САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет _____Компьютерных технологий и управления__
(название факультета)
Кафедра _____Вычислительной техники_____
(название кафедры)
Направление подготовки (специальность) ____Информатика и вычислительная техника_____

О Т Ч Е Т

о практике

Тема задания: «»

Студент ____Казанцева А.И. _____3105_____
(Фамилия И.О.) (номер группы)

Руководитель практики _____Соснин В. В._____
(Фамилия И.О., должность и место работы)

Оценка, рекомендуемая руководителем _____

Практика пройдена с оценкой _____

Подписи членов комиссии
_____(_____)
(Фамилия И.О.)
_____(_____)
(Фамилия И.О.)
_____(_____)
(Фамилия И.О.)

Дата _____

Санкт-Петербург
2014г.

Содержание:

Система контроля версий Git

Существует несколько типов систем контроля версий:

- Локальные – имеется простая база данных, в которой хранятся версии необходимых файлов.
- Централизованные – имеется сервер, на котором хранятся все файлы, клиенты получают копии этих файлов.(прим. CVS)
- Распределенные – в отличии от централизованных, клиенты получают копию всего репозитория, при потере данных, можно восстановить их с помощью клиентской копии. К такому типу СКВ относится Git

Основным отличием Git от других систем контроля версий, таких как CVS, является система хранения данных. Вместо патчей (изменений) используются наборы слепков. При коммите сохраняется слепок того, как выглядят файлы в данный момент(если файл не изменялся, то используется ссылка на его предыдущую версию).

Три состояния файлов: зафиксированный, измененный, подготовленный.

Процесс работы:

1. Внести изменения в файлы
2. Подготовка файлов (индексация)
3. Коммит (подготовленные файлы перемещаются в каталог Git'а на постоянное хранение)

Изученные команды Git:

git init - создание репозитория

git add <filename> - индексация файла (указывается, что файл изменен и должен быть включен в следующий коммит)

git commit – производит коммит проиндексированных файлов

git clone <url>– происходит копирование всех данных с сервера клиенту

git status – определение состояния файлов

git diff – показывает конкретные различия в файлах

git rm <filename>– удалить файл из индекса(при использовании ключа –f принудительное удаление)

git mv <from> <to>- перемещение файла (по сути происходит создание нового файла, в указанном месте, и удаление старого)

git log – просмотр истории, при использовании разных ключей возможны разные форматы получения данных, за разный период, разное количество коммитов и т.д.

git revert HEAD – отмена последнего коммита

git checkout -- <file> - отмена изменений файла (используется до индексации)

git reset HEAD <file> - отмена индексации

Метки(tag) – позволяют помечать определенные моменты в истории, как важные

Бывают легковесные – указатель на коммит(сохраняется только контрольная сумма коммита) и аннотированные(кроме суммы хранятся сведения об авторе, дате и т.д.)

Изученные команды для работы с метками:

git tag – просмотр всех меток

git tag v1.5-lw – создание легковесной метки

git tag –a v1.5 - создание аннотированной метки

Ветка – подвижный указатель на один из коммитов, по умолчанию ветка называется master, указывает на последний сделанный коммити передвигается автоматически

git branch <name> - создание новой ветки

git checkout <name> - переход на другую ветку

git merge <name> - слияние веток

git branch –d <name> -удаление ветки

Конфликты при слиянии веток возможны при изменении одной и той же части кода по-разному(нужно изменить конфликтующие файлы и проиндексировать)

Ссылки на использованные материалы:

1. http://git-scm.com/book
2. http://githowto.com/ru/
3. http://pcottle.github.io/learnGitBranching/