УО “Белорусский государственный технологический университет”

Факультет информационных технологий

Программная инженерия

Лабораторная работа 9

По дисциплине «Основы программной инженерии»

На тему «Системы контроля версий. Основы работы в Git »

Выполнил:

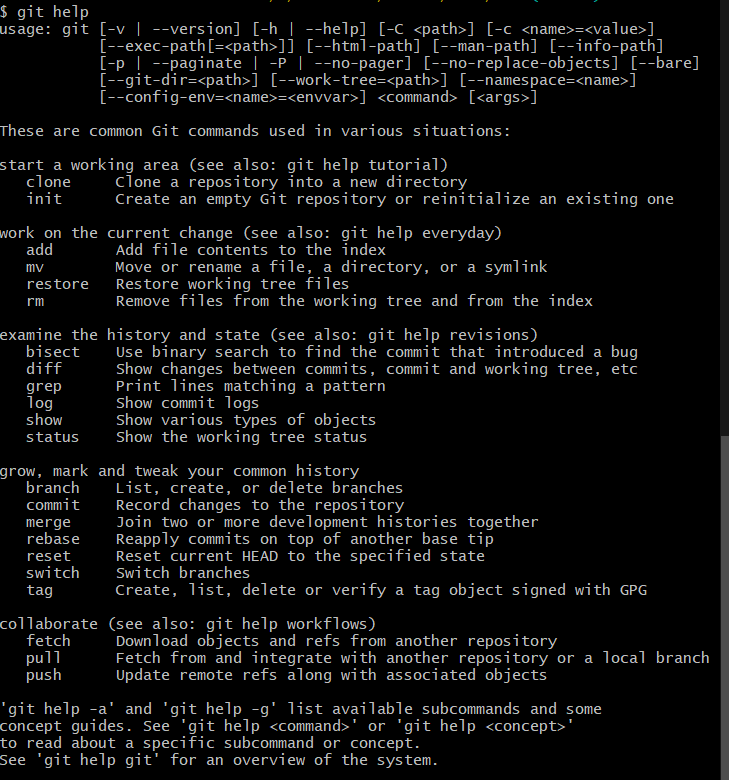
Студент 1 курса 8 группы

Борисов Никита Андреевич

Преподаватель: Ольга Александровна

4.Установленная версия

5. Все возможные команды можно получить с помощью команды $git help



6. Перейти в проводнике в рабочую папку, где планируется создать репозиторий, и запустить Git Bash с помощью контекстного меню: «Git Bash Here».

Инициализация репозитория в выбранной папке выполняется командой $git init

"**Init**" - это сокращение от "**initialize**" ( т.е. "инициализироать" или "включать", "запускать").

Это значит, что когда Вы пишете "**git init**", Git "**включается**" или "**запускается**" для данного репозитория (т.е.

папки).

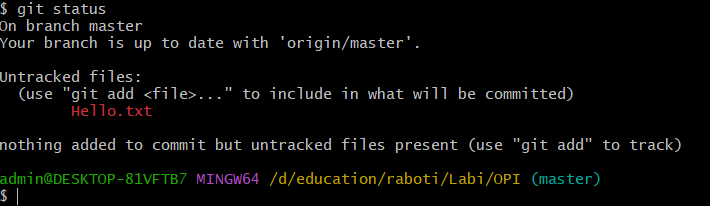
В папке появилась новая скрытая папка: .git – локальный репозиторий



Текущее состояние (status) репозитория отображается командой: $ git status

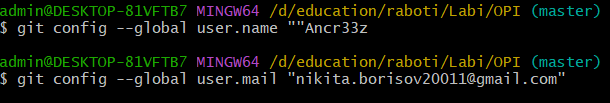
Кроме того, была создана ветка master:

(Ветка в git -перемещаемый указатель на один из коммитов -способ сохранения изменений в коде)

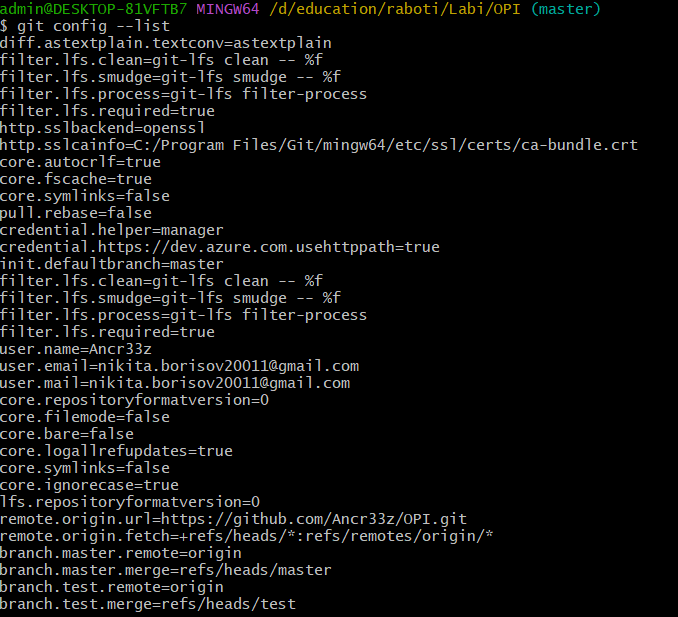


Красным цветом отмечаются новые и модифицированные файлы и папки.(Но в данном случае зелёным, поскольку это не новый файл)

7.

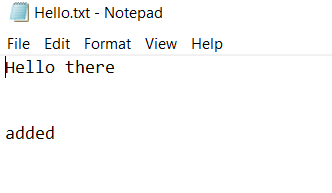


8. настройки конфигурационного файла командой $git config --list



С помощью $git config –list можно посмотреть список всех установленных настроек , для отображения параметров.

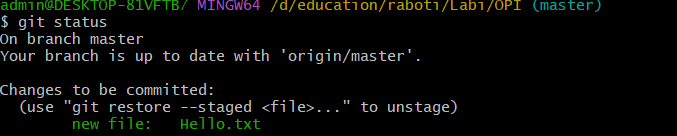
9. В папке находится файл Hello.txt со следующим содержимым:



Добавим файл Hello.txt в репозиторий индексированных файлов командой:$git add Hello.txt

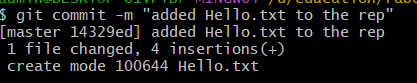


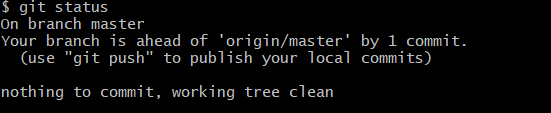
Текущее (обновленное) состояние репозитория отображается командой:$git status



Теперь файл Hello.txt проиндексирован.

Эти изменения можно зафиксировать в репозитории командой: $ git commit -m "added Hello.txt to the repo" Ключ –m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo")





Git информирует об успешном создании нового коммита (в ветку master добавлен 1 файл):

[master (root-commit) 10f2aa1] added Hello.txt to the repo 1 file changed, 4 insertions(+)

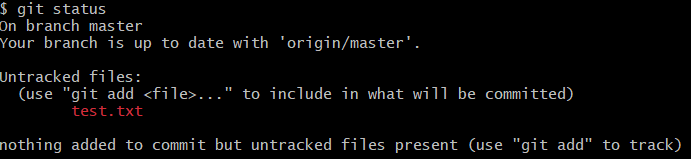
Теперь состояние файла Hello.txt зафиксировано в репозитории.

(git commit - это команда для записи индексированных изменений в репозиторий Git.)

(m- вносит описание изменений)

10.

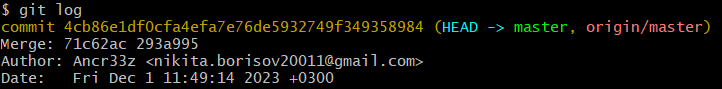
11.



Мы можем просматривать, какие файлы были добавлены или изменены. Например, если у нас появится новый файл, он будет отображаться красным.

Если Git начнет за ним "следить", файл станет зеленым.

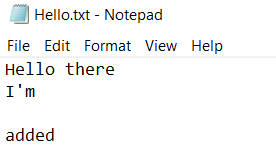
Git status- можно посмотреть текущее состояние нашего репозитория

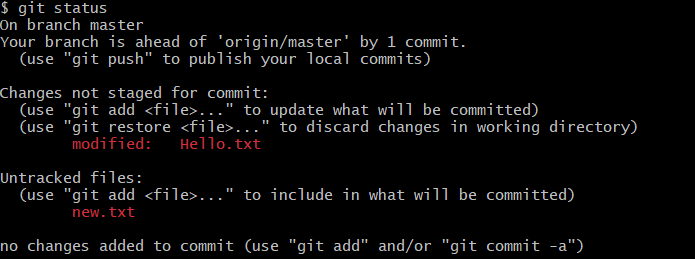
12-17.

которая показывает лог commits начиная с указателя HEAD

Она показывает список всех выполненных коммитов, отсортированных по дате добавления. Сверху мы видим самые последние коммиты/Из этого вывода мы можем узнать кто, когда и какие коммиты делал

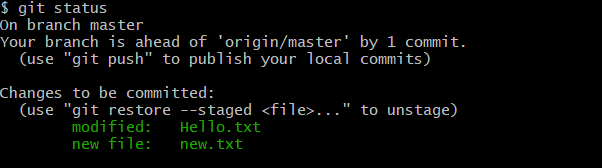
В текстовый файл Hello.txt внесли изменения и создали файл new.txt



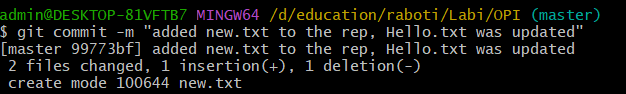


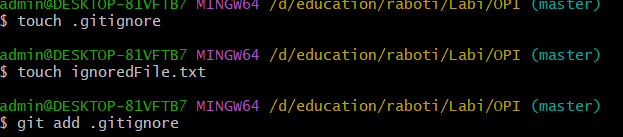
Hello.txt new.txt отабражаются красными поскольку это новые файлы



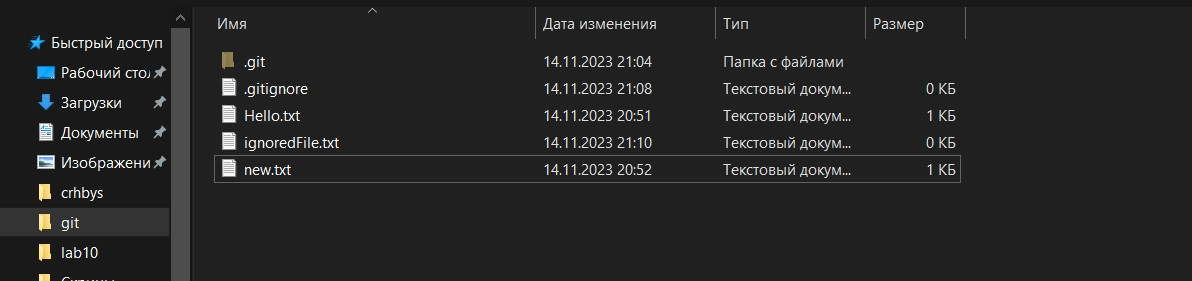
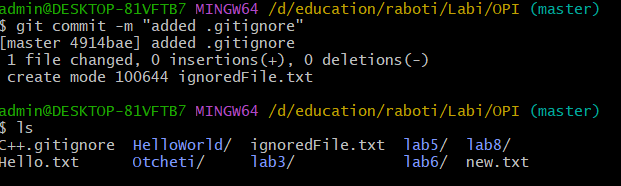


Использовали $git status для обновленной информации





Создадим файл:$touch .gitignore

Создадим файл:$touch .ignoredFile.t

$ls , мы можем увидеть , что .gitignore не выводиться в консоль, тоесть игнорируем

git add – добавляет файлы в индекс

git commit – выполняет коммит проиндексированных файлов в репозиторий

git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы

Все коммиты в Git логируются. Просмотреть журнал можно с помощью команды $ git log

В Git существует три места, где хранятся настройки: − на уровне системы; − на уровне пользователя; − на уровне проекта (репозитория).

Все параметры будут помещены в файл с настройками .gitconfig, расположенном в домашнем каталоге пользователя



**Вопросы/ответы**

**1.Что такое система контроля версий, для чего ее используют?**

Система управления версиями (от англ. Version Control System, VCS или Revision Control System, RCS) – программное обеспечение для облегчения работы с изменяющейся информацией и разработки проекта совместно с коллегами.

Назначение систем контроля версий :

автоматическое создание архива (бэкап) для синхронизации кодовой базы;

отслеживание изменений (кто, когда и зачем сделал изменения);

совместная работа над одним и тем же проектом;

отслеживание ошибок (Bug трекинговая система).

**2.Какой принцип хранения файлов использует Git?**

Git является распределенным, то есть не зависит от одного центрального сервера , на котором находятся файлы. Вместо этого он работает полностью локально , сохраняя данные на жестком диске, которые называются репозиторием.

**Git** — распределённая система контроля версий, позволяющая сохранять изменения, внесённые в файлы, которые хранятся в репозитории. Сами изменения сохраняются в виде снимков, называемых коммитами. Они могут размещаться на разных серверах, поэтому вы всегда восстановите код в случае сбоя, а также без проблем откатитесь до любого предыдущего состояния

**3.В чем отличие Git от других систем контроля версий?**

Отличие Git от других систем контроля версий заключается в его распределенной структуре. Каждый пользователь имеет локальную копию репозитория, что обеспечивает быструю работу и возможность работы без подключения к сети.

Основное отличие Git от любой другой системы контроля версий (включая Subversion и её собратьев) — это подход к работе со своими данными. Концептуально, большинство других систем хранят информацию в виде списка изменений в файлах. Эти системы (CVS, Subversion, Perforce, Bazaar и т. д.) представляют хранимую информацию в виде набора файлов и изменений, сделанных в каждом файле, по времени (обычно это называют контролем версий, **основанным на различиях**).

Git не хранит и не обрабатывает данные таким способом. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён. Git представляет свои данные как, скажем, **поток снимков**.

**4.В каких трех основных состояниях файлы могут находиться в Git ?**

Файлы в Git могут находиться в трех основных состояниях: *измененные (modified),* *подготовленные (staged) и зафиксированные (committed).*

**5.Что такое индексация файла? Какой командой она выполняется?**

Индексация файла - это процесс добавления измененных файлов в индекс, чтобы они были готовы к фиксации. Это выполняется командой "git add"

**6. Что такое фиксация файла? Какой командой она выполняется?**

Фиксация файла - это процесс сохранения изменений в индексированных файлах в репозитории. Это выполняется командой "git commit".

**7.Продемонстрируйте команду проверки выбранных настроек.**

Команда для проверки выбранных настроек в Git может выглядеть так: "git config --list".

**8.Как инициализировать репозиторий в существующей папке?**

Для инициализации репозитория в существующей папке нужно выполнить команду "git init" в этой папке.

**9.Как указать файлы, за которыми должна следить система?**

Чтобы указать файлы, за которыми должна следить система, нужно добавить их в индекс с помощью команды "git add".

**10.Как выполнить фиксацию изменений?**

Для выполнения фиксации изменений нужно использовать команду "git commit -m 'Сообщение о фиксации'".

**11.Какой командой определить состояния файлов?**

Состояния файлов можно определить с помощью команды "git status".

**12.Для чего создается файл .gitignore? Поясните его структуру.**

Файл .gitignore создается для игнорирования определенных файлов и папок при работе с Git. Его структура состоит из шаблонов для игнорируемых файлов и папок.