Отчёт по лабораторной работе № 1

Работа с Git и Markdown

Абу Сувейлим Мухаммед Мунифович

Содержание

# Цель работы

* Во-первых, ознкомится с Git и Markdown.
* Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
* В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

# Задание

* Создать новую репозиторию на GitHub
* Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
* В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

# Теоретическое введение

Git — абсолютный лидер по популярности среди современных систем управления версиями. Это развитый проект с активной поддержкой и открытым исходным кодом. Система Git была изначально разработана в 2005 году Линусом Торвальдсом — создателем ядра операционной системы Linux. Git применяется для управления версиями в рамках колоссального количества проектов по разработке ПО, как коммерческих, так и с открытым исходным кодом. Система используется множеством профессиональных разработчиков программного обеспечения. Она превосходно работает под управлением различных операционных систем и может применяться со множеством интегрированных сред разработки (IDE).

# Выполнение лабораторной работы

1. Во-первых, я устновил Git и gh на моем пк.
2. Выполнил следующие команды, чтобы git узнал мое имя и электронную почту.

git config --global user.name "Mukhammed"  
git config --global user.email "mohammedswelem13@gmai.com"

1.1.1

1. Настройл core.autocrlf с параметрами true, чтобы сделать все переводы строк текстовых файлов в главном репозитории одинаковы.

git config --global core.autocrlf true  
git config --global core.safecrlf true

1.1.2

1. Чтобы избежать нечитаемых строк, я установил соответствующий флаг.

git config --global core.quotepath off

1.1.3

1. Создал пустой каталог hello и файл hello.html

* 1.2.1 1.2.1

1. Я intilized git, чтобы потом создать репозиторию.

1.2.3

1. Проверил текущее состояние репозитория.

1.2.4

1. Добавил HTML-теги к нашему приветствию.

<h1>Hello, World!</h1>

1.3.1

1. В первой строке ввел комментарий: «Added h1 tag».

1.4.1

1. Изменил страницу «Hello, World», чтобы она содержала стандартные теги html и body. Далее, добавил заголовки HTML (секцию head) к странице «Hello, World».

1.4.2а

1.4.2b

1. Я Получил список произведенных изменений

1.4.3

1. Возвращался назад в историю используя команду checkout и вернитеся к последней версии в ветке master.

1.4.4a

1.4.4b

1. Далее, я создал новые теги v1 и v1-beta.

1.4.5

1. Измененил в файл hello.html в виде нежелательного комментария

1.6.1а

1. Выполнил сброс буферной зоны

1.6.3

1. Далее, я изменил файл hello.html и сделал коммит

1.7.2а

1.7.2

1. Я сделал коммит с новыми изменениями, отменяющими предыдущие

1.7.3

1. Я сбросил коммит к предшествующим коммиту Oops

1.8.4

1. Я удалил тега oops

1.9.1

1. После того как я добавил комет, я решил добавить еше мой номер телефона, но без нового комета.

1.10.3

1. Создал каталог lib и в нем переносил страницу.

1.11.1

1. Я нашел инфо про тагов и веток.

1.14.5а

1.14.5b

1.14.5c

1.14.5d

1. Вывол последный коммит с помощью SHA1 хэша

1.15.2

1. Поиск дерева

1.15.3

1. Вывод каталога lib

1.15.4

1. Вывод файла hello.html

1.15.5

1. Создал ветку style и файл lib/style.css

1.16.1

1.16.2

1. Просмотрил текущие ветки

1.19.2

1. Слияние веток

1.20.1

1. Слияние master с веткой style

1.22.1а

1. Сделал коммит решения конфликта

1.22.3

1. Сбросил ветку style

1.23.1

1. Перебазирование

1.25а

1. Создать клон репозитория hello

1.27.2

1. Просмотрим историю репозитория

1.28.2

1. Слейте извлеченные изменения в локальную ветку master

1.32.1

1. Проверка файла README.md

1.31.3

1. Создал чистый репозиторий

1.35

1. Добавление удаленного репозитория

1.36

1. Извлечение общих изменений

1.38

# Выводы

Git is harder than you would intially be allowed to think. This mainly steams from what seems more abstraction (compared to doing creating repos and files manually on GitHub) but it is also the reason to why it works best when dealing with a signficant number of files and dirctories. Manging big systems is best suited for linux based OS compared to Windows or MacOS.

# Список литературы

Репозиторий: https://github.com/yamadharma/course-directory-student-template.