Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

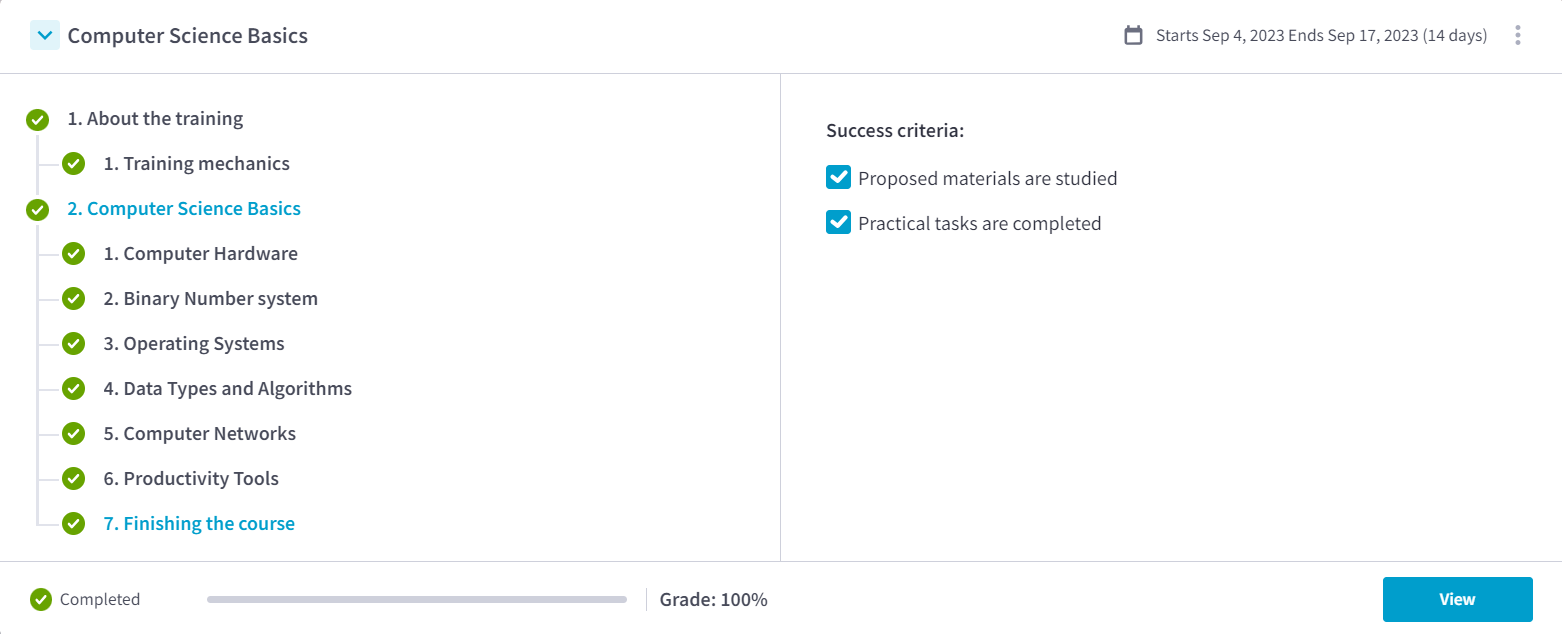
Тестирование программного обеспечения

Студент: Гарабажиу М.А.

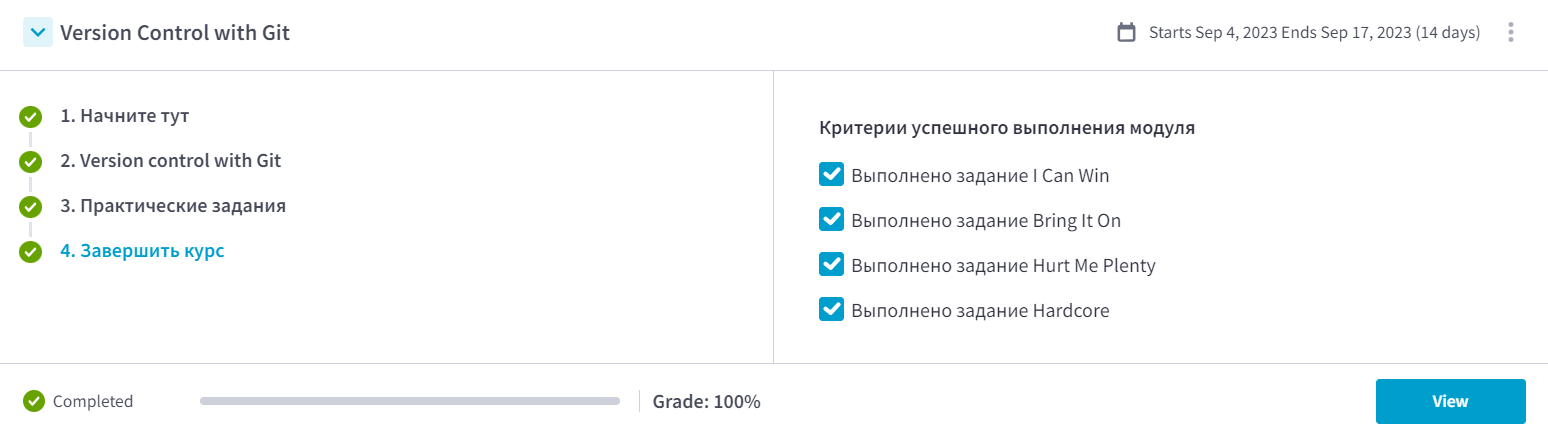
ФИТ 3 курс 8 группа

Преподаватель: Панченко О.Л.

Минск 2023



Модуль “Computer Science Basics”



Модуль “Version Control with Git”

1. I Can Win
2. Установите git и сгенерируйте пару ssh ключей. Авторизуйте публичный ключ на github.com.

ssh-keygen -t rsa -C "your\_email@example.com"  
//сгенерится дефолтный ключ и поместится в папку C:\Users\Personal Computer\.ssh   
//id\_rsa (приватный) и id\_rsa.pub (публичный) документы

//-C (comment) флаг комментария к ключу не обязателен

//-t (type) флаг типа ключа

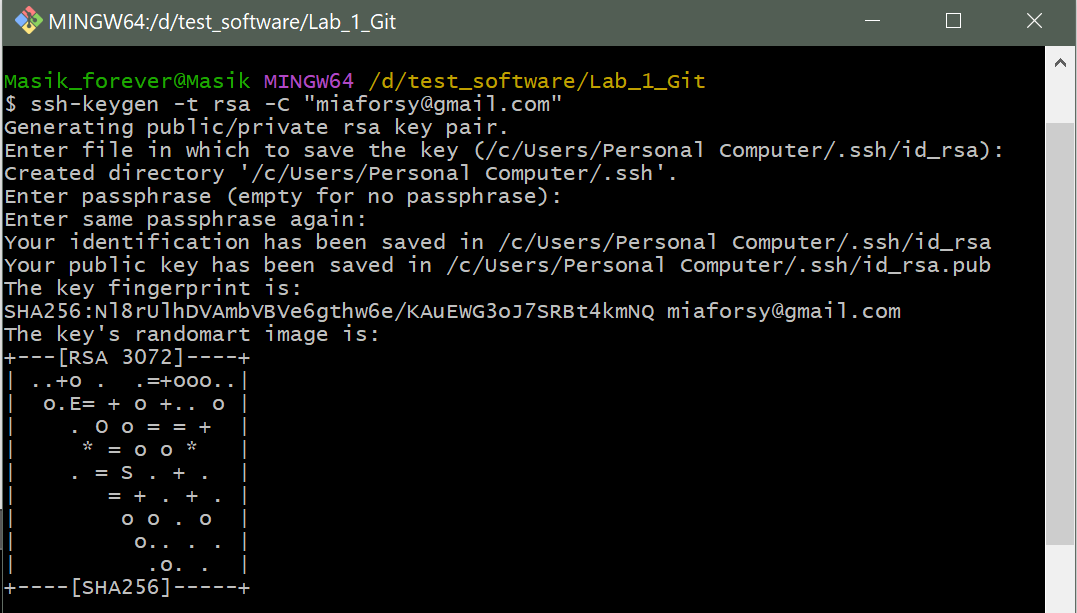
ssh-keygen -t rsa -C "your\_email@example.com" -f ~/.ssh/id\_rsa\_second

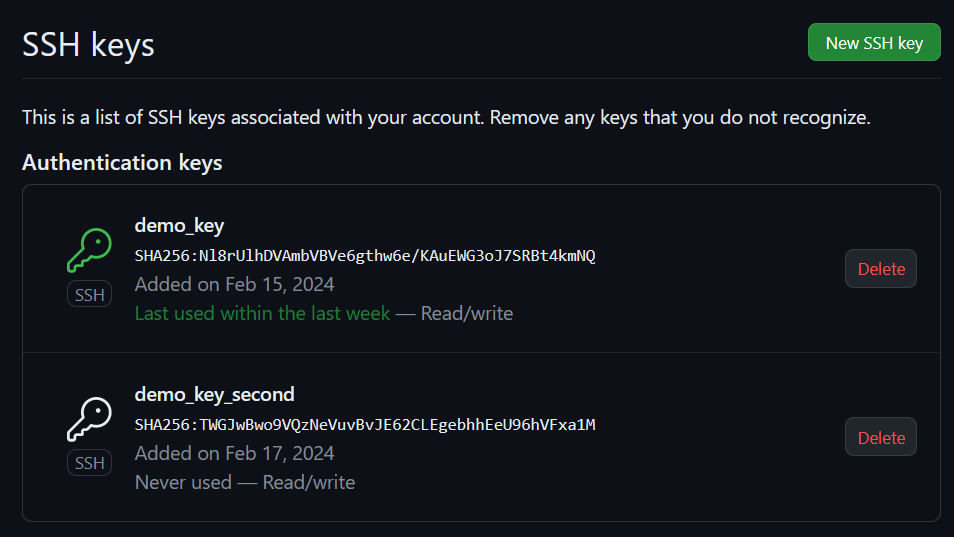
//флаг -f (file) указывает на имя файла ключа, чтобы сделать второй ключ в одной дефолтной папке

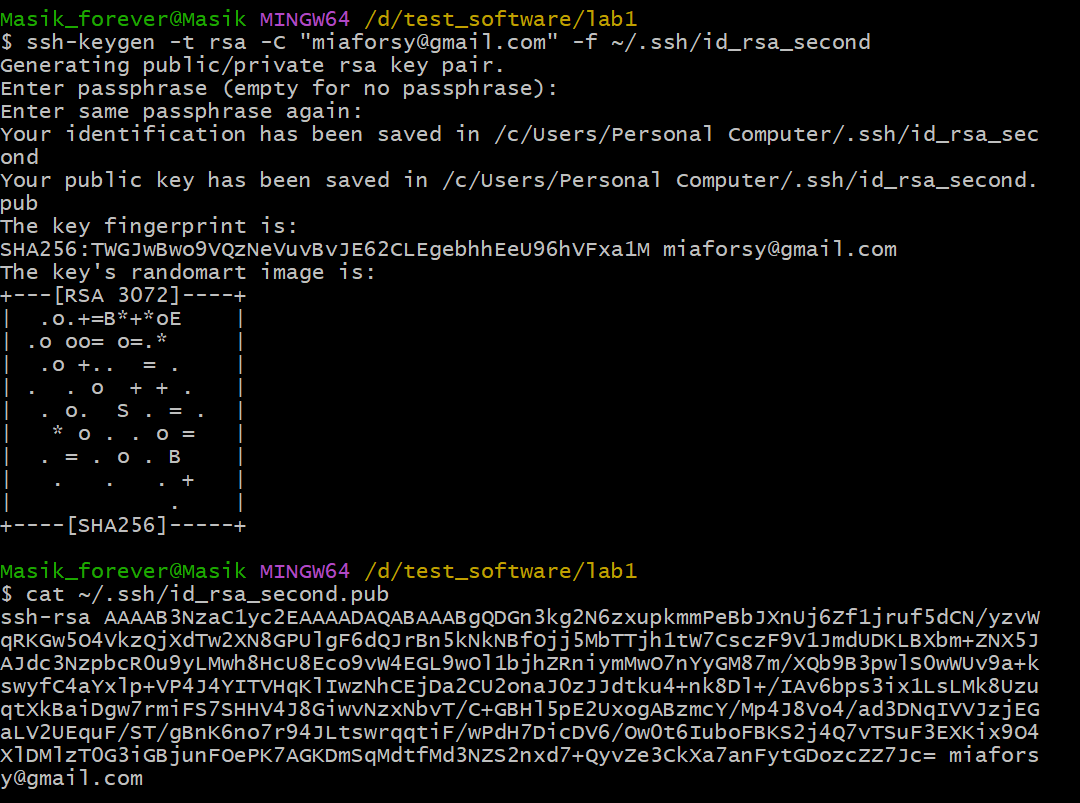
//~ домашний каталог пользователя

cat ~/.ssh/id\_rsa.pub

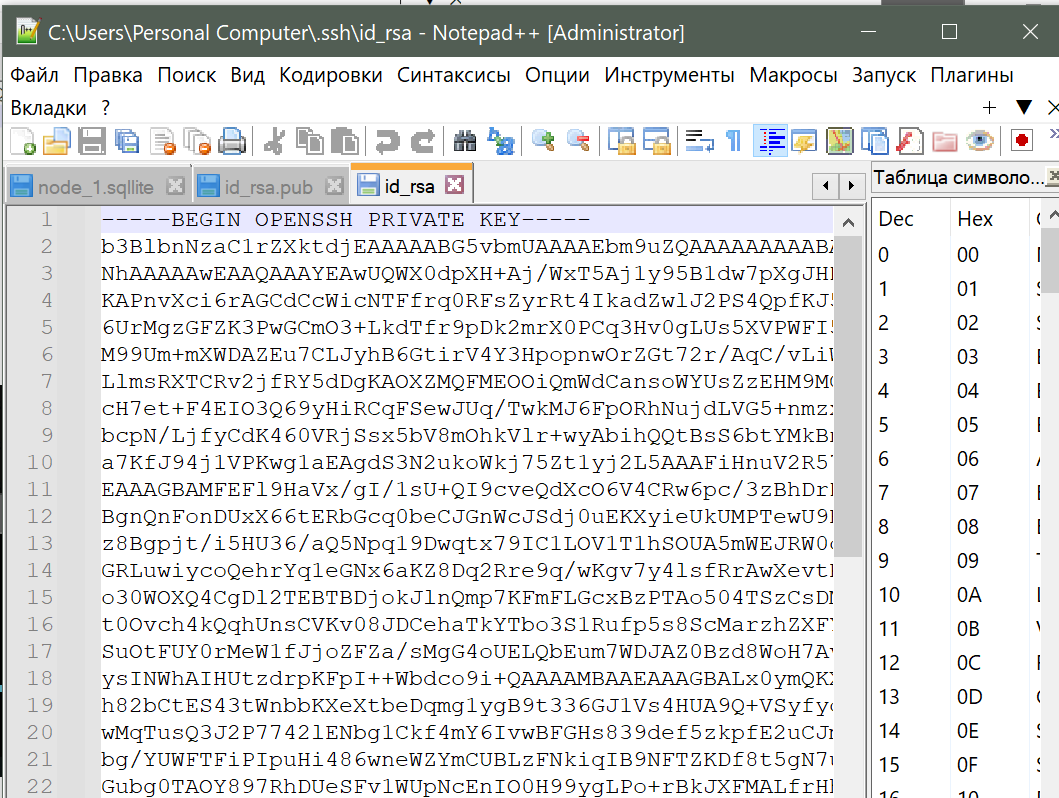
//содержимое файла прочесть



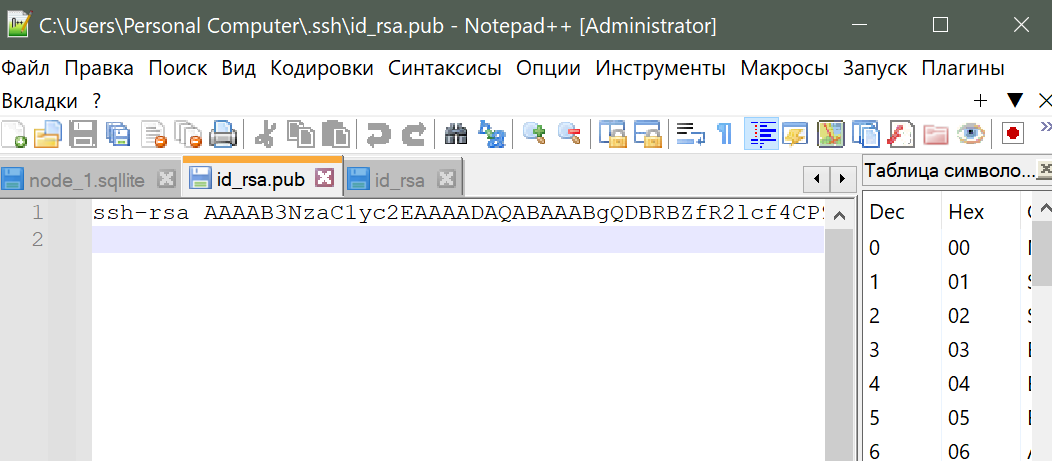




Private:



Public:



1. Укажите свой user.name и user.email в git.

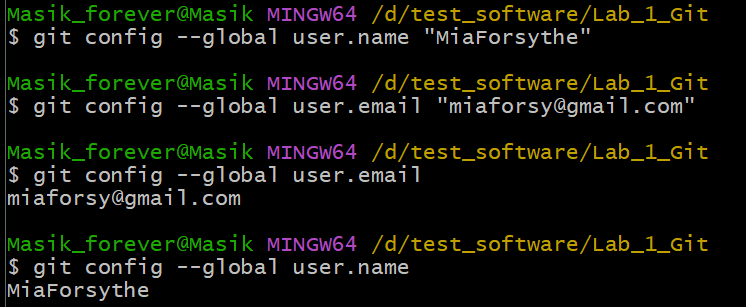
git config --global user.name "your\_name"  
//указать свое гит имя  
//config для настройки параметров гит  
//--global флаг глобальности (применимо ко всем репозиториям, а не только к текущему)

//-- дальше аргумент команды (идут после - часто)

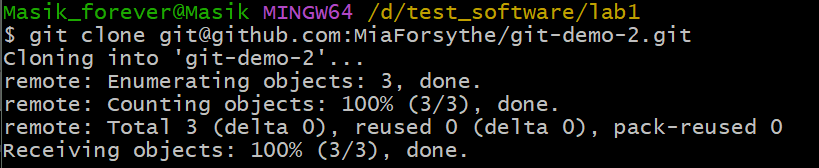
//- дальше параметр или флаг или опция

git config --global user.name

// выведет имя указанное ранее



1. Используя веб-интерфейс github’а создайте новый public репозиторий **git-demo** содержащий README файл (Initialize with a README) на github.com.
2. Склонируйте созданный репозиторий git-demo на свой компьютер.



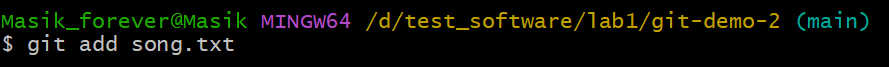
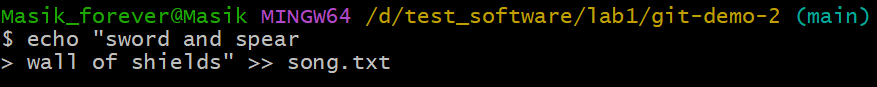
git clone git@github.com:MiaForsythe/git-demo-2.git

// клонировать репозиторий который создали вручную в интерфейсе гита  
// [git@github.com:MiaForsythe/git-demo-2.git](mailto:git@github.com:MiaForsythe/git-demo-2.git) это копирование на гите ссылки на созданный реп  
// README.md галочка на создание автоматически  
// при клонировании git init делается автоматически – инициализация гита репозитория - .git – скрытая папка где отслеживаются все изменения – без нее это будет обычная папка на компе но никак не репозиторий

1. Создайте текстовый файл названием song.txt и поместите туда фрагмент текста песни:

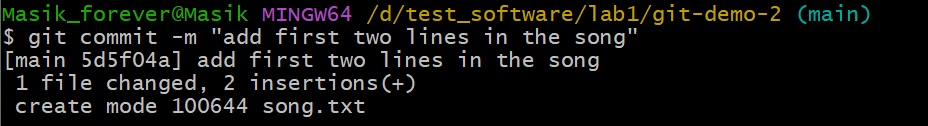
sword and spear  
wall of shields

echo "sword and spear  
wall of shields" >> song.txt  
//поместить содержимое в файл, даже если его не существует еще  
  
git add song.txt



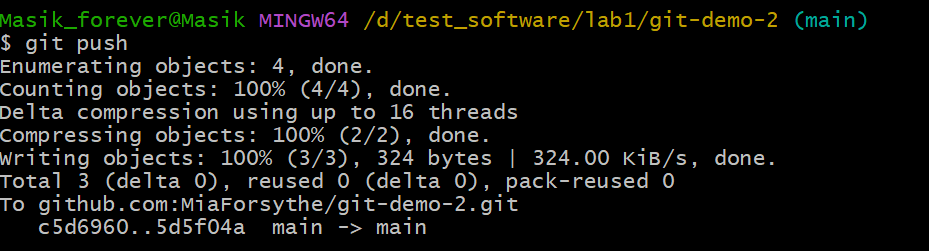
1. Сделайте коммит с названием "**add first two lines in the song**".

git commit -m "add first two lines in the song"  
//-m (message)



1. Отправьте локальные изменения на сервер.

git push  
// локальные изменения на удаленный сервер (с локального на удаленный репозиторий)  
//push в переводе нажать/давить/толкать

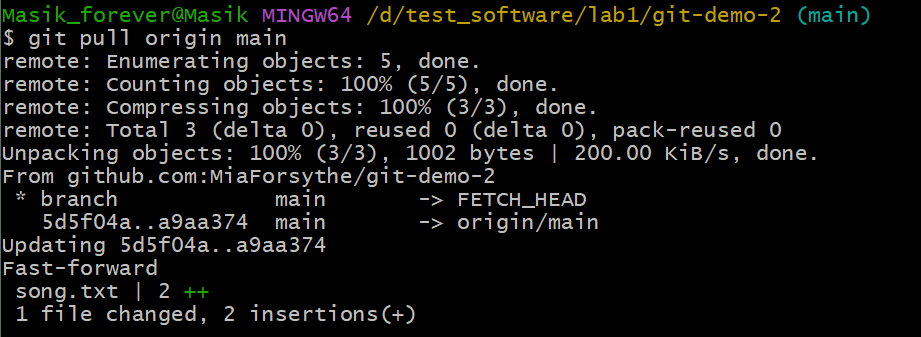


1. Убедитесь, что на github’е есть файл song.txt с текстом песни.
2. Используя веб-интерфейс github’а добавьте вторую половину текста песни в файл song.txt:

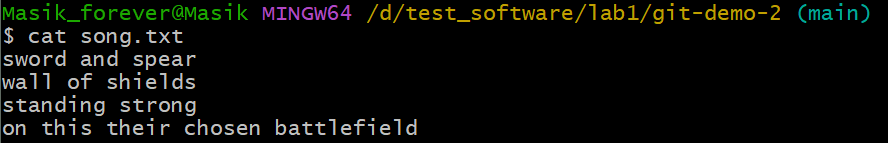
standing strong  
on this their chosen battlefield

1. Сделайте коммит с названием "**add two more lines to the song**".
2. Выполните синхронизацию локального репозитория с github’ом (“накатите” актуальную версию мастера с github’а выполнив команду pull в локальном репозитории).

git pull origin main  
// то что на гите те изменения внести в локальный реп  
//pull в переводе вытащить/вытянуть



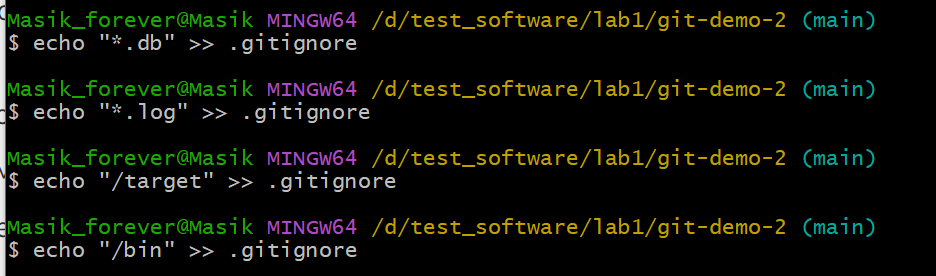
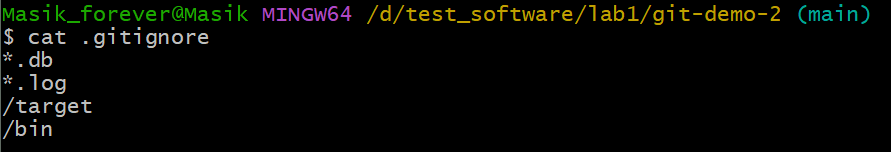
1. Убедитесь, что коммит, который вы создали на github’е, подтянулся и у вас полный текст песни в файле song.txt.



1. Bring It On
2. Добавьте в проект файл .gitignore и настройте так чтобы скрыть файлы с расширением **.db**, **.log** и директории с названиями **target** или **bin**.

//target папки хранят собранные/скомпилированные файлы проектов по типу gradle или java  
// bin папки хранят исполняемые файлы (бинарные)  
// можно вручную создать можно через команды:  
echo "\*.db" >> .gitignore

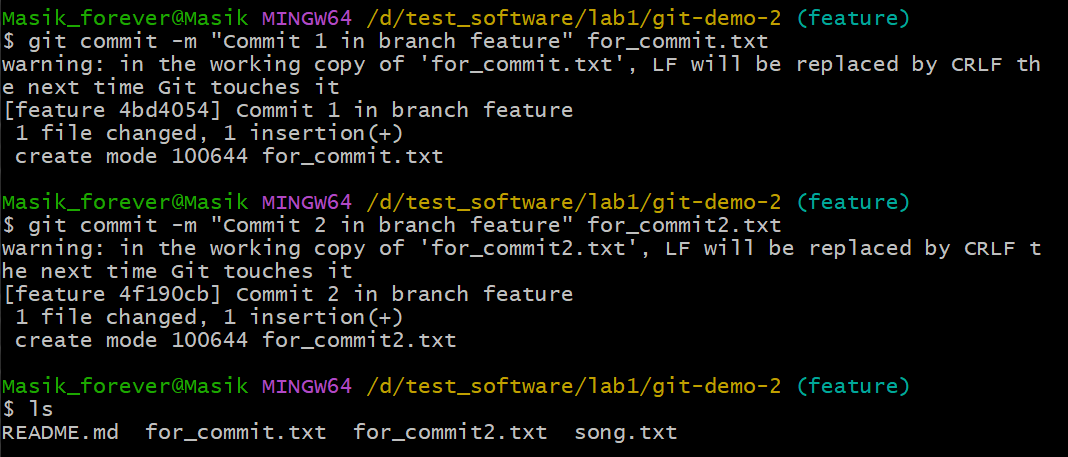
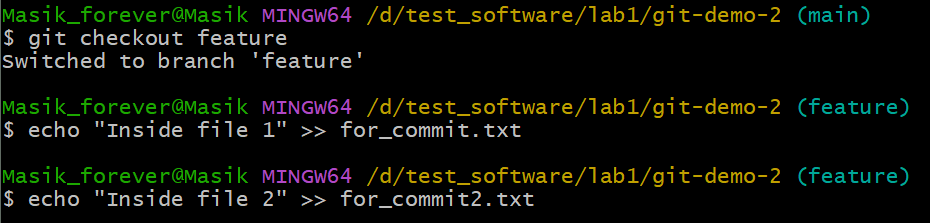
echo "/target" >> .gitignore

1. Создайте ветку feature и добавьте в неё два коммита

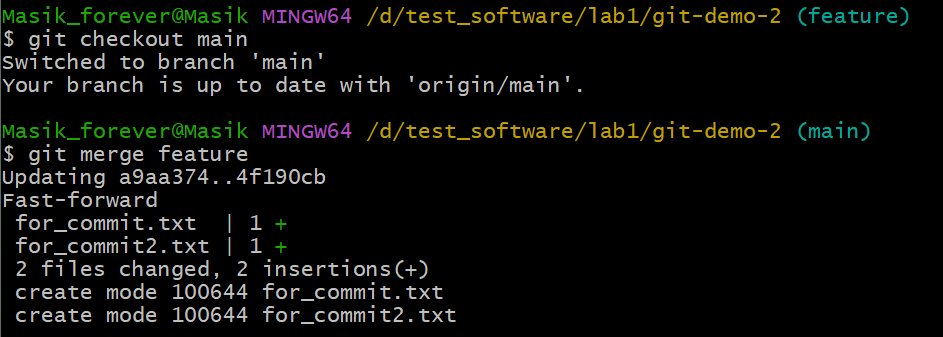
git branch feature   
git checkout feature

//branch ветка checkout переключать



1. Смержите ветку feature в master

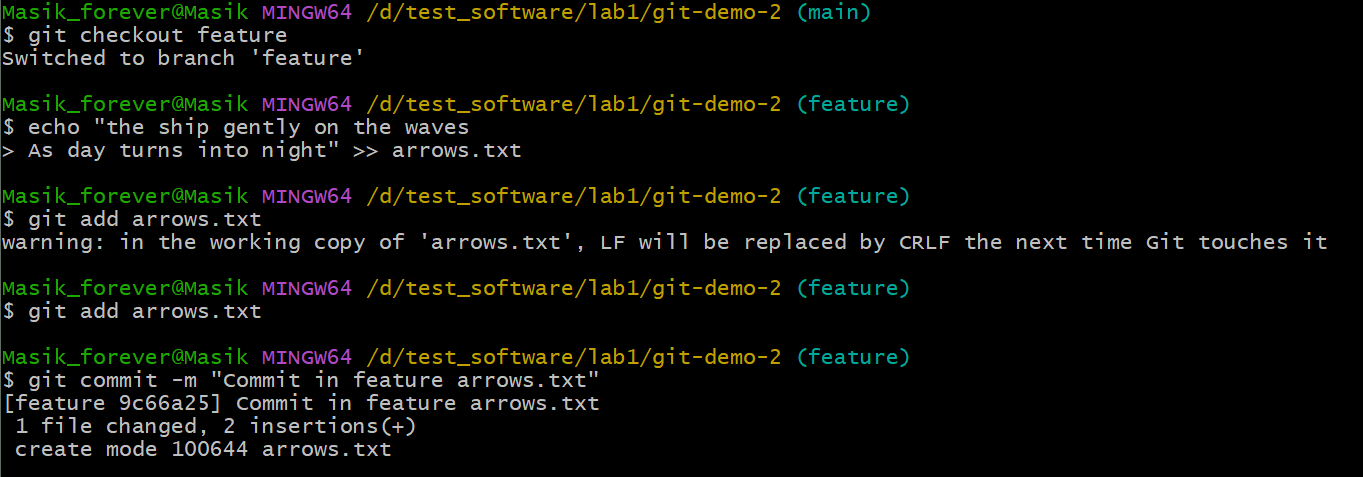
git checkout feature  
//слияние ветки feature в main



4. Вернитесь в feature и создайте файл arrows.txt cледующего содержания:

The ship glides gently on the waves  
As day turns into night

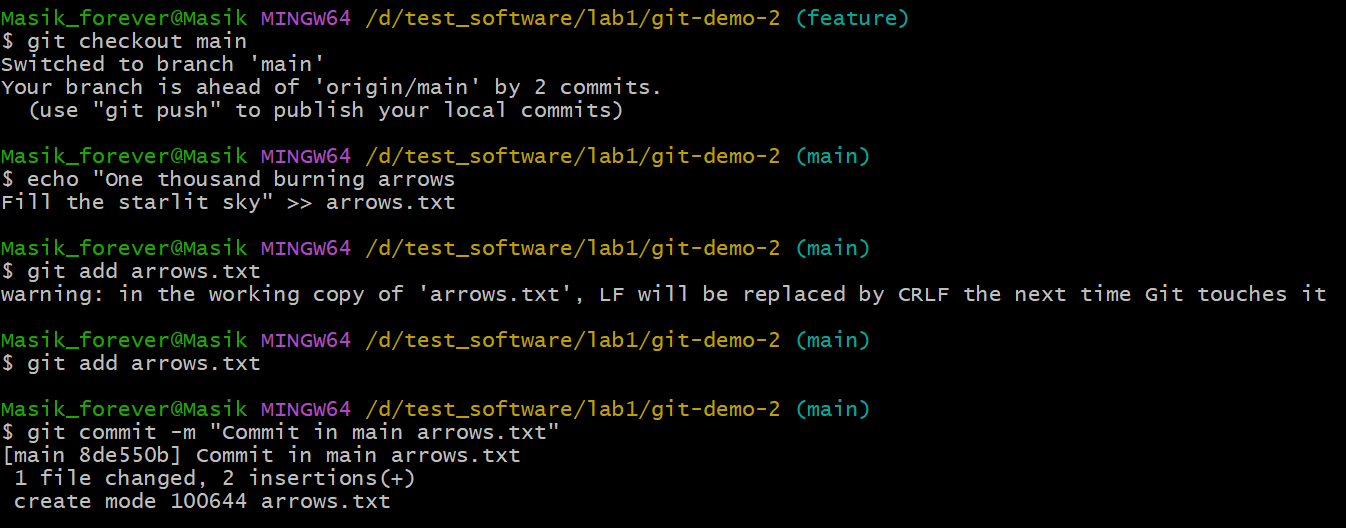
Выполните коммит.



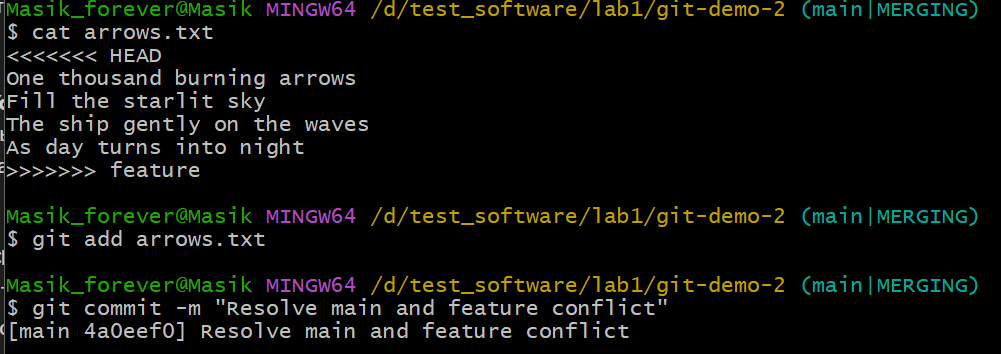
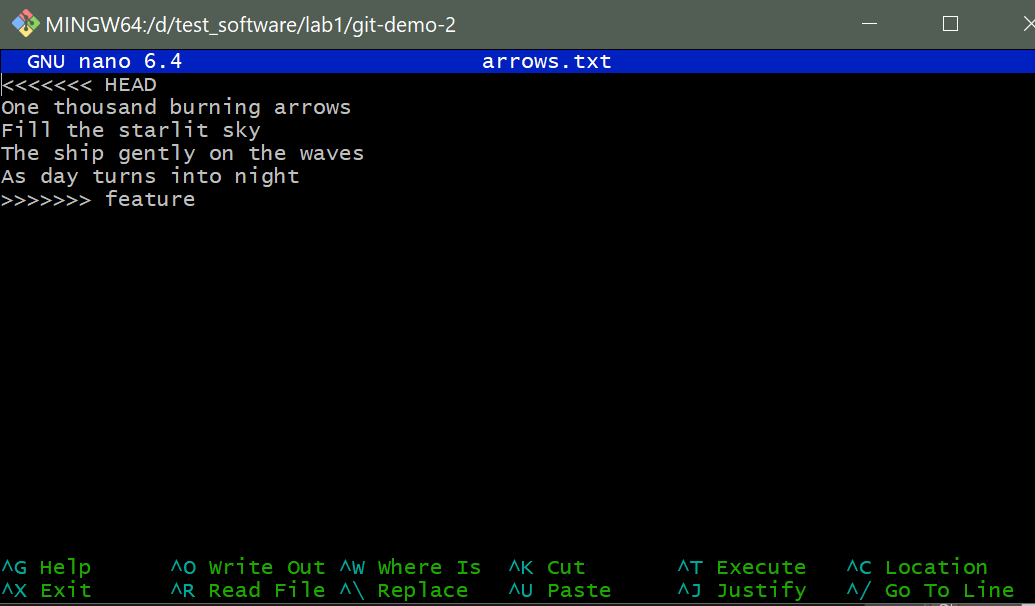
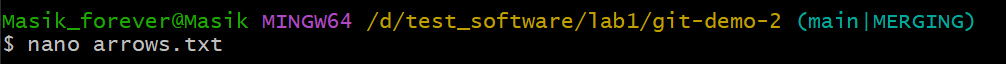
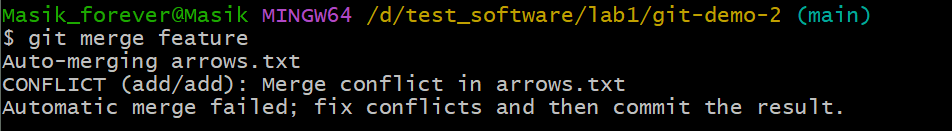
5. Перейдите в master. Создайте там файл arrows.txt и добавьте следующий текст:

One thousand burning arrows  
Fill the starlit sky

Выполните коммит.



6. Смержите feature в master решив конфликт: сохраните все 4 строки в файле arrows.txt в порядке их добавления в пунктах 4 и 5.

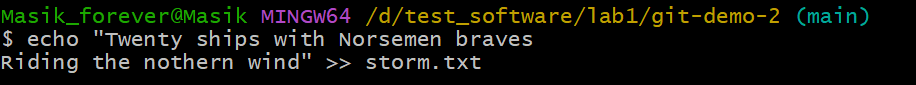
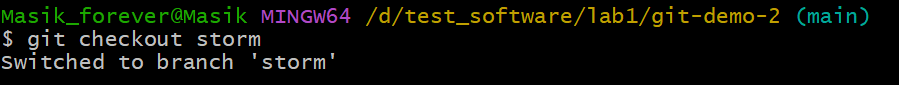
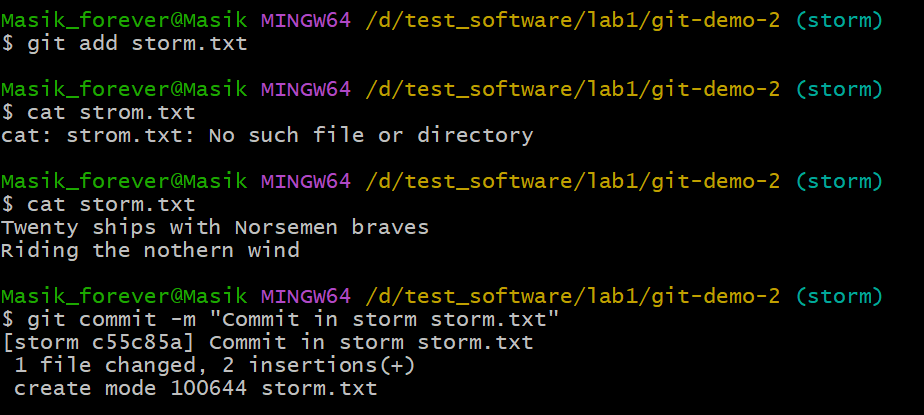


//nano текстовый редактор в командной строке

1. Hurt Me Plenty

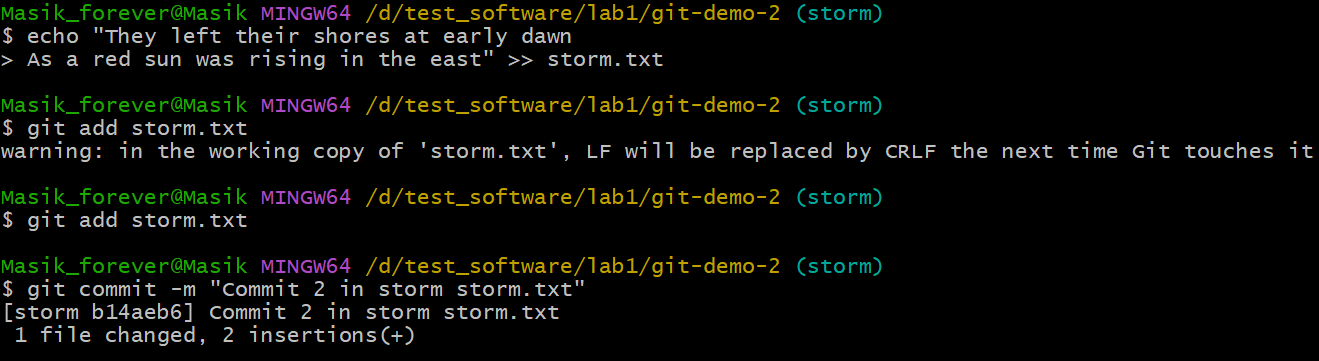
1. Создайте ветку storm и добавьте коммит в файл storm.txt:

Twenty ships with Norsemen braves  
Riding the northern wind

2. Добавьте еще 2 строки в storm.txt и сделайте еще один коммит:

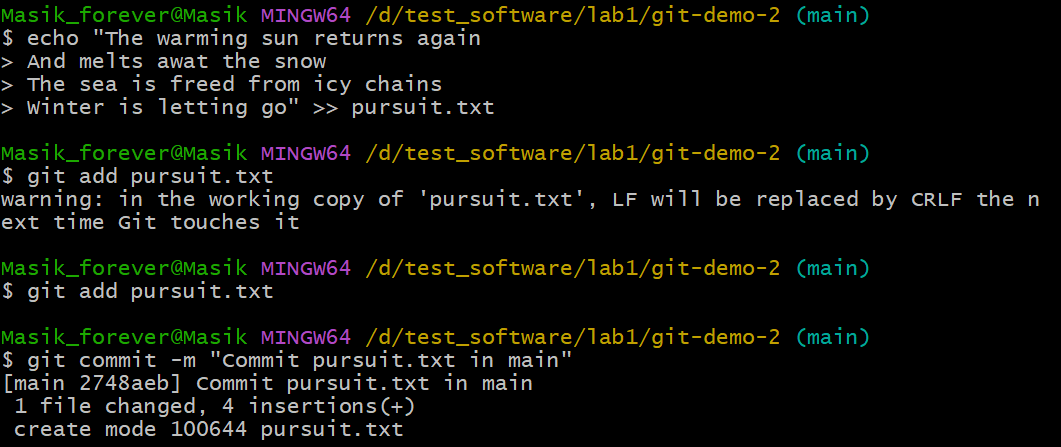
They left their shores at early dawn  
As a red sun was rising in the east



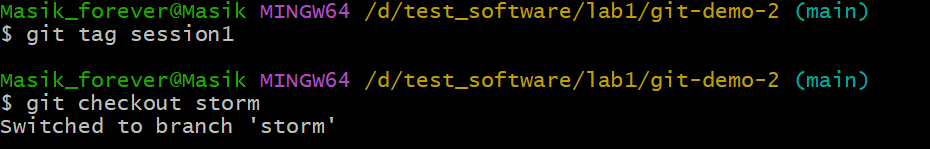
3. Вернитесь в master и создайте файл pursuit.txt с текстом ниже:

The warming sun returns again  
And melts away the snow  
The sea is freed from icy chains  
Winter is letting go

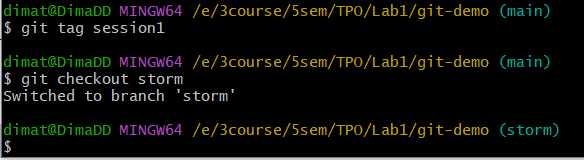
Выполните коммит.

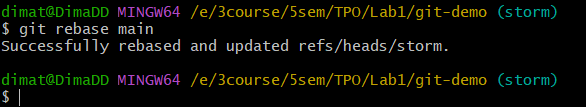


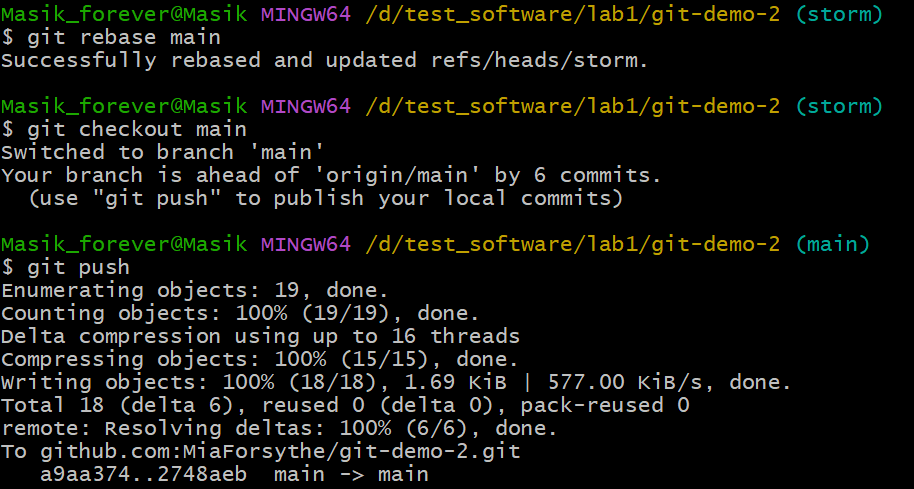
1. Отметьте коммит тегом session1 и перейдите в ветку storm

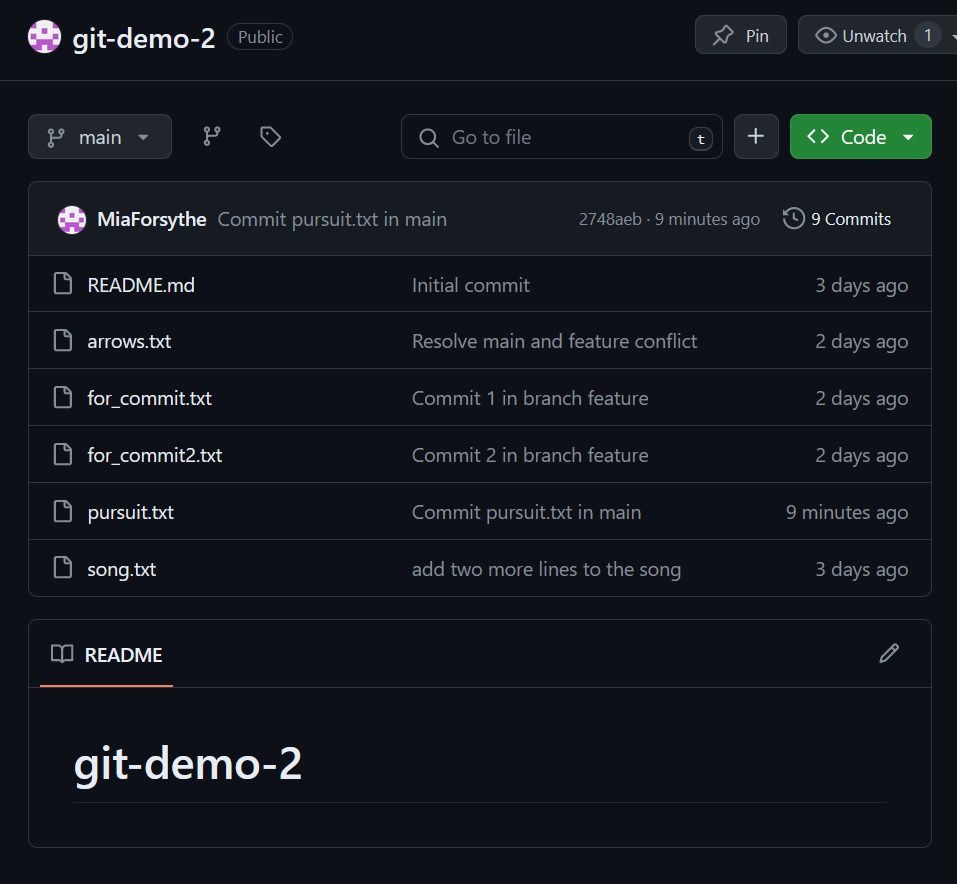


5. Сделайте rebase ветки storm так чтобы она содержала последний коммит из мастера.









1. Hardcore

1. Сделайте push вашего репозитория и убедитесь, что все коммиты есть на github.



git push origin main  
//слияние ветки feature в main

или git push origin main

2. Сделайте новый репозиторий на github.

3. Смените remote в локальном репозитории так, чтобы fetch и push шел на новый репозиторий который был создан в предыдущем шаге.

4. Сделайте push и убедитесь, что второй репозиторий на гитхабе выглядит так же, как и первый.

5. Верните настройки remote в исходное состояние: пул и пуш первого локального репозитория ведет в один удаленный репозиторий на гитхабе.

