Министерство Образования Республики Молдова

Технический Университет Молдовы

Кафедра Автоматики и Информационных Технологий

Лабораторная работа №2

По дисциплине: «MIDPS»

Тема: «Система контроля версий»

Выполнил: студент группы TI-145: Русу Дмитрий

Проверила: Кожану Ирина

Кишинёв 2016

Цель работы:

- Интегрирование и использование CLI (basic level)
- Удаленное администрирование машин Linux используя SSH (remote code editing)
- Version Control Systems (git || mercurial || svn)
- Скомпилировать код C/C++/Java/Python через среду CLI, используя компиляторы gcc/g++/javac/python

Теоретическая часть:

Here's how you start a project with GIT.

First, if you have not already done so, set yourself up to use GIT:

```
git config --global user.name "My Name"
git config --global user.email me@mydomain.com
git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
```

Now, export your project from whatever source code repository you used to use (you **were** already using a source code management system, right?):

NOTE! Use your source code management software's *export* utility to export your source code *without* your source code management software's special directories. Both CVS and Subversion litter your source code with CVS/ and .svn/ files respectively. You *don't* want to check those in to your fresh GIT repository!

```
cd ~/workspace
svn export https://www.mydomain.com:5001/myproject/trunk ./myproject
```

Now, import your project into GIT, like so:

```
cd myproject
git init
git add .
git commit
```

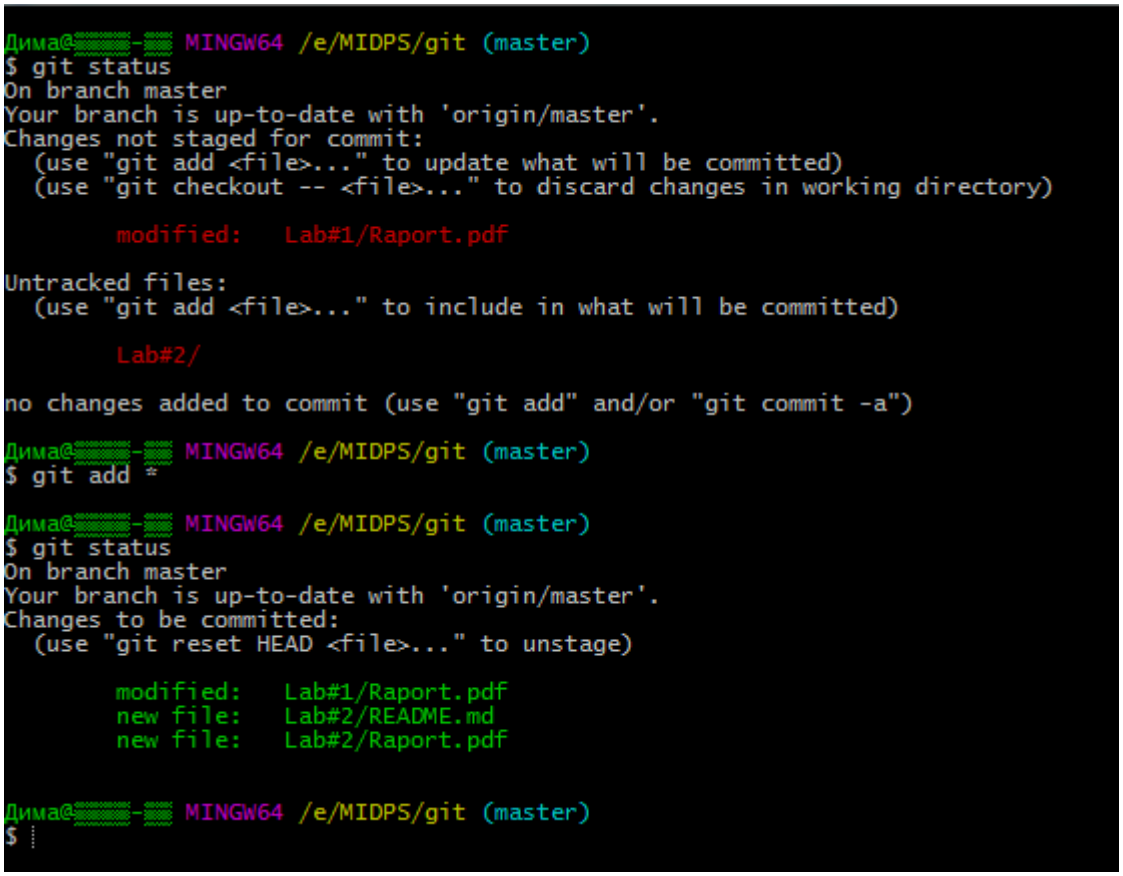
Of course, you want to share with your friends by setting up a bare remote repository. The easiest way is to have a shared machine where you and your friends all have ssh accounts. In all of your home directories, you have a directory called shared/ which is readable/writeable by everyone in your group. You all keep your shared bare git repositories in your shared/ directories on your remote machine, and life is good.

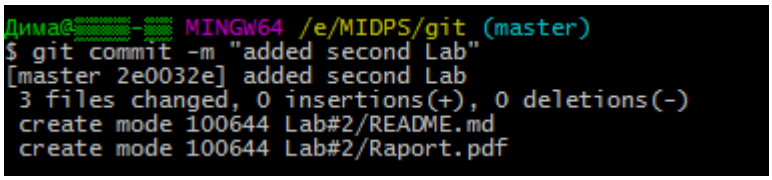
Continuing from where we left off, here's how you set up your bare repository:

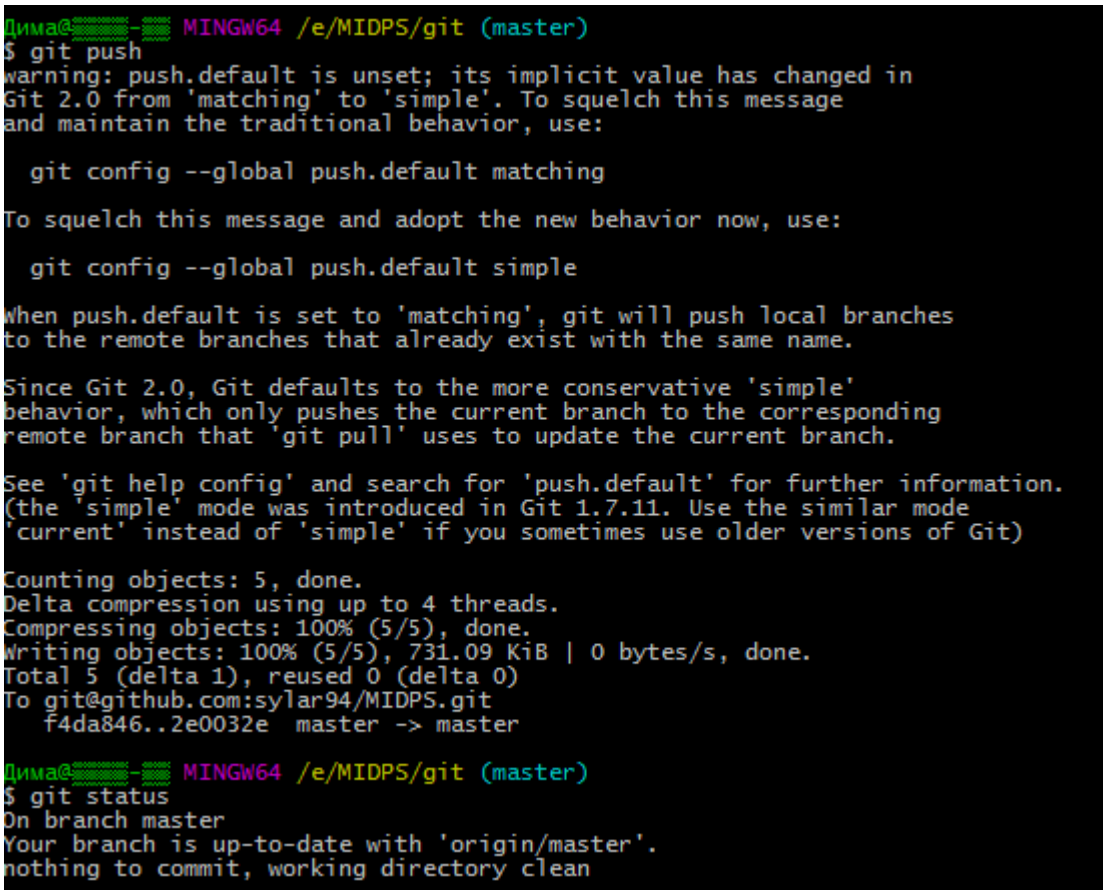
```
cd ..  # this is important; you must be outside your project dir
git clone --bare ./myproject myproject.git
scp -r myproject.git mydomain.com:shared
```

Now you can do nightly pushes to your bare git repository (can you say "easy backup"?) and your friends can pull your work.

Скриншоты работы в git bash:







Вывод:

В ходе данной лабораторной работы мы изучили основы работы в системе контроля версий. Произвели настройку git для работы на локальном компьютере при помощи git bash и создали репозиторий для последующих лабораторных работ. Создали вторую ветку и произвели пару commit-ов и push-ов.