

Лабораторная 6

№5

```
MINGW64:/
andre@HOYT MINGW64 /
$ 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 a tag object signed with GPG

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.

andre@HOYT MINGW64 /
$
```

№6

```
andre@HOYT MINGW64 /
$ git init
Reinitialized existing Git repository in C:/Users/andre/Downloads/Git/.git/

andre@HOYT MINGW64 /
$ |
```

№7

```
andre@HOYT MINGW64 /  
$ git config --global user.name Andrey  
  
andre@HOYT MINGW64 /  
$ git config --global user.email andrey_ploxo@mail.ru
```

№8

```
andre@HOYT 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=openssl  
http.sslcainfo=C:/Users/andre/Downloads/Git/mingw64/etc/ssl/certs/ca-bundle.crt  
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=Andrey  
user.email=andrey_ploxo@mail.ru
```

№9

Инициализация репозитория была выполнена на шаге 6. После инициализации в папке появился скрытый каталог `.git`, который содержит все необходимые файлы для работы с Git

#10

```
andre@HOYT MINGW64 /  
$ touch Hello.txt
```

#11

```

andre@HOYT MINGW64 / (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    LICENSE.txt
    ReleaseNotes.html
    bin/
    cmd/
    dev/
    etc/
    git-bash.exe
    git-cmd.exe
    lol1/
    mingw64/
    unins000.dat
    unins000.exe
    unins000.msg
    usr/

nothing added to commit but untracked files present (use "git add" to track)
andre@HOYT MINGW64 / (master)
$

```

#12

```

andre@HOYT MINGW64 / (master)
$ git add Hello.txt

```

№13

```

andre@HOYT MINGW64 / (master)
$ git commit -m "добавлен файл"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    LICENSE.txt
    ReleaseNotes.html
    bin/
    cmd/
    dev/
    etc/
    git-bash.exe
    git-cmd.exe
    lol1/
    mingw64/
    unins000.dat
    unins000.exe
    unins000.msg
    usr/

nothing added to commit but untracked files present (use "git add" to track)
andre@HOYT MINGW64 / (master)

```

№14

```
commit e545d35d5f0c8253186e6dd3a417dc66aa45e0a2 (HEAD -> master, qoq)
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:35:52 2024 +0300

    1qoq

commit e9f020d712b4006dd43a2f46faa10c4f289e0d06
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:34:09 2024 +0300

    1pop

commit d59a72595f6e1198dbd04f3ab6b6706d39d38804
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:27:38 2024 +0300

    1pop

commit c2ca32ca126ccf8ec1c65da94a7b4ce3dee062cd
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:08:52 2024 +0300

    pop1
~
~
~
~
~
~
~
~
~
(END)...skipping...
commit e545d35d5f0c8253186e6dd3a417dc66aa45e0a2 (HEAD -> master, qoq)
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:35:52 2024 +0300

    1qoq

commit e9f020d712b4006dd43a2f46faa10c4f289e0d06
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:34:09 2024 +0300

    1pop

commit d59a72595f6e1198dbd04f3ab6b6706d39d38804
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:27:38 2024 +0300

    1pop

commit c2ca32ca126ccf8ec1c65da94a7b4ce3dee062cd
Author: andrey <andrey_ploxo@mail.ru>
Date: Fri Dec 6 15:08:52 2024 +0300
```

№15

```
andre@HOYT MINGW64 / (master)
$ touch newfile.txt

andre@HOYT MINGW64 / (master)
$ git add Hello.txt newfile.txt

andre@HOYT MINGW64 / (master)
$
```

№16

git add <file> — индексация файла

git commit -m "сообщение" — фиксация изменений.

git status — проверка состояния.

git log — просмотр истории коммитов.

№17

```
andre@HOYT MINGW64 /lol1
$ git init
Initialized empty Git repository in C:/Users/andre/Downloads/Git/lol1/.git/

andre@HOYT MINGW64 /lol1 (master)
$ touch .gitignore

andre@HOYT MINGW64 /lol1 (master)
$
```

№18

```

andre@HOYT MINGW64 / (master)
$ git branch test

andre@HOYT MINGW64 / (master)
$ git checkout branch
error: pathspec 'branch' did not match any file(s) known to git

andre@HOYT MINGW64 / (master)
$ git checkout test
A       newfile.txt
Switched to branch 'test'

andre@HOYT MINGW64 / (test)
$

```

№19

```

andre@HOYT MINGW64 / (test)
$ touch test.html

andre@HOYT MINGW64 / (test)
$ git add test.html

andre@HOYT MINGW64 / (test)
$ git commit -m "добавлен файл html"
[test 99ce17c] добавлен файл html
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newfile.txt

andre@HOYT MINGW64 / (test)
$

```

№20

```

create mode 100644 newfile.txt

andre@HOYT MINGW64 / (test)
$ git branch
master
qoq
* test

andre@HOYT MINGW64 / (test)
$

```

```

andre@HOYT MINGW64 / (master)
$ git init my-repo
Initialized empty Git repository in C:/Users/andre/Downloads/Git/my-repo/.git/

andre@HOYT MINGW64 / (master)
$ cd my-repo

andre@HOYT MINGW64 /my-repo (master)
$ echo "BCEM ПРИВЕТ" > file.txt

andre@HOYT MINGW64 /my-repo (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it

andre@HOYT MINGW64 /my-repo (master)
$ git commit -m "добавлен файл"
[master (root-commit) 40afa4f] добавлен файл
1 file changed, 1 insertion(+)
create mode 100644 file.txt

andre@HOYT MINGW64 /my-repo (master)
$ git checkout -b branch1
Switched to a new branch 'branch1'

andre@HOYT MINGW64 /my-repo (branch1)
$ echo "изменение в branch1" > file.txt

andre@HOYT MINGW64 /my-repo (branch1)
$ git commit -am "измен в branch1"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it
[branch1 15010b9] измен в branch1
1 file changed, 1 insertion(+), 1 deletion(-)

andre@HOYT MINGW64 /my-repo (branch1)
$ git ckeckout master
git: 'ckeckout' is not a git command. See 'git --help'.

The most similar command is
checkout

andre@HOYT MINGW64 /my-repo (branch1)
$ git checkout master
Switched to branch 'master'

andre@HOYT MINGW64 /my-repo (master)
$ echo "gdbdfgdr" > file.txt

andre@HOYT MINGW64 /my-repo (master)
$ git commit -am "изменения в master"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it
[master 4b0ad11] изменения в master
1 file changed, 1 insertion(+), 1 deletion(-)

andre@HOYT MINGW64 /my-repo (master)
$ git merge branch1
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt

```

```

$ echo gdbdfgdr > file.txt

andre@HOYT MINGW64 /my-repo (master)
$ git commit -am "изменения в master"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it
[master 4b0ad11] изменения в master
1 file changed, 1 insertion(+), 1 deletion(-)

andre@HOYT MINGW64 /my-repo (master)
$ git merge branch1
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.

andre@HOYT MINGW64 /my-repo (master|MERGING)
$ git add file.txt

andre@HOYT MINGW64 /my-repo (master|MERGING)
$ git commit -m "разрешен конфликт"
[master 513b4c3] разрешен конфликт

andre@HOYT MINGW64 /my-repo (master)
$ git merge branch1
Already up to date.

andre@HOYT MINGW64 /my-repo (master)
$

```

hello.txt

Файл

Изменить

Просмотр

```

|<<<<<< HEAD
|gdbdfgdr
|=====
|изменение в branch1
|>>>>>> branch1

```

```

<<<<<< HEAD
gdbdfgdr
изменение в branch1
>>>>>> branch1

```