Отчет, лаба 6, ВРПО


```
$ git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy
-fetch]
           [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envva
r>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
   show      Show various types of objects
   status    Show the working tree status

grow, mark and tweak your common history
   backfill  Download missing objects in a partial clone
   branch    List, create, or delete branches
   commit    Record changes to the repository
   merge     Join two or more development histories together
   rebase    Reapply commits on top of another base tip
   reset     Reset current HEAD to the specified state
   switch    Switch branches
   tag       Create, list, delete or verify tags

collaborate (see also: git help workflows)
   fetch     Download objects and refs from another repository
   pull      Fetch from and integrate with another repository or a local branch
   push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

olgal@Danik MINGW64 ~
```

```
olgal@Danik MINGW64 ~
$ git config --global user.name "Danik"

olgal@Danik MINGW64 ~
$ git config --global user.email harzy228@gmail.com

olgal@Danik MINGW64 ~
$ config --list
bash: config: command not found
```

```
olgal@Danik MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Danik
user.email=harzy228@gmail.com
```

```
olgal@Danik MINGW64 ~
$ git init
Initialized empty Git repository in C:/Users/olgal/.git/

olgal@Danik MINGW64 ~ (master)
$
```

```
olgal@Danik MINGW64 /c
$ cd C:/user/-

olgal@Danik MINGW64 /c/user/-
$ git init
Initialized empty Git repository in C:/user/-/.git/

olgal@Danik MINGW64 /c/user/- (master)
$
```

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        -.sln
        -/
        .vs/
        Hello.txt.txt

nothing added to commit but untracked files present (use "git add" to track)

olgal@Danik MINGW64 /c/user/- (master)
$
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:    Hello.txt.txt

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


olgal@Danik MINGW64 /c/user/- (master)
$ |
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   Hello.txt.txt
#
# Untracked files:
#       -.sln
#       -/
#       .vs/
#
~
~
~
~
```

```
$ git add .gitignore

olgal@Danik MINGW64 /c/user/- (master)
$ git commit -m "Добавлен .gitignore"
[master (root-commit) 67329b4] Добавлен .gitignore
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
 create mode 100644 Hello.txt.txt

olgal@Danik MINGW64 /c/user/- (master)
$ 
```

```
olgal@Danik MINGW64 /c/user/- (master)
$ git branch test

olgal@Danik MINGW64 /c/user/- (master)
$ git checkout test
Switched to branch 'test'

olgal@Danik MINGW64 /c/user/- (test)
$ 
```

```
olgal@Danik MINGW64 /c/user/- (test)
$ touch test.html

olgal@Danik MINGW64 /c/user/- (test)
$ git add test.html

olgal@Danik MINGW64 /c/user/- (test)
$ git commit -m "Добавлен файл test.html"
[test 498ab70] Добавлен файл test.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.html

olgal@Danik MINGW64 /c/user/- (test)
$
```

```
olgal@Danik MINGW64 /c/user/- (test)
$ git branch
  master
* test
```

Находимся на ветке test

```
olgal@Danik MINGW64 /c/user/- (test)
$ git branch
  master
* test

olgal@Danik MINGW64 /c/user/- (test)
$ git checkout master
Switched to branch 'master'

olgal@Danik MINGW64 /c/user/- (master)
$ git merge test
Updating 67329b4..498ab70
Fast-forward
 test.html | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.html
```

```
olgal@Danik MINGW64 /c/user/- (master)
$ git checkout master
Already on 'master'

olgal@Danik MINGW64 /c/user/- (master)
$ git add test.html

olgal@Danik MINGW64 /c/user/- (master)
$ git checkout test
Switched to branch 'test'

olgal@Danik MINGW64 /c/user/- (test)
$ git add test.html

olgal@Danik MINGW64 /c/user/- (test)
$ git commit -m "Изменение в файле test.html в ветке test
> "
On branch test
Untracked files:
```
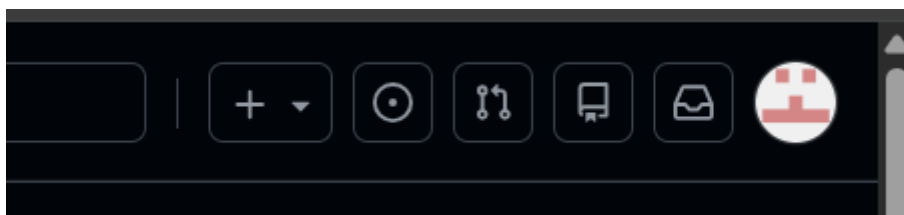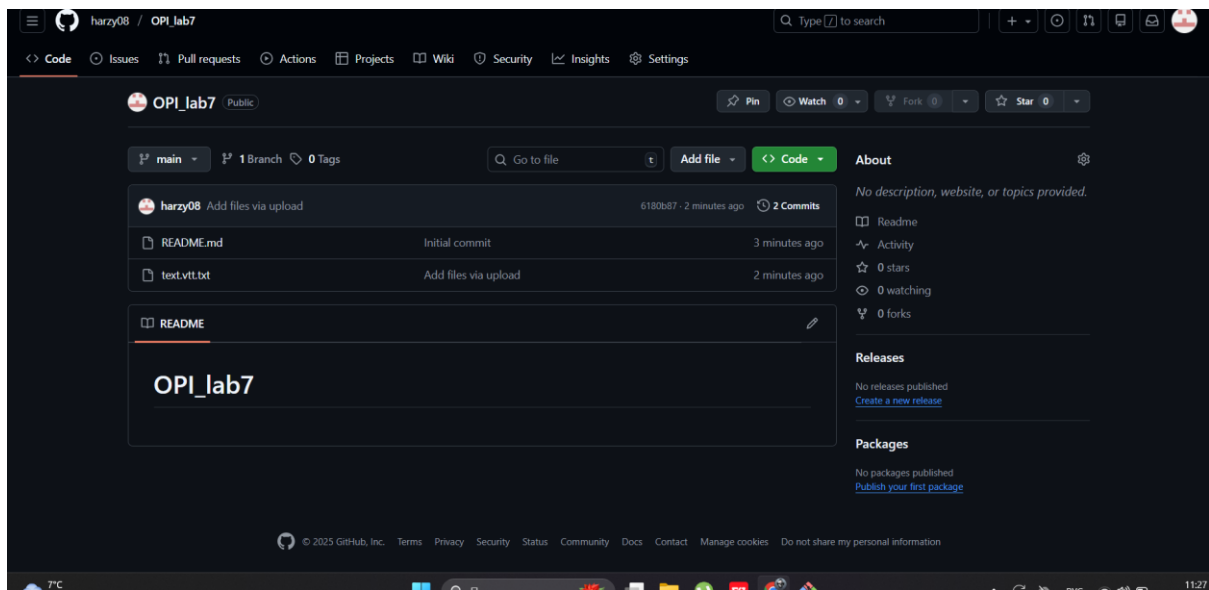


```
olgal@Danik MINGW64 /c/user/- (test)
$ git checkout master
Switched to branch 'master'

olgal@Danik MINGW64 /c/user/- (master)
$ git merge test
Already up to date.

olgal@Danik MINGW64 /c/user/- (master)
$
```

Отчет, лаба 7, ВРПО

```
olgal@Danik MINGW64 /c/Лазаренко
$ git clone https://github.com/harzy08/OPI_lab7
Cloning into 'OPI_lab7'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.

olgal@Danik MINGW64 /c/Лазаренко
$
```

```
olgal@Danik MINGW64 /c/Лазаренко
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Danik
user.email=harzy228@gmail.com

olgal@Danik MINGW64 /c/Лазаренко
$
```

```
olgal@Danik MINGW64 /c/Лазаренко
$ cd C:/user

olgal@Danik MINGW64 /c/user
$ cd C:/user/-

olgal@Danik MINGW64 /c/user/- (master)
$ nano README.md

olgal@Danik MINGW64 /c/user/- (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it

olgal@Danik MINGW64 /c/user/- (master)
$
```

```
olgal@Danik MINGW64 /c/user/- (master)
$ git remote -v
origin   https:/github.com/harzy08/OPI_lab7 (fetch)
origin   https:/github.com/harzy08/OPI_lab7 (push)

olgal@Danik MINGW64 /c/user/- (master)
$
```

```
olgal@Danik MINGW64 /c/user/- (master)
$ git push origin master
ssh: Could not resolve hostname https: Name or service not known
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

olgal@Danik MINGW64 /c/user/- (master)
$
```