Руководство по использованию системы контроля версий Git, сервисов GitHub и GitHub Classroom для выполнения лабораторных работ

Шаг 1. Установка Git

Необходимо установить программное обеспечение для работы с системой контроля версий Git в той операционной системе, в которой будет выполняться лабораторная работа. Например, в случае выполнения лабораторной работы в ОС Linux, которая установлена на виртуальной машине, запущенной в ОС Windows, необходимо установить программное обеспечение для работы с Git в гостевой ОС Linux, а не на основной (хостовой) ОС Windows.

На Ubuntu установить Git можно с помощью команды **apt-get install git**. Команды для установки Git для других дистрибутивов Linux, установщики для Windows и Mac OS можно найти на официальном сайте системы контроля версий Git: https://git-scm.com.

Шаг 2. Регистрация на сайте github.com

В случае, если учетная запись на GitHub пользователем ранее не создавалась, необходимо перейти на сайт https://github.com и зарегистрироваться.

Шаг 3. Открыть ссылку с индивидуальным заданием

Для получения доступа к индивидуальному репозиторию с заданием следует перейти по ссылке на GitHub, выданной преподавателем. В случае если пользователь не авторизован на сайте, откроется окно авторизации, показанное на рисунке 1. В этом окне необходимо авторизоваться (ввести логин и пароль), либо перейти к предыдущему шагу и зарегистрироваться на GitHub, если он еще не был выполнен, нажав на ссылку «Create an account».

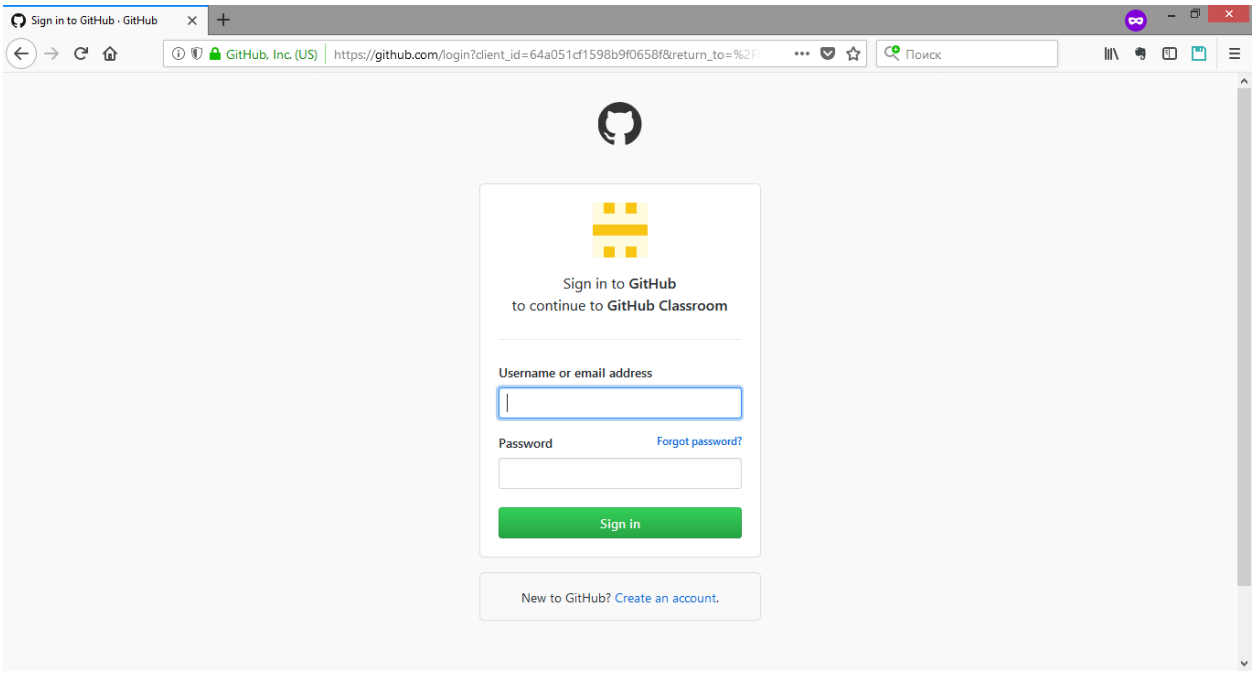

Рисунок 1. Окно авторизации

Если пользователь уже авторизован на GitHub, после перехода по ссылке на GitHub, выданной преподавателем, следует перейти к следующему шагу.

Шаг 4. Инициализация репозитория

После успешной авторизации GitHub создаст для авторизовавшегося пользователя индивидуальный репозиторий с заданием. Появится окно, показанное на рисунке 2. Далее необходимо перейти по ссылке на репозиторий с заданием, расположенной после слов «Your assignment has been created here».

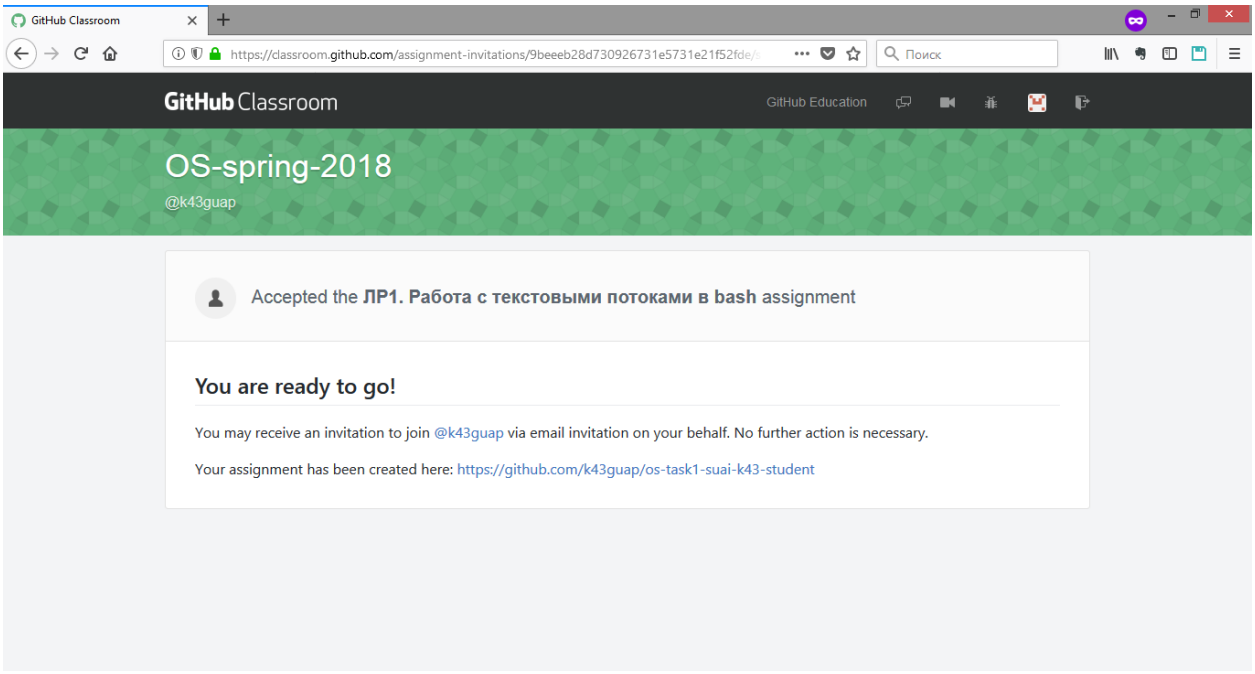

Рисунок 2. Процесс инициализации репозитория завершен

Шаг 5. Подключение системы автоматического тестирования

Сборка и тесты будут запускаться автоматически при каждом новом коммите в репозиторий. Тем не менее не следует злоупотреблять этим процессом и полагаться только на автоматические тесты. Перед каждым коммитом следует запускать тесты самостоятельно на своей машине, чтобы убедиться, что добавляемый в репозиторий код является полностью работоспособным.

Так же следует иметь в виду, что в некоторых лабораторных автоматическое тестирование может подключаться к репозиторию не сразу. Если в течение суток после создания репозитория тесты не заработали, следует обратиться к преподавателю по электронной почте или лично. В случае обращения по электронной почте обязательно укажите ФИО, номер группы и ссылку на репозиторий.

Шаг 6. Клонирование репозитория

Окно с открытым репозиторием на GItHub показано на рисунке 3. Для выполнения задания на лабораторную работу репозиторий необходимо клонировать (создать локальную рабочую копию) в операционную систему пользователя. Дальнейшая работа будет вестись уже с этой локальной рабочей копией репозитория.

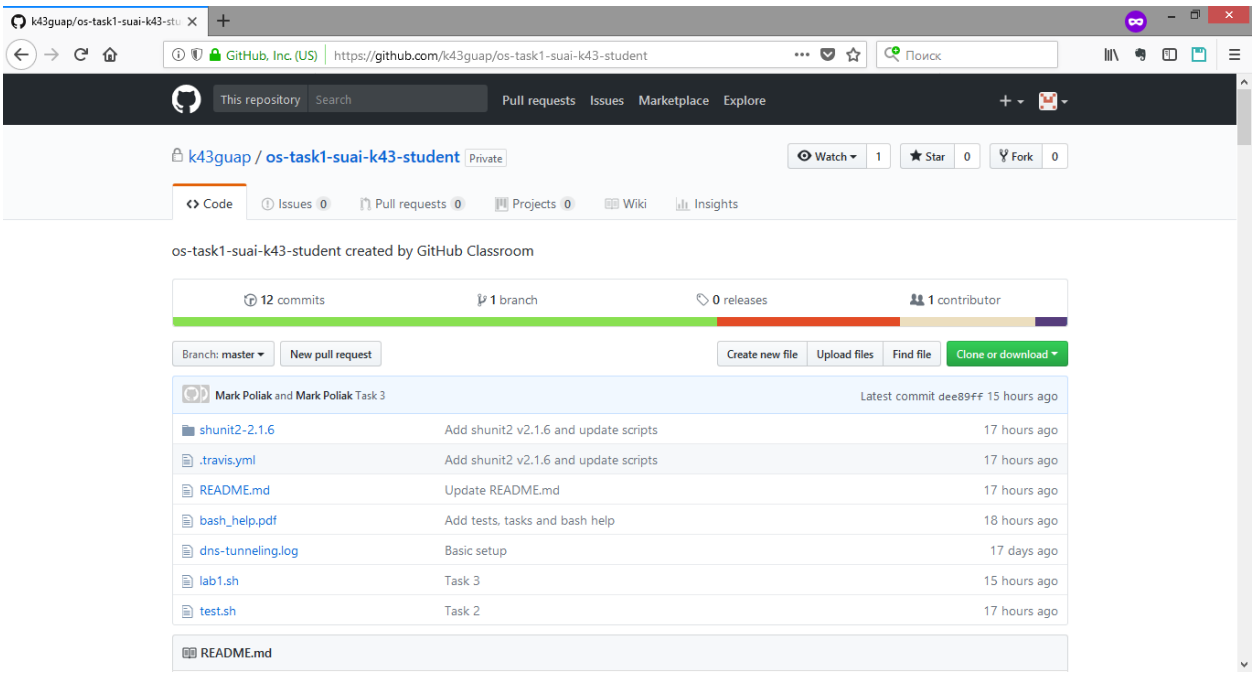

Рисунок 3. Репозиторий с заданием на лабораторную работу

Для того, чтобы клонировать репозиторий необходимо нажать на зеленую кнопку «Clone or download» на главной странице репозитория на GitHub, как показано на рисунке 4. В открывшемся всплывающем окошке необходимо скопировать ссылку, нажав на кнопку «Copy to clipboard».

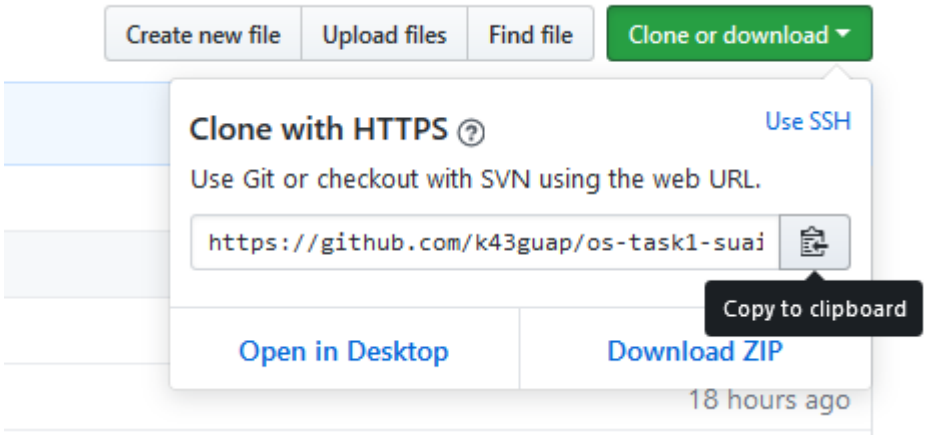

Рисунок 4. Получение ссылки для клонирования репозитория

Далее необходимо открыть терминал (командный интерпретатор) в той ОС, где будет создаваться копия репозитория. В терминале следует сменить текущую директорию на ту, где будет располагаться копия репозитория, а затем ввести команду **git clone <ссылка_на_репозиторий>**. Вместо **<ссылка_на_репозиторий>** следует вставить только что скопированную ссылку со страницы репозитория на GitHub. При выполнении команды клонирования репозитория, в зависимости от настроек, может появиться запрос имени пользователя и пароля от аккаунта на GitHub, после чего начнется копирование файлов. Пример выполнения указанной команды в окне терминала ОС Ubuntu показан на рисунке 5.

Рисунок 5. Клонирование репозитория

Шаг 7. Работа с локальной копией репозитория

После того, как репозиторий был клонирован на локальную машину (создана локальная копия репозитория), в текущей директории появится папка с именем репозитория. В ней и находится локальная копия репозитория. Необходимо перейти в эту папку, выполнив команду **cd <имя_репозитория>**.

Далее следует выполнить задание на лабораторную работу, отредактировав имеющиеся в локальной копии репозитория файлы и при необходимости добавив новые. По завершении выполнения задания необходимо сохранить сделанные изменения в локальной копии репозитория, выполнив команды:

git add <список_файлов_через_пробел>

git commit

Первая команда, **git add …**, указывает, изменения в каких файлах необходимо сохранить в репозитории. Вторая команда, **git commit**, сохраняет изменения (создает «точку восстановления») во всех ранее указанных файлах. С помощью команды **git status** можно проверить, какие файлы были изменены, удалены или добавлены после выполнения последней команды **git commit**. При выполнении команды **git commit** потребуется ввести текстовое сообщение, описывающее внесенные в файлы изменения. Сообщение должно быть коротким, но в то же время информативным. Пример выполнения команд показан на рисунке 6.

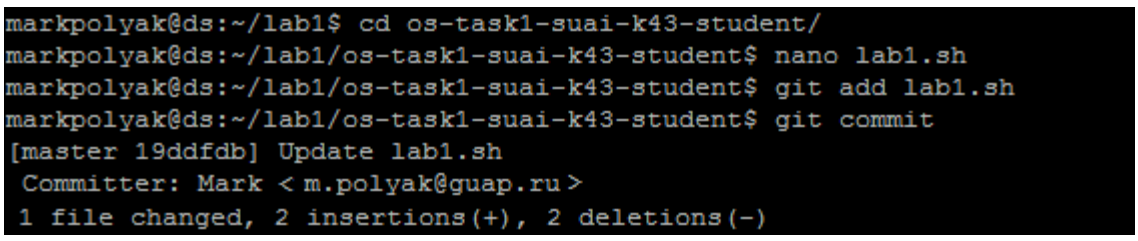

Рисунок 6. Работа с локальной копией репозитория

Шаг 8. Синхронизация локальной копии репозитория с GitHub

После успешного выполнения команды **git commit** необходимо синхронизировать локальную копию репозитория с основным репозиторием на GitHub. Для этого используется команда **git push**. При ее выполнении вновь может потребоваться ввести имя пользователя и пароль учетной записи на GitHub. Пример показан на рисунке 7.

```
markpolyak@ds:~/lab1/os-task1-suai-k43-student$ git push
Username for 'https://github.com': m.polyak@guap.ru
Password for 'https://m.polyak@guap.ru@github.com':
Counting objects: 3, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/k43guap/os-task1-suai-k43-student.git
   dee89ff..19ddfdb  master -> master
markpolyak@ds:~/lab1/os-task1-suai-k43-student$
```

Рисунок 7. Синхронизация локальной копии репозитория с GitHub

Шаг 9. Автоматическая сборка и тестирование

После синхронизации локальной версии репозитория с GitHub должны автоматически запуститься тесты. В главном окне репозитория на GitHub необходимо открыть раздел «commits», как показано на рисунке 8.

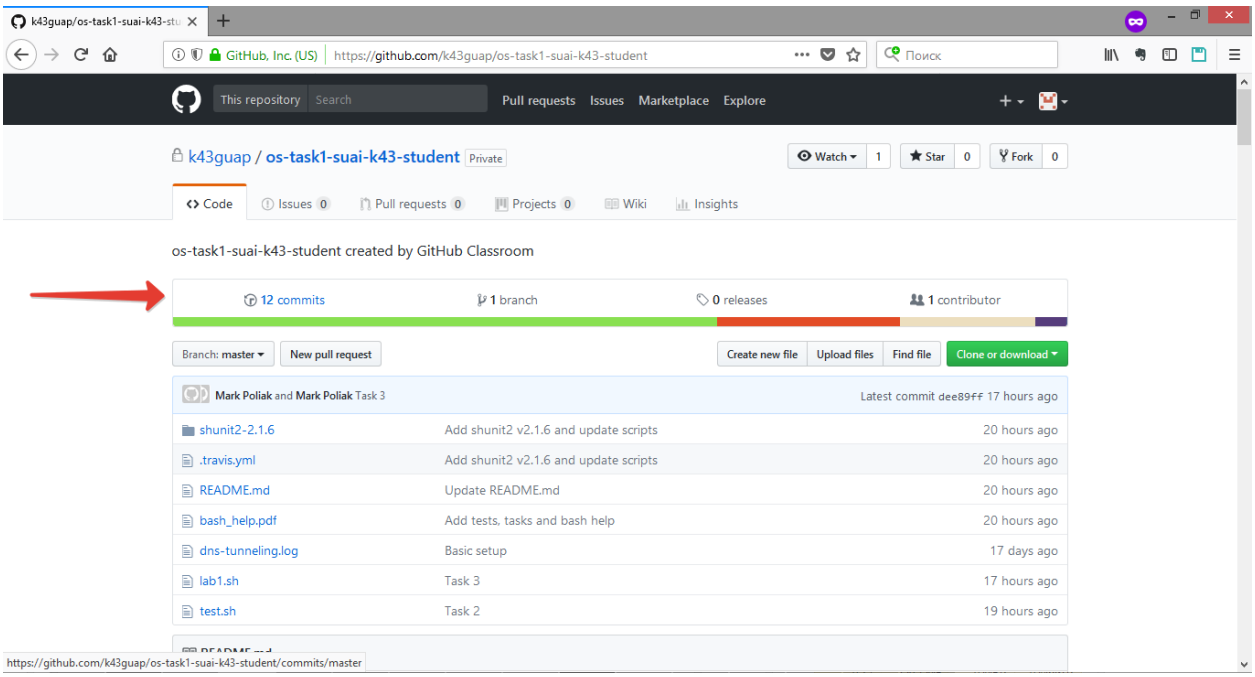

Рисунок 8. Раздел «commits»

В разделе «commits» репозитория на сайте GitHub отображаются все когда-либо сделанные коммиты в данный репозиторий любыми пользователями. При этом последние коммиты отображаются сверху. На рисунке 9 показаны результаты выполнения тестов для трех коммитов. Красный крестик под текстовым описанием коммита обозначает, что код в этом коммите не прошел тесты. Зеленая галочка там же означает, что тесты пройдены успешно. В этом случае можно приступить к подготовке отчета по лабораторной работе. Желтый кружок обозначает, что тесты запущены и еще выполняются, в этом случае стоит подождать. Наконец, если в течение пяти

минут напротив текстового описания последнего коммита не появилось ни одного из трех вышеперечисленных графических символов (в т.ч. отсутствует и желтый кружок), это означает, что автоматические тесты к репозиторию не подключены. В этом случае следует подождать, а при длительной задержке с подключением автоматического тестирования следует связаться с преподавателем в соответствии с алгоритмом из шага 5.

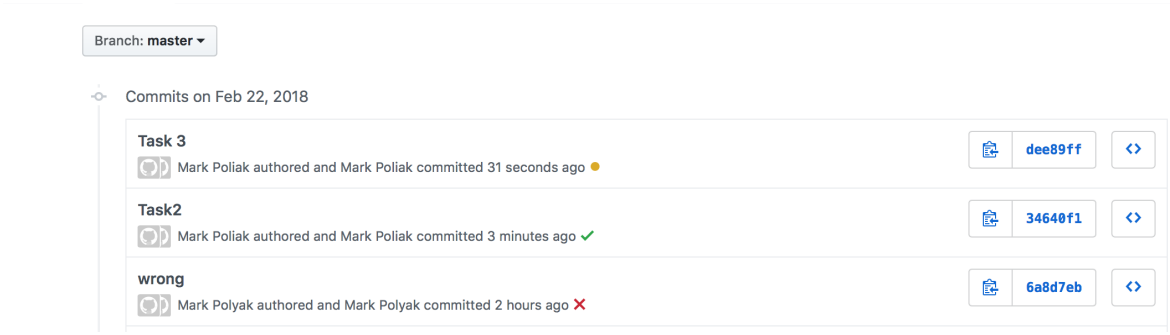

Рисунок 9. Автоматическое выполнение тестов