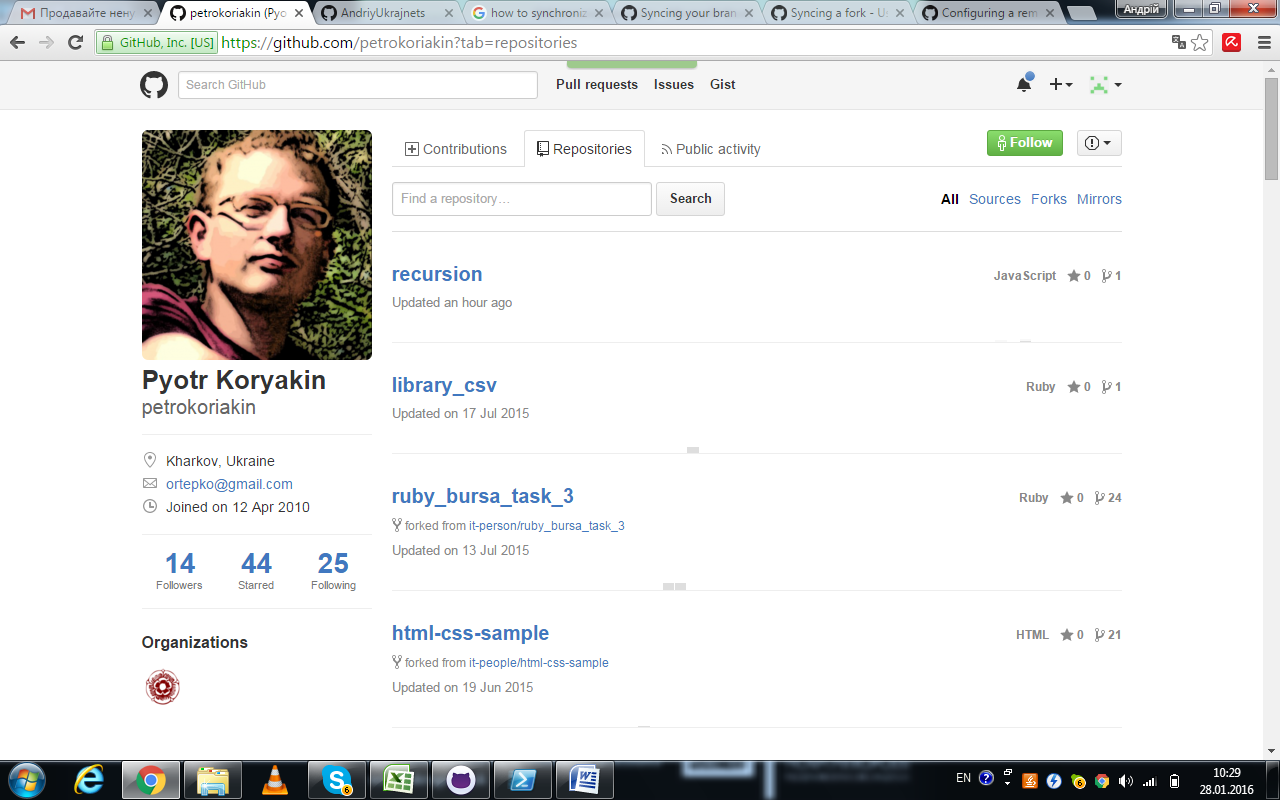
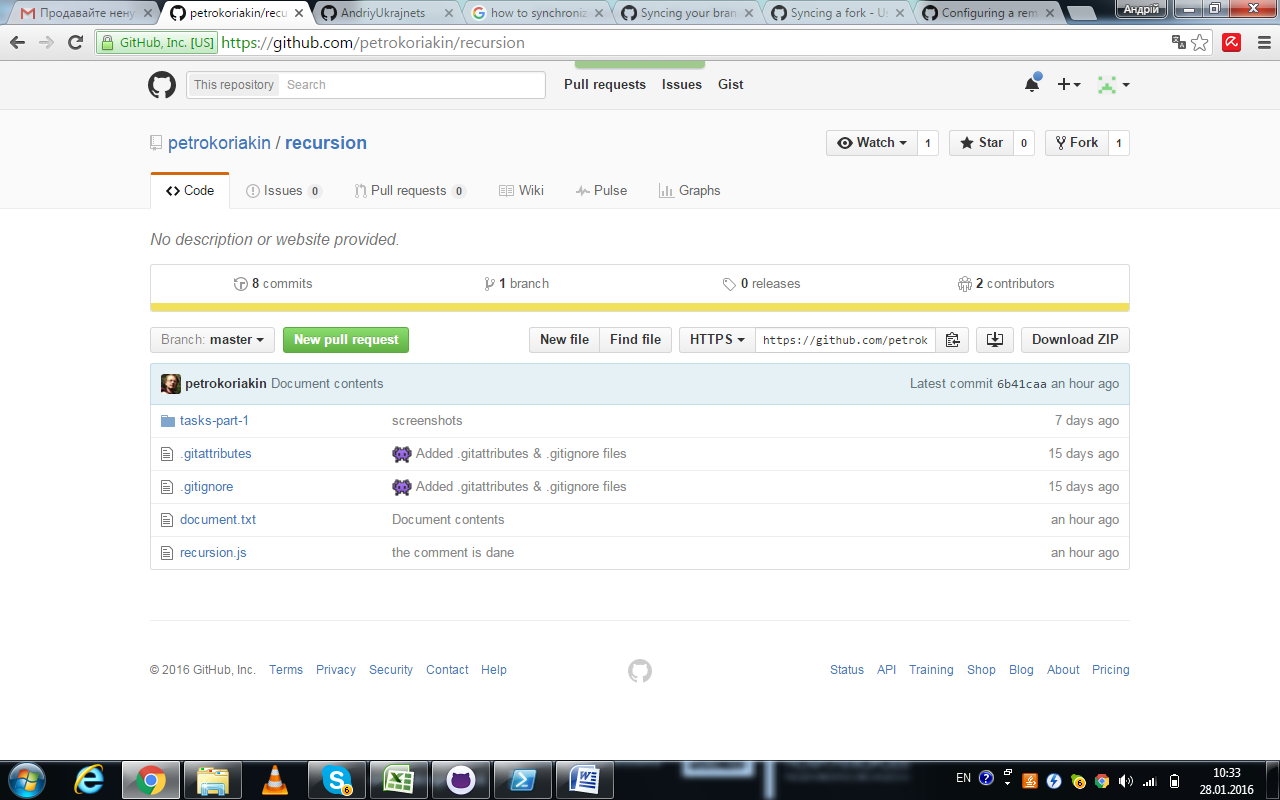
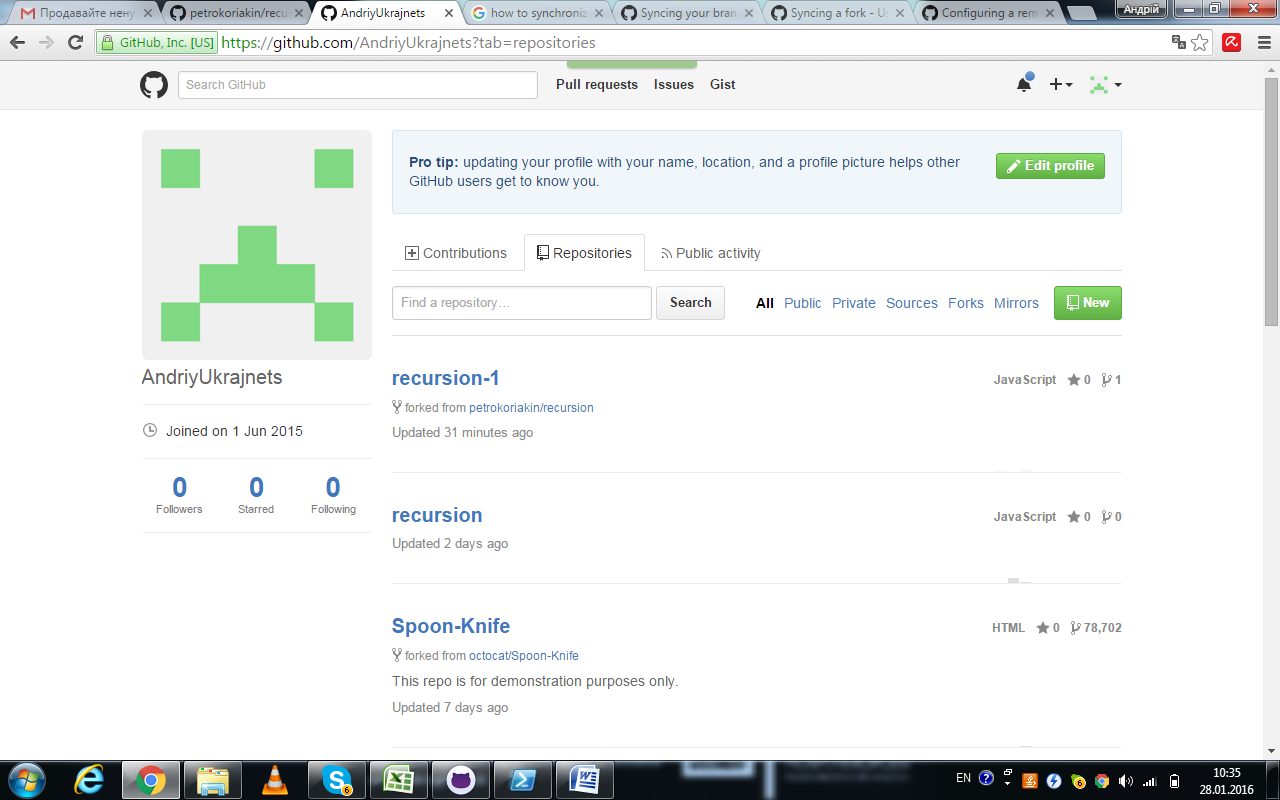
**В этом примере мы используем симбиоз способов работы над проектом – Github, декстопное приложение Github и консоль Git Shel.**

1. **Копирование репозитория.**
2. Заходим к пользователю, репозиторий которого ходим скопировать (форкнуть). В данном случае будем работать над файлом recursion, одного хорошего программиста.
3. Заходим в этот репозиторий, вверху справа жмем Fork. Т.о., мы скопировали себе этот репозиторий.



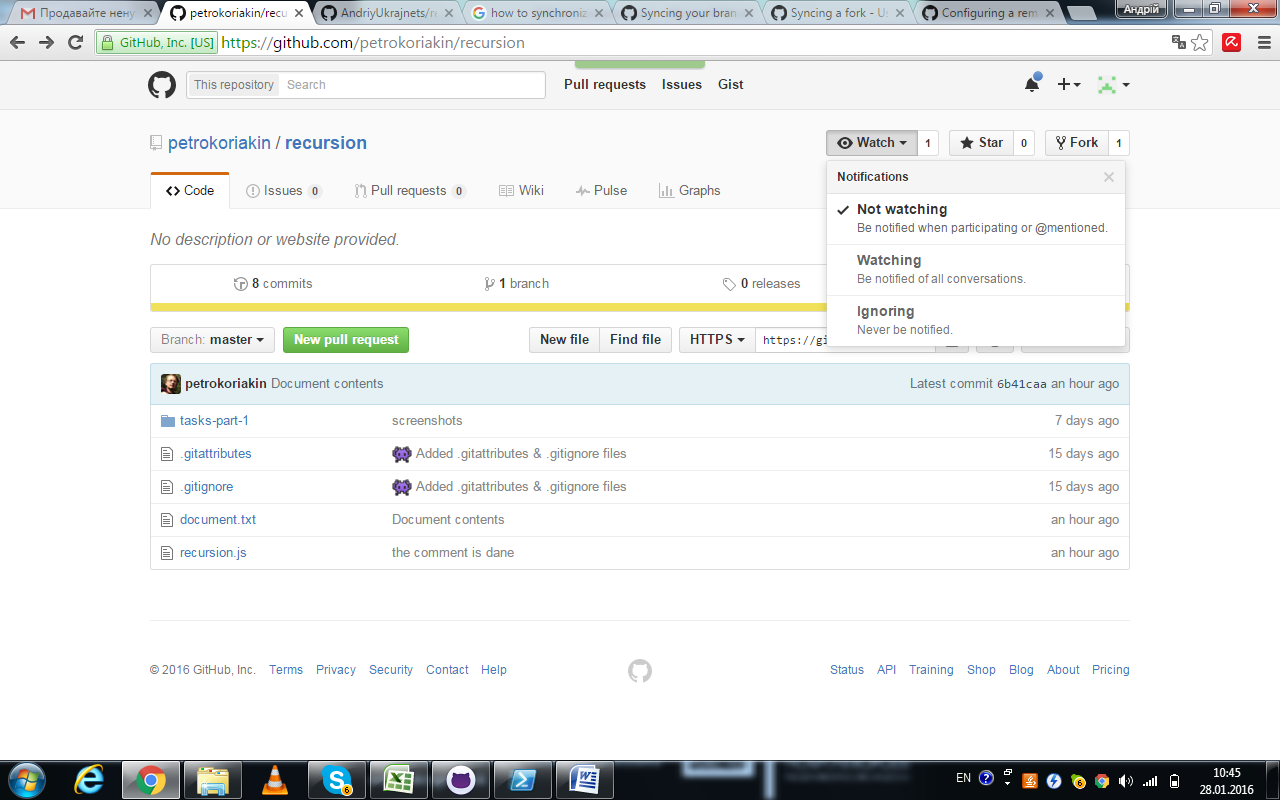
Поскольку репозиторий с таким названием у меня уже был, Github создаст у меня файл recursion-1



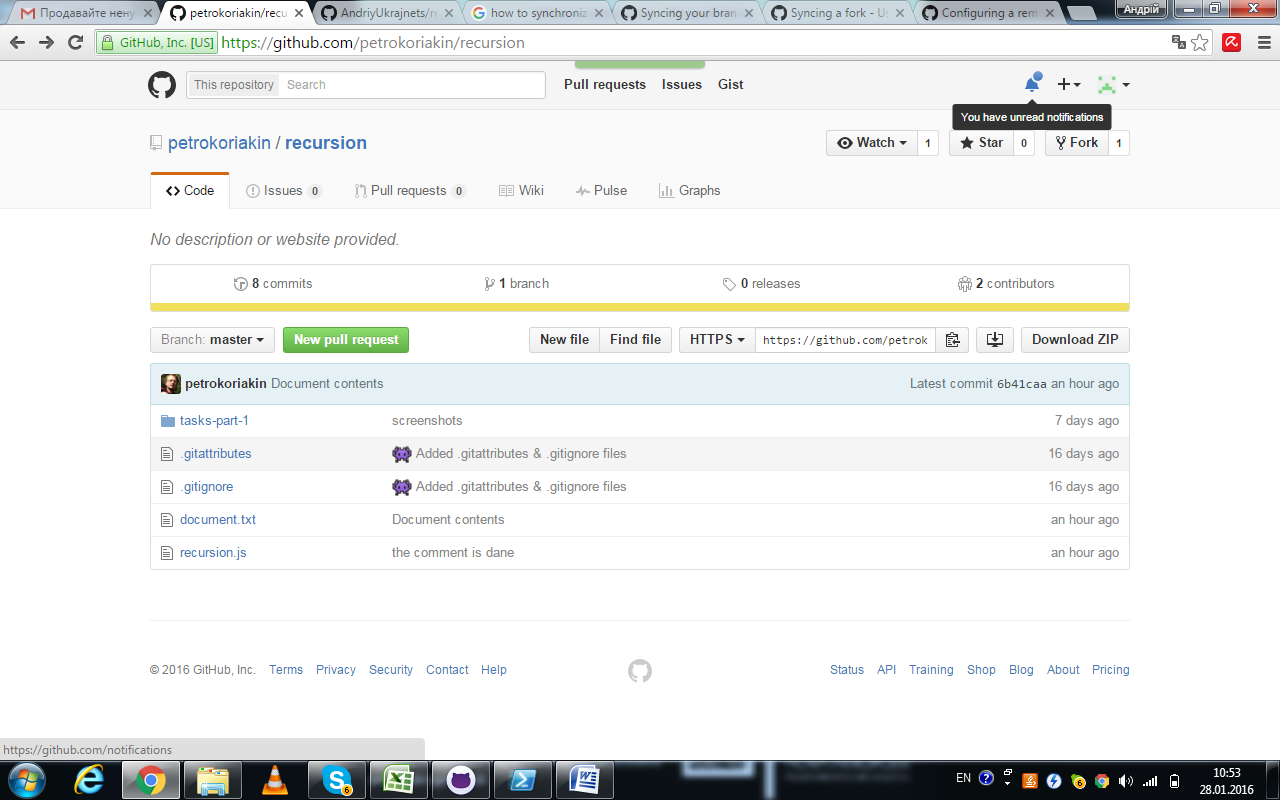
1. **Отслеживание изменений.**

Чтоб быть в курсе изменений, которые были сделаны в этом (родительском) репозитории можем:

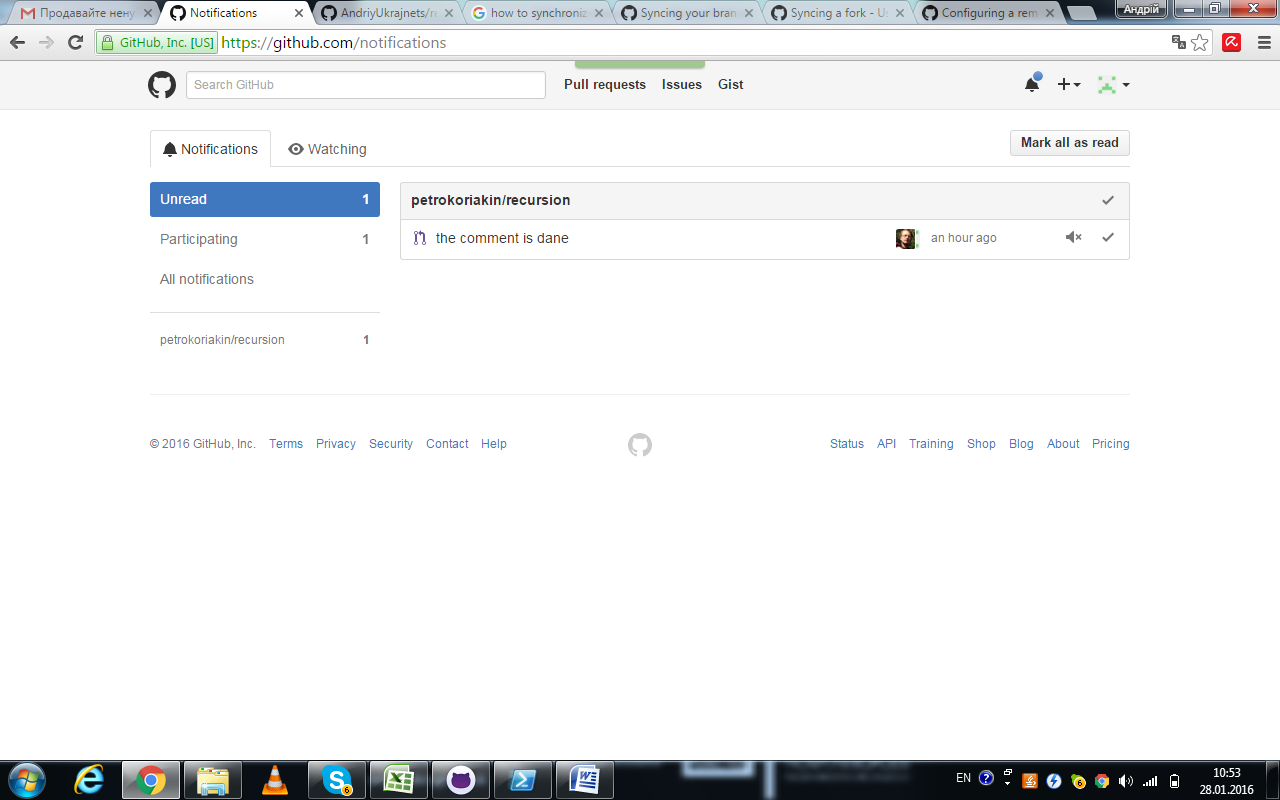
1. Вверху справа выбрать Watch, далее – Watching.
2. Можем просто зайти в этот репозиторий и посмотреть, когда и в каком файле были сделаны соответствующие изменения. В данном случае я вижу, что файл Document.txt был изменен около 1 часа назад.



1. Пометка на иконке “колокольчик” вверху справа – если она с синим кругом, значит есть какие-то изменения в проектах, над которыми работаем и которые фолловим (за которыми следим).

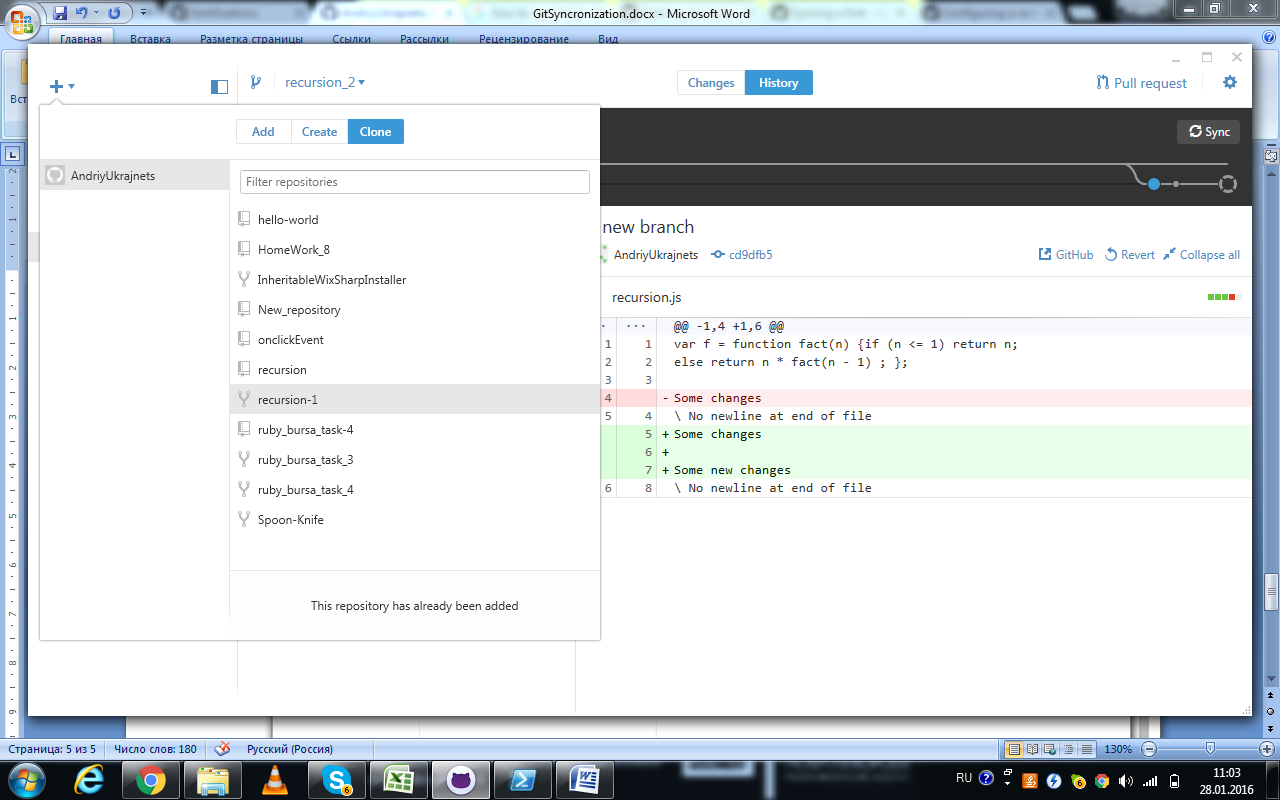


Когда я кликну на колокольчик, он покажет коммент, который был сделан тем хорошим программистом. Но тут может быть вообще большой список файлов, в которые были внесены какие-то изменения.

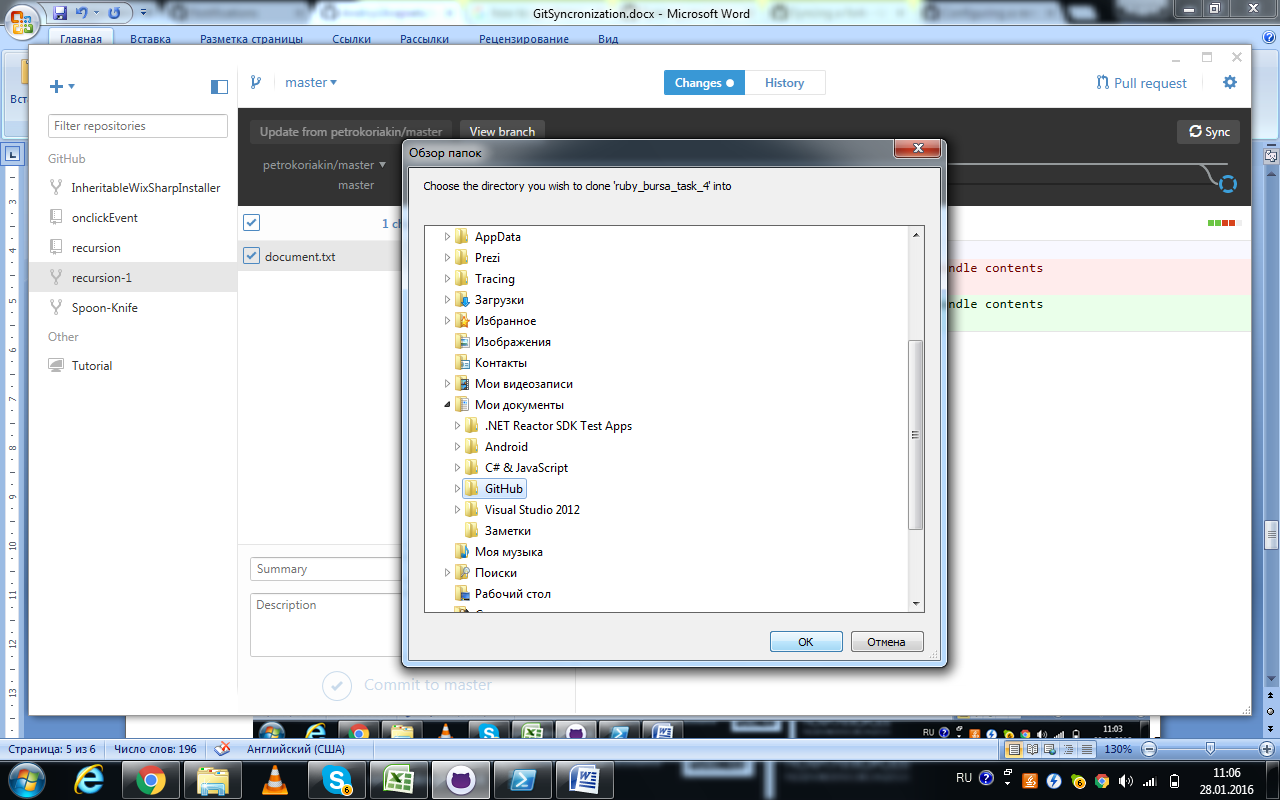


С данного момента Github создает такую связку между мной и PetroKoriakin, и будет держать нас в курсе тех изменений, которые происходят в проекте

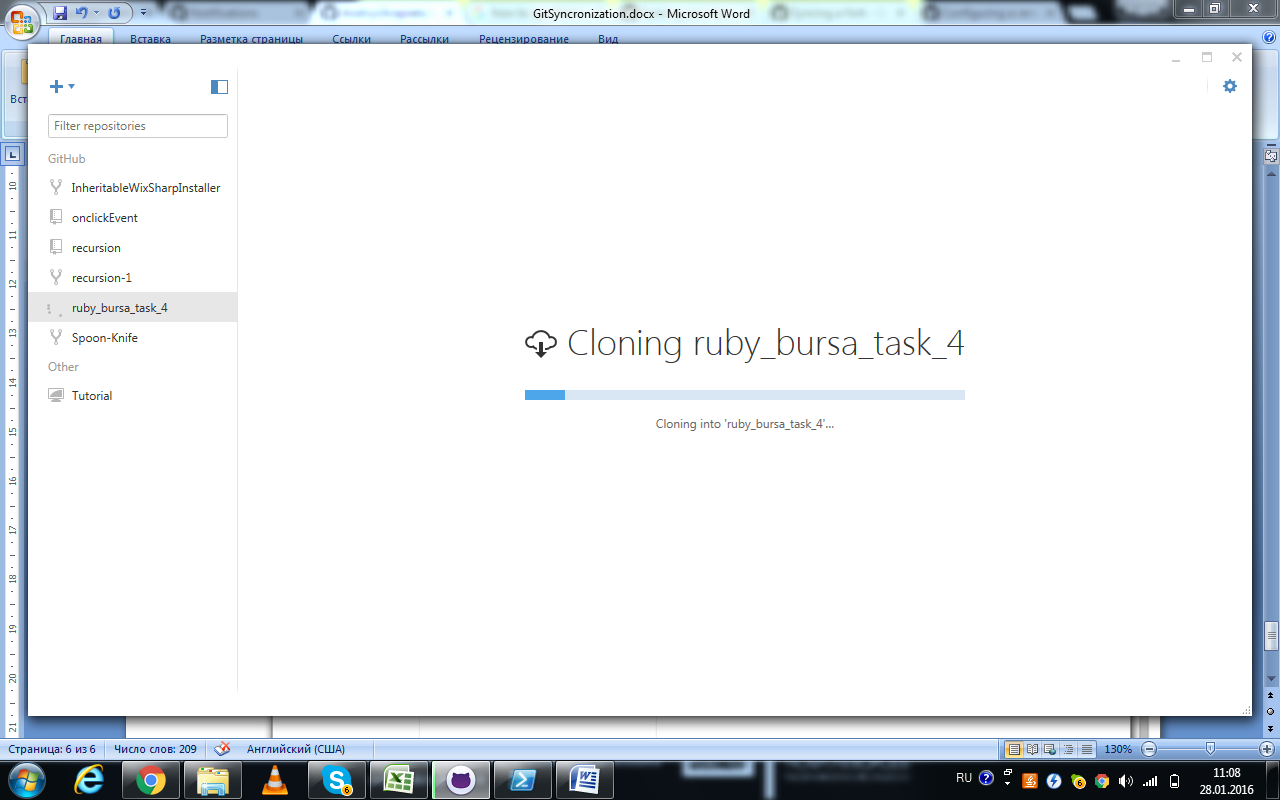
1. **Копирование репозитория на локальную машину.**
2. Заходим в десктопное приложение GitHub, вверху слева жмем плюс, из списка возможных для копирования репозиториев выбираем нужный. В данном случае я выбираю recursion-1. Жмем Clone.



1. В окне “Обзор папок”, выбираем куда копировать, - в данном случае в папку Github.



Мы должны увидеть окно – копирование репозитория (Cloning …….)



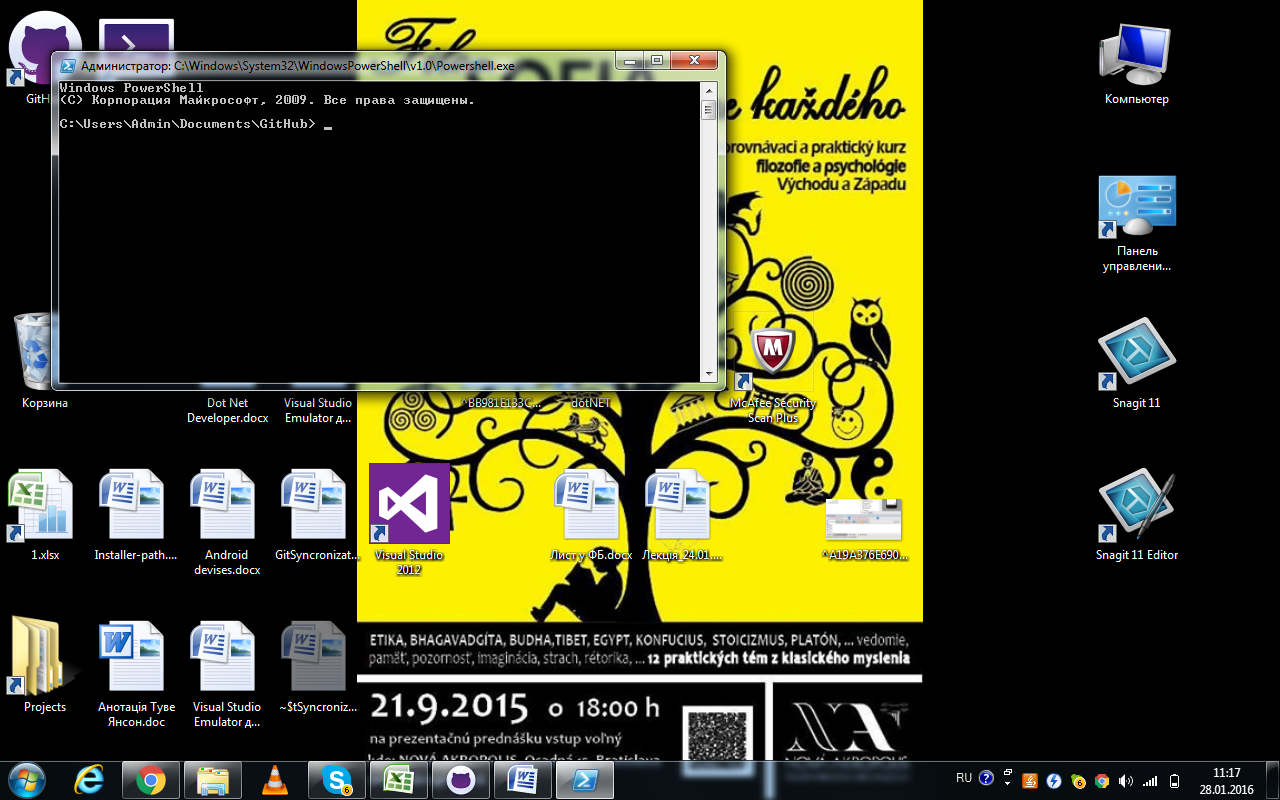
1. **Работа с консолью**

В данном моменте откроем косоль. У вас должны быть доступны на рабочем столе, и в меню запуска иконки GitHub (десктопное приложение) и Git Shel (консоль).

Для чего нужна консоль? Многие команды недоступны из GitHub (десктопное приложение) или из github.com. В Git Shel (консоль) возможности шире.



Заходим в Git Shell. Вы увидите путь к папке в которую Github переносит файлы:



Прежде, давайте погуляем тут по консолию

1. Зайти в папку:

*> cd .\foldername*

В данном случае foldername это recursion. Зайдем в репозиторий recursion

Если были созданы еще какие-то ветки этого репозитория в квадратных скобках мы увидим эти ветки

1. Посмотреть отличия между файлами (отслеживание внесенных недавно изменений):

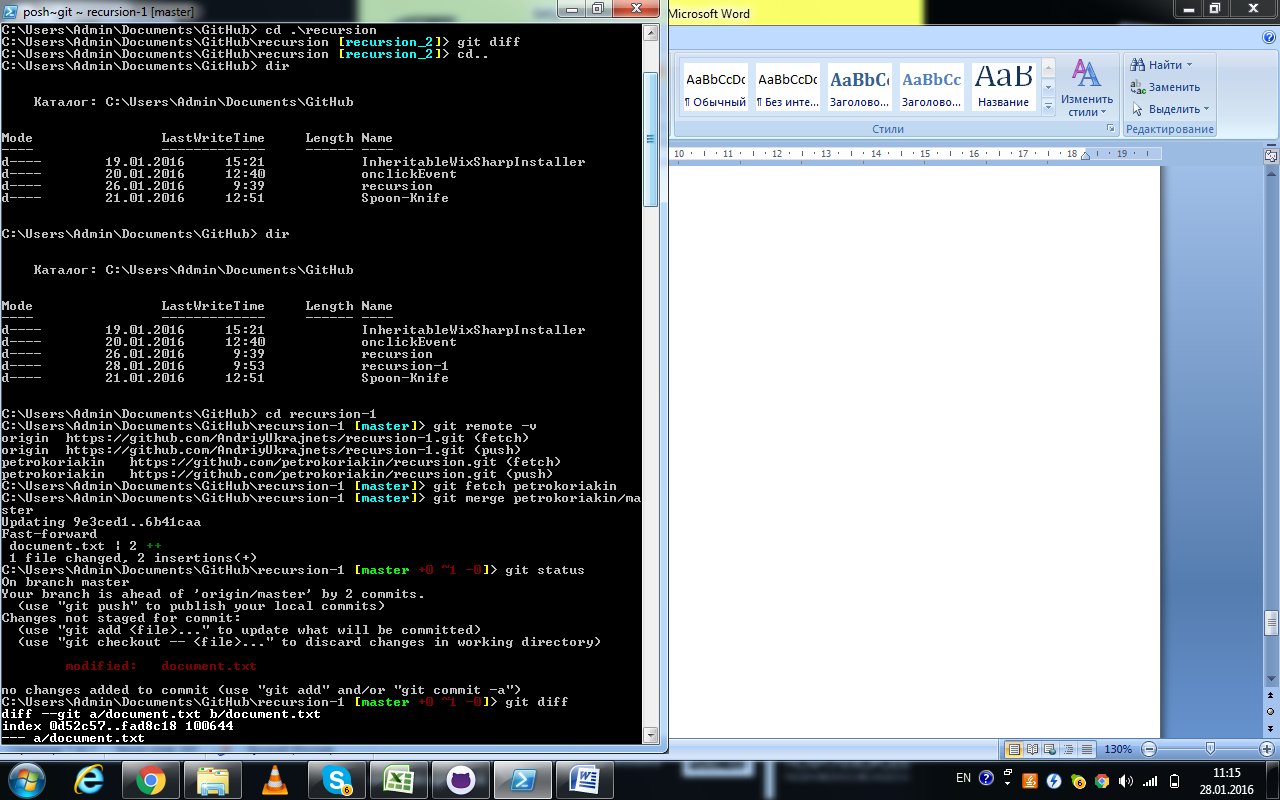
*> git diff*

1. Выйти из папки, поскольку нас интересует репозиторий recursion-1:

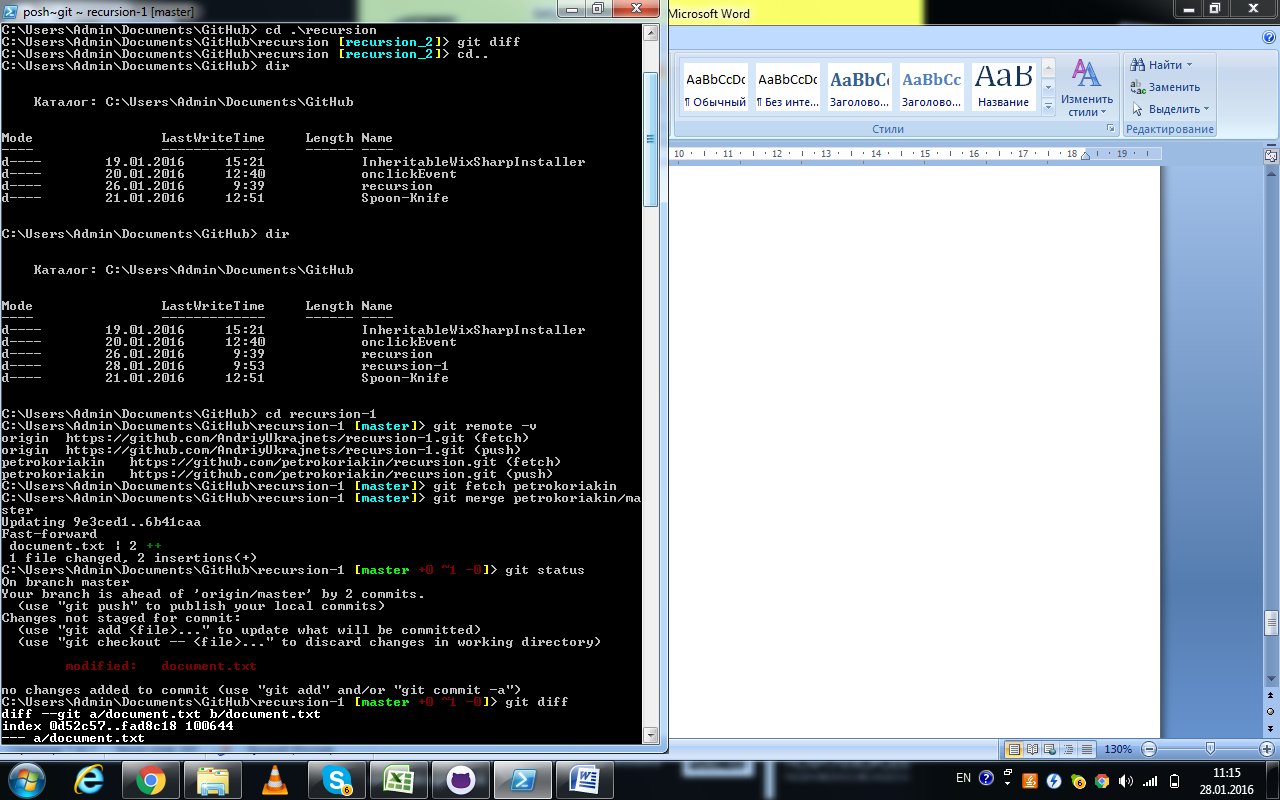
*> cd..*

1. Посмотреть, что есть в папке

> dir



1. Просмотр содержимого папки (предыдущая команда dir) покажет список папок и время последнего изменения.



1. **Синхронизация файлов**

Инфо к последующим шагам:

https://help.github.com/articles/configuring-a-remote-for-a-fork/

https://help.github.com/articles/syncing-a-fork/

1. Заходим в папку recursion-1

*> cd recursion-1*

1. Посмотреть, что происходило с данным удаленным репозиторием

*> git remote -v*

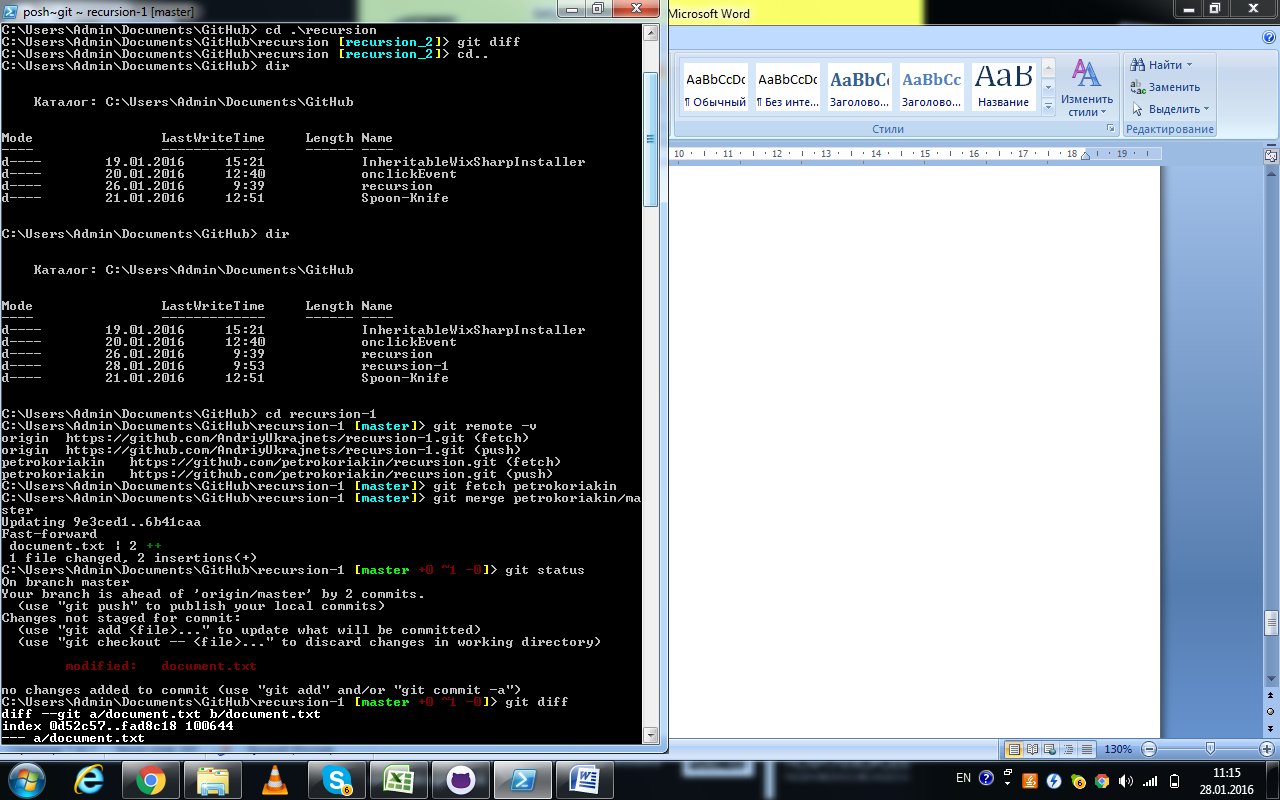
1. Ряд команд, которые позволят добиться синхронизации:

*> git fetch*

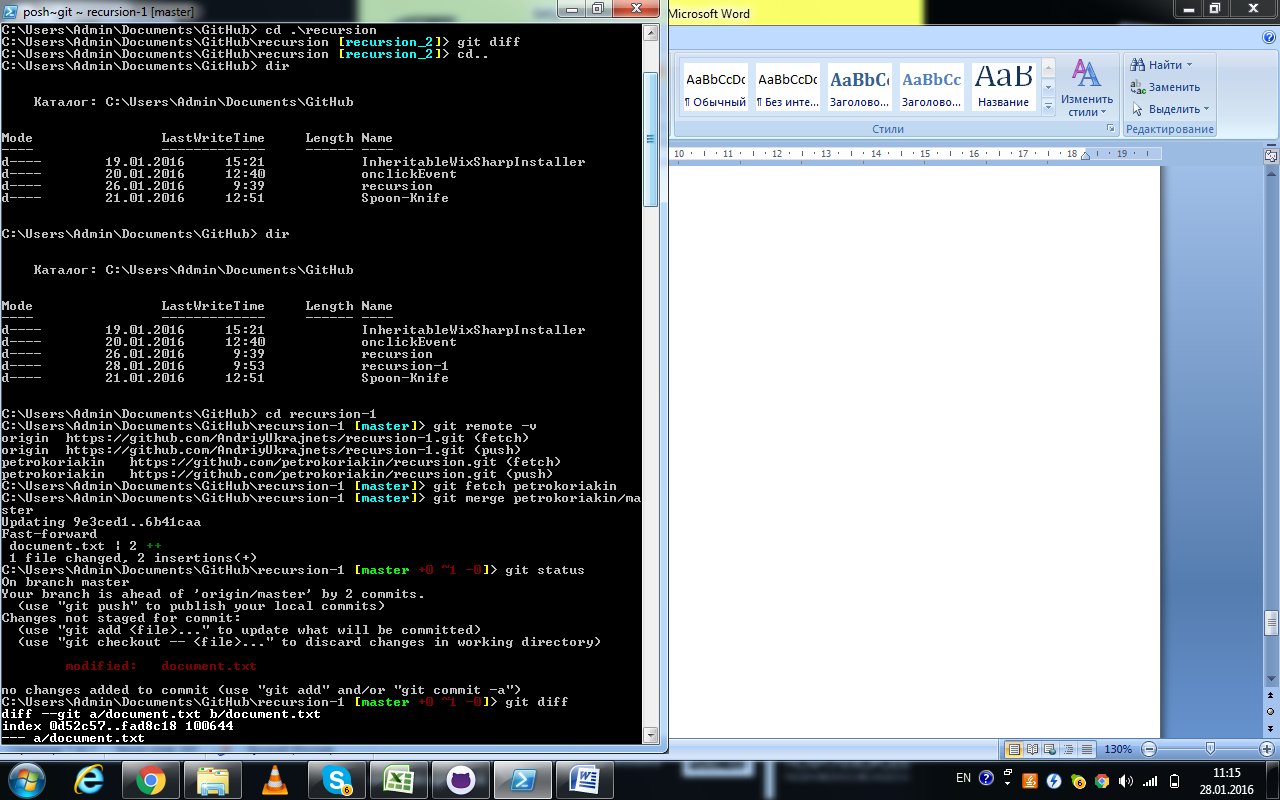
*> git merge*

1. Все ли хорошо получилось:

> git status

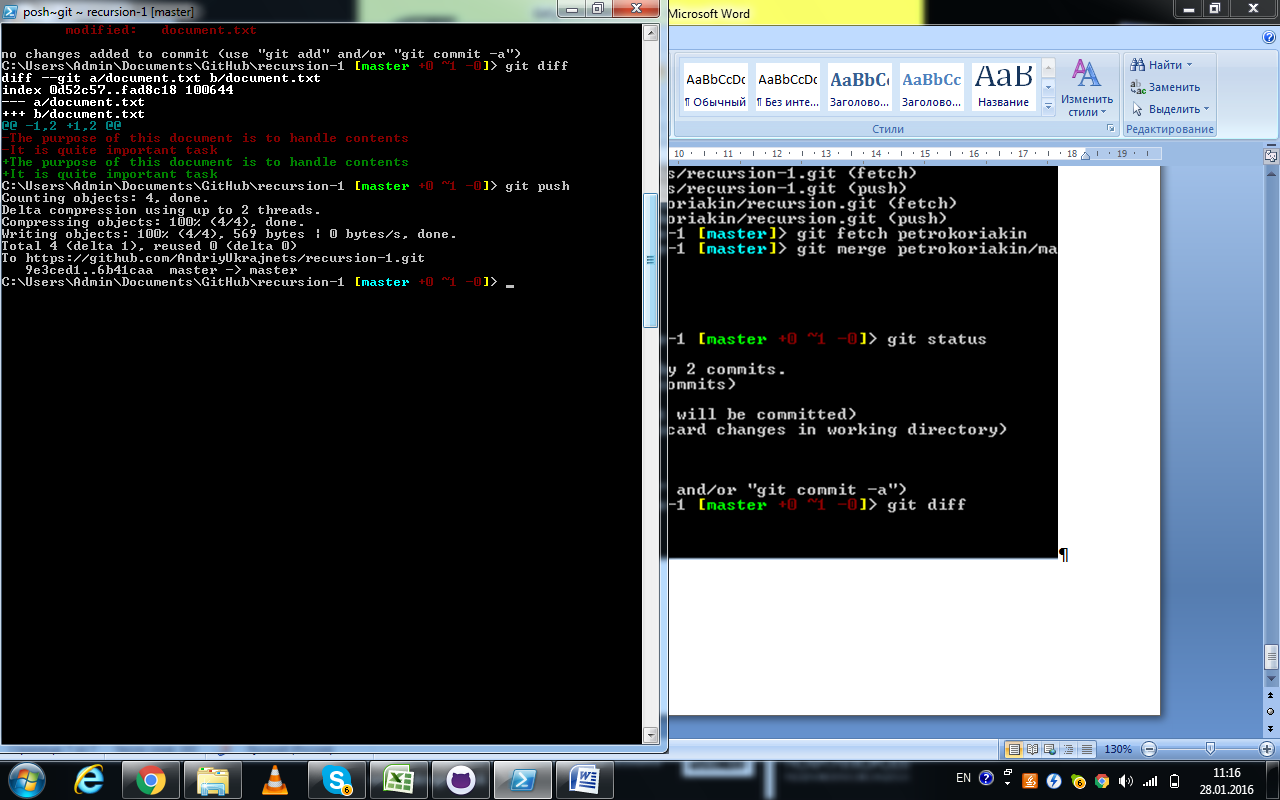


1. Изменения, которые были сделаны в документе будут выделены красным с пометкой modifiet

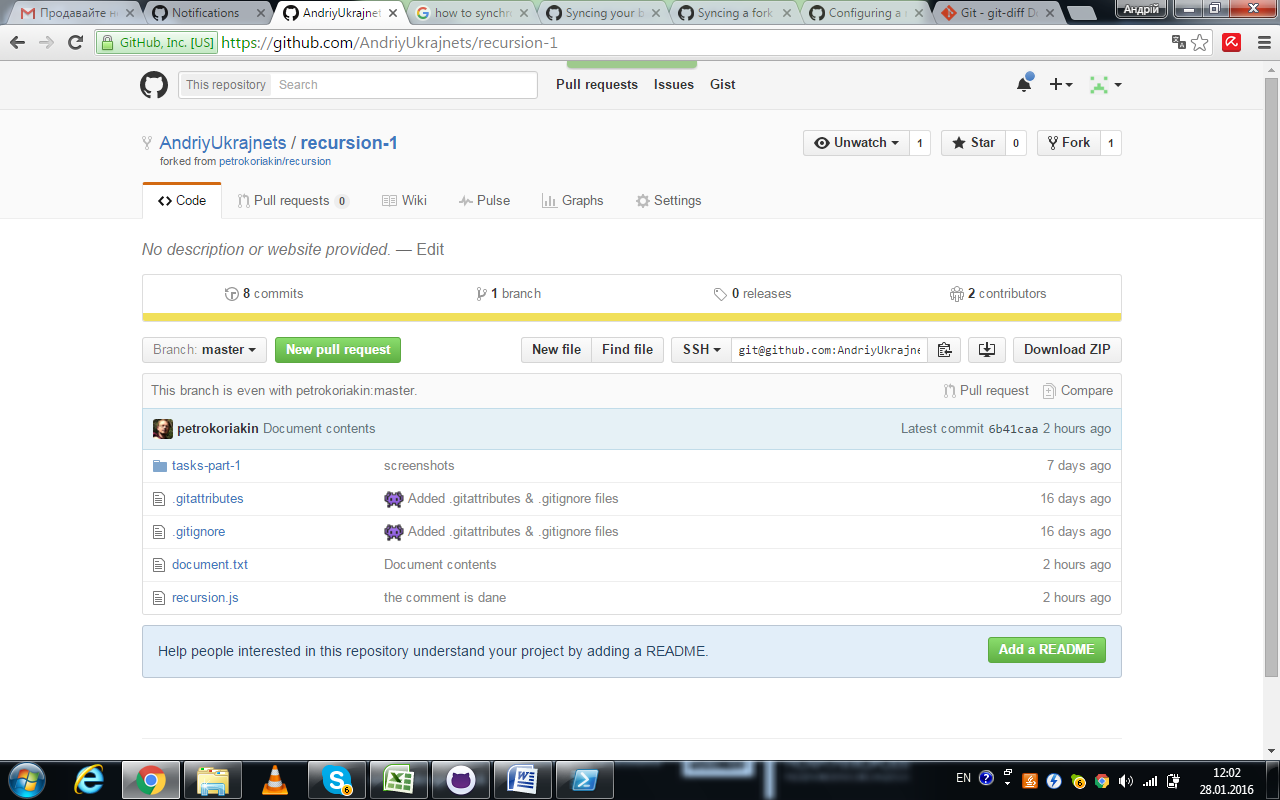


1. Перенесем то, что скопировали себе, из изменений, которые сделал petrokoriakin:

> git push.



1. **Проверка сделанных изменений на github.com.**
2. В репозитории своей страницы на github нахожу файл document.txt



1. В этом файле вижу тот комментарий, который был внесен PetroKoriakin и находится теперь в файле уже моего репозитория.

