РАБОТА №7 ЛАБОРАТОРНАЯ. НАСТРОЙКИ ДОСТУПА К РЕПОЗИТОРИЮ

* 1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Целью работы является получение практических навыков ра- боты с депозитарием проекта Github.

* 1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Git** – распределѐнная система управления версиями. Проект был создан для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года.

**GitHub** – крупнейший[[7]](https://ru.wikipedia.org/wiki/GitHub#cite_note-7) веб-сервис для хостинга IT-проектов и их совместной разработки.

Среди проектов, использующих Git – ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt, ряд дистрибутивов Linux.

Программа является свободной и выпущена под лицензией GNU GPL версии 2. По умолчанию используется TCP порт 9418.

Ядро Git представляет собой набор утилит командной строки с параметрами. Все настройки хранятся в текстовых файлах конфигу- рации. Такая реализация делает Git легко портируемым на любую платформу и даѐт возможность легко интегрировать Git в другие системы (в частности, создавать графические git-клиенты с любым желаемым интерфейсом).

Репозиторий Git представляет собой каталог файловой систе- мы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов, и хранилище, содер- жащее собственно файлы. Структура хранилища файлов не отража- ет реальную структуру хранящегося в репозитории файлового дере- ва, она ориентирована на повышение скорости выполнения опера- ций с репозиторием. Когда ядро обрабатывает команду изменения (неважно, при локальных изменениях или при получении патча от другого узла), оно создаѐт в хранилище новые файлы, соответст- вующие новым состояниям изменѐнных файлов. Существенно, что

никакие операции не изменяют содержимого уже существующих в хранилище файлов.

По умолчанию репозиторий хранится в подкаталоге с названи- ем «.git» в корневом каталоге рабочей копии дерева файлов, храня- щегося в репозитории. Любое файловое дерево в системе можно превратить в репозиторий git, отдав команду создания репозитория из корневого каталога этого дерева (или указав корневой каталог в параметрах программы). Репозиторий может быть импортирован с другого узла, доступного по сети. При импорте нового репозитория автоматически создаѐтся рабочая копия, соответствующая послед- нему зафиксированному состоянию импортируемого репозитория (то есть не копируются изменения в рабочей копии исходного узла, для которых на том узле не была выполнена команда commit).

В первую очередь надо установить клиент git: обязательно потре- буется консольный клиент, доступный по ссылке [http://git-](http://git-/) scm.com/downloads (поддерживаются основные ОС), графический кли- ент можно установить по желанию, исходя из своих предпочтений.

Далее работа с git будет объясняться на примере работы с кон- сольным клиентом по следующим причинам:

* Чтобы складывалось понимание происходящего и при воз- никновении проблем вы могли четко объяснить, что вы делали, и было видно, что пошло не так.
* Все нажатия кнопок в графических клиентах в итоге сводят к выполнению определѐнных команд консольного клиента, в то же время возможности графических клиентов ограничены по сравне- нию с консольным
* У тех, кто будет работать в классе на стоящих там компью- терах, не будет другого выбора, кроме как пользоваться консоль- ным клиентом (на сколько мне известно, никаких графических кли- ентов для git там не установлено)

***Аккаунт и репозитории на github.com.*** Для того что бы ис- пользовать web-сервис репозитория необходимо зарегистрироваться на https://github.com/. После чего можно будет создавать свои репо- зитории или присоединиться к работе над проектами коллег, сделав копию (fork) другого репозитория. Вам предлагается начать с соз- дания fork-а к заведенному репозиторию https://github.com/andreiled/mipt-cs-4sem.

***Создание локального репозитория, связанного с удаленным репозиторием.*** Следующим шагом после создания репозитория на github, называемого далее удаленным репозиторием, является соз- дание локальной копии этого репозитория на своем компьютере. Особенностью git является наличие на локальном компьютере пол- ной копии репозитория со всей информацией об истории измене- ний.

1. Открываем консольный клиент.

* На Windows после установки клиента появляется пункт Git Bash в контекстом меню папки. Достаточно перейти в желаемую папку и воспользоваться этим пунктом меню.
* На Unix системах достаточно открыть терминал и перейти в нужную директорию. При стандартной установке консольного кли- ента будет доступна команда git без дополнительных усилий.

1. Выполняем команду git clone https://github.com/%user\_login%/%repo\_name%.git. Полную https ссылку на репозиторий для его выкачивания можно также найти на странице самого репозитория на github. После этого в текущей пап- ке появится новая папка с именем %repo\_name%, содержащая ко- пию удаленного (remote) репозитория.
2. Переходим в созданную папку репозитория и настраиваем

его:

1. git config user.name ivan.ivanov
2. git config user.email [ivanov@example.com](mailto:ivanov@example.com)

Внесение и оформление изменений в локальном репозито-

***рии.*** Воспользовавшись командой git status можно узнать, на какой ветке (branch) репозитория вы сейчас находитесь, какие изменения присутствуют в вашей рабочей копии и другую информацию.

*Рабочей копией называется совокупность файлов в локальной папке репозитория за исключением служебных файлов.*

После внесения каких-либо изменений в рабочую копию их можно «закоммитить» в локальный репозиторий:

* сначала нужная часть изменений подготавливается к ком- миту с использованием команды git add %file\_path%
* после чего производится коммит командой git commit Использование команды без аргументов откроет текстовый редак- тор, где надо будет написать комментарий для коммита, коммит обязательно должен иметь комментарий. Другим вариантом задания

комментария к коммиту является использование команды git commit -m «%commit\_message%»

Историю изменений можно посмотреть командой git log или git log --name-only. Если вся история изменений не умещается на экране, то можно пользоваться клавишами прокрутки на клавиатуре («стрелочки», PgUp, PgDown), выход из режима просмотра измене- ний осуществляется нажатием клавиши «q».

*Загрузка локальных изменений в удаленный репозиторий.* По- сле того, как были выполнены нужные локальные коммиты, изме- нения можно загрузить в удаленный репозиторий с помощью ко- манды git push origin master. GIT клиент при этом запросит имя пользователя и пароль для доступа к github.

Выполнение этой команды может закончиться с ошибкой, если в локально репозитории отсутствуют последние изменения, имею- щиеся в удаленном репозитории. Для решения этой проблемы надо выполнить команду git pull, которая скачает последние изменения из удаленного репозитория и смержит их с вашими локальными правками, после чего можно повторить команду git push.

* 1. ПОРЯДОК ВЫПОЛНЕНИЯ И ЗАДАНИЕ ДЛЯ РАБОТЫ
     1. Зарегистрируйтесь на сервисе https://github.com/.
     2. Создайте локальный репозитарий, связанный с заданным глобальным репозитарием.
     3. Настройте параметры доступа к локальному репозитарию.
     4. Выполните загрузку изменений выполненных в локальном репозитарии в глобальный репозитарий.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие функции выполняет система контроля версиями?
2. Что такое локальный репозиторий?
3. Какие функции выполняет сервис **GitHub**?
4. Какие действия нужно выполнить для получения доступа к функционалу **GitHub?**
5. Каким образом изменения в локальном репозитории GIT можно передать в глобальный репозитории **GitHub?**