch.



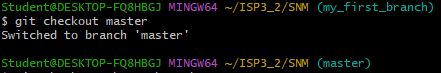
4. Перейти на ветку и создать новый файл in\_branch.txt, закоммитить изменения.



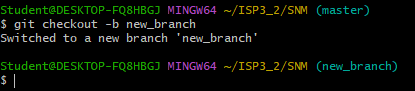
создать новый файл in\_branch.txt



5. Вернуться на ветку master.

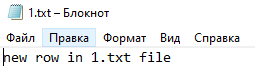


6. Создать и сразу перейти на ветку new\_branch.



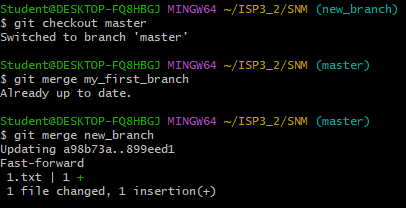
7. Сделать изменения в файле 1.txt, добавить строчку “new row in 1.txt file”, закоммитить изменения.







8. Перейти на ветку master и слить ветки master и my\_first\_branch, после чего слить ветки master и new\_branch.

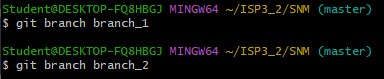


9. Удалить ветки my\_first\_branch и new\_branch.

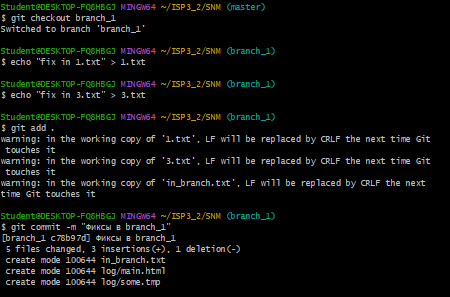




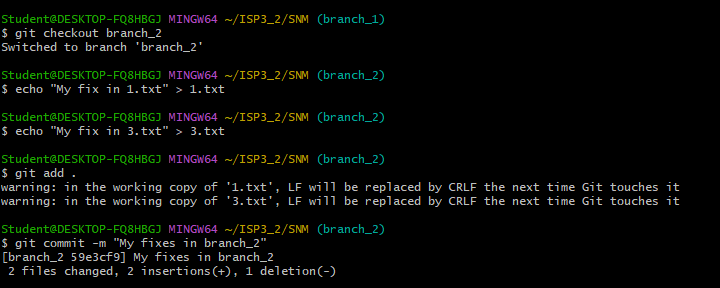
10. Создать ветки branch\_1 и branch\_2.



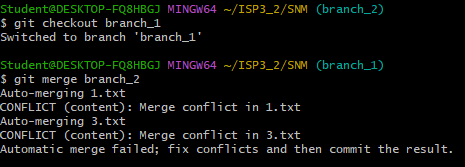
11. Перейти на ветку branch\_1 и изменить файл 1.txt, удалить все содержимое и добавить текст “fix in 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “fix in 3.txt”, закоммитить изменения.



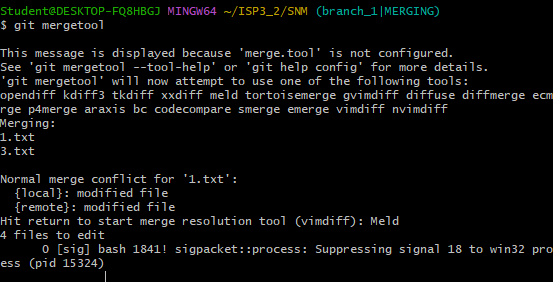
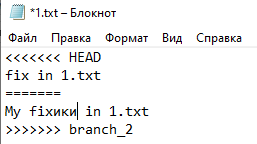
12. Перейти на ветку branch\_2 и также изменить файл 1.txt, удалить все содержимое и добавить текст “My fix in 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “My fix in 3.txt”, закоммитить изменения.



13. Слить изменения ветки branch\_2 в ветку branch\_1.



14. Решить конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду git mergetool c утилитой Meld.



**Лабораторная работа №6**

**«Прятанье»**

**Тема:** Механизм прятанья.

**Порядок выполнения работы**

1. Изучить теоретическую часть работы.

2. Продолжить работу с созданным репозиторием.

3. Проверить какие ветки слиты с веткой master, а какие нет.

4. Удалить все ветки слитые с master.

5. Создать новую ветку work и перейти в нее.

6. Изменить файл 1.txt.

7. Спрятать данные изменения.

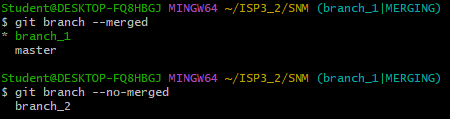
8. Развернуть обратно данные изменения с опцией --index.

9. Развернуть спрятанные изменения в новую ветку.

10. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

**Ход работы:**

3. Проверить какие ветки слиты с веткой master, а какие нет.



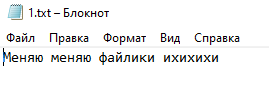
4. Удалить все ветки слитые с master.



5. Создать новую ветку work и перейти в нее.



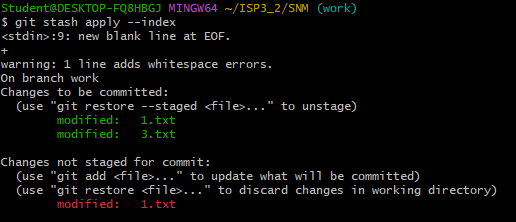
6. Изменить файл 1.txt.



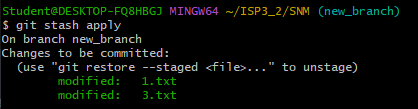
7. Спрятать данные изменения.



8. Развернуть обратно данные изменения с опцией --index.



9. Развернуть спрятанные изменения в новую ветку.



**Лабораторная работа №7**

**«Работа с удаленным репозиторием»**

*Тема: Работа с удаленным репозиторием. Github.com.*

**Порядок выполнения практической работы**

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием.
3. Пройти регистрацию на сайте github.com.
4. Настроить доступ к github по SSH.
5. Склонировать репозиторий git@github.com:mavose/tusur\_{номер груп-

пы}.git.

1. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы

**Ход работы:**

