ЛАБОРАТОРНАЯ РАБОТА № 7

“Исследование распределенных систем контроля версий Git при коллективной разработке программных продуктов”

1 ЦЕЛЬ РАБОТЫ

Исследовать основные подходы к организации взаимодействия команды разработчиков с использованием распределенной системы контроля версий (DVCS). Приобрести практические навыки установки и настойки DVCS Git, организации ветвей разработки и осуществление слияния.

3 ХОД РАБОТЫ

Для начала работы в созданной директории производим процедуру создания локального репозитория – git init. Добавляем файл в репозиторий – git add. Проверяем статус репозитория – git status.

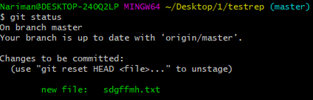


Рисунок 1 – Проверка статуса файлов репозитория

git commit –m “created test file”

После этого все изменения сохранены в локальном репозитории.

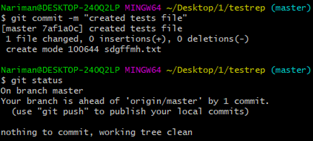


Рисунок 2 – Данные о изменении

Теперь запушим изменения в удаленный репозиторий.

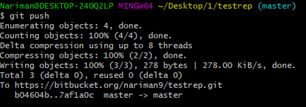


Рисунок 3 – Внесение изменений в удаленный репозиторий

Изменим исходный репозиторий и снова обновим.



Рисунок 4 – Получение ревизии из удаленного репозитория

Проверим изменения в локальном репозитории.



Рисунок 5 – Изменения в локальном репозитории

Создадим новую ветвь в одном из репозиториев.

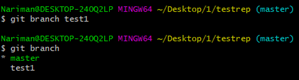


Рисунок 6 – Создание ветвей

После внесения изменений в локальные репозитории произведём слияние ветвей.

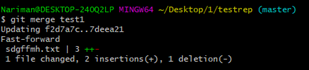


Рисунок 7 – Слияние ветвей

Команды используемые при выполнении работы:

git init – создает локальный репозиторий

git add . – добавляет новые файлы в систему контроля версий

git commit –m ‘CommitName’– фиксация изменений в текущем репозитории с комментарием «CommitName»

git push – вытягивает данные из локального репозитория в удаленный

git pull – вытягивает данные из удаленного в логакльный репозиторий

git merge – сливает ветки в одну

hg branch – показывает именованные ветви и выделяет текущую

hg branch <имя> – создает ветвь

ВЫВОД

В ходе выполнения лабораторной работы были исследованы основные свойства и механизмы работы с системой контроля версий GIT. Было отмечено различие синтаксиса команд в отличие от ранее рассмотреной Mercurial. Были закреплены знания по работе с распределенными системами контроля версий. Также было отмечено, что в настоящее время данная технология является обязательной для каждого разработчика, так как предоставляется совершенное удобство и возможности для работы с программный продуктом.