

**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

## 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:

```
Дима@MINGW64 /e/MIDPS/git (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Lab#1/Raport.pdf

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Lab#2/

no changes added to commit (use "git add" and/or "git commit -a")

Дима@MINGW64 /e/MIDPS/git (master)
$ git add *
```

```
Дима@MINGW64 /e/MIDPS/git (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Lab#1/Raport.pdf
        new file:   Lab#2/README.md
        new file:   Lab#2/Raport.pdf

Дима@MINGW64 /e/MIDPS/git (master)
$ .....
```

```
Дима@MINGW64 /e/MIDPS/git (master)
$ git commit -m "added second Lab"
[master 2e0032e] added second Lab
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Lab#2/README.md
create mode 100644 Lab#2/Raport.pdf
```

```
Дима@MINGW64 /e/MIDPS/git (master)
$ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 731.09 KiB | 0 bytes/s, done.
Total 5 (delta 1), reused 0 (delta 0)
To git@github.com:sylar94/MIDPS.git
 f4da846..2e0032e master -> master

Дима@MINGW64 /e/MIDPS/git (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

## 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.