

Front matter

lang: ru-RU title: Управление версиями. subtitle: ЛР по ОС №2 author: Танрибергенов Эльдар Марсович group: НПИбд-02-20

Отчёт

о выполнении лабораторной работы № 2.

Управление версиями.

Выполнил:

студент группы НПИбд-02-20

Танрибергенов Эльдар Марсович.

Студ. билет № 1032208074

Москва 2021 г.

Цель работы:

- *Изучить идеологию и применение средств контроля версий.*

Ход работы:

Вывод:

В результате лабораторной работы я ознакомился с системой контроля версий GIT.

Ответы на контрольные вопросы:

1. Система контроля версий --- это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии.

2. VCS -- это [хранилище]{.underline}; [коммитнуть]{.underline} -- внести изменения в файлы репозитория; [история]{.underline} -- какие изменения вносились в изначальный файл в процессе разработки; [рабочая копия]{.underline} -- копия файла, с которой работает человек

****3. Централизованные -- существует центральный репозиторий для всех пользователей, откуда они файлы утаскивают и обратно вставляют: CVS, Subversion; ****

Децентрализованные -- центрального репозитория нет: Git, Bazaar, Mercurial

****4. То же самое, что в пункте пять, только обновляться до последней версии репозитория не надо ****

5. Создать репозиторий/обновиться до последней версии репозитория>получить файлы из репозитория>внести изменения в локальные версии этих файлов>залить изменения в общий репозиторий

6. Возможность отката и одновременной работы над несколькими версиями

7. создание основного дерева репозитория:

git init

-- получение обновлений (изменений) текущего дерева из центрального репозитория:

git pull

-- отправка всех произведённых изменений локального дерева в центральный репозиторий:

git push

-- просмотр списка изменённых файлов в текущей директории:

git status

-- просмотр текущих изменений:

git diff

-- добавить все изменённые и/или созданные файлы и/или каталоги:

git add .

-- добавить конкретные изменённые и/или созданные файлы и/или каталоги:

git add имена_файлов

-- удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории):

git rm имена_файлов

-- сохранить все добавленные изменения и все изменённые файлы:

git commit -am 'Описание коммита'

8. Вопрос не понятен.

9. Для параллельной работы с разными версиями программы.

10. Внести расширения игнорируемых файлов в репозиторий gitignore-ом. Чтобы не загружать временные и объектные файлы на сервер.