Software Engineering 2:

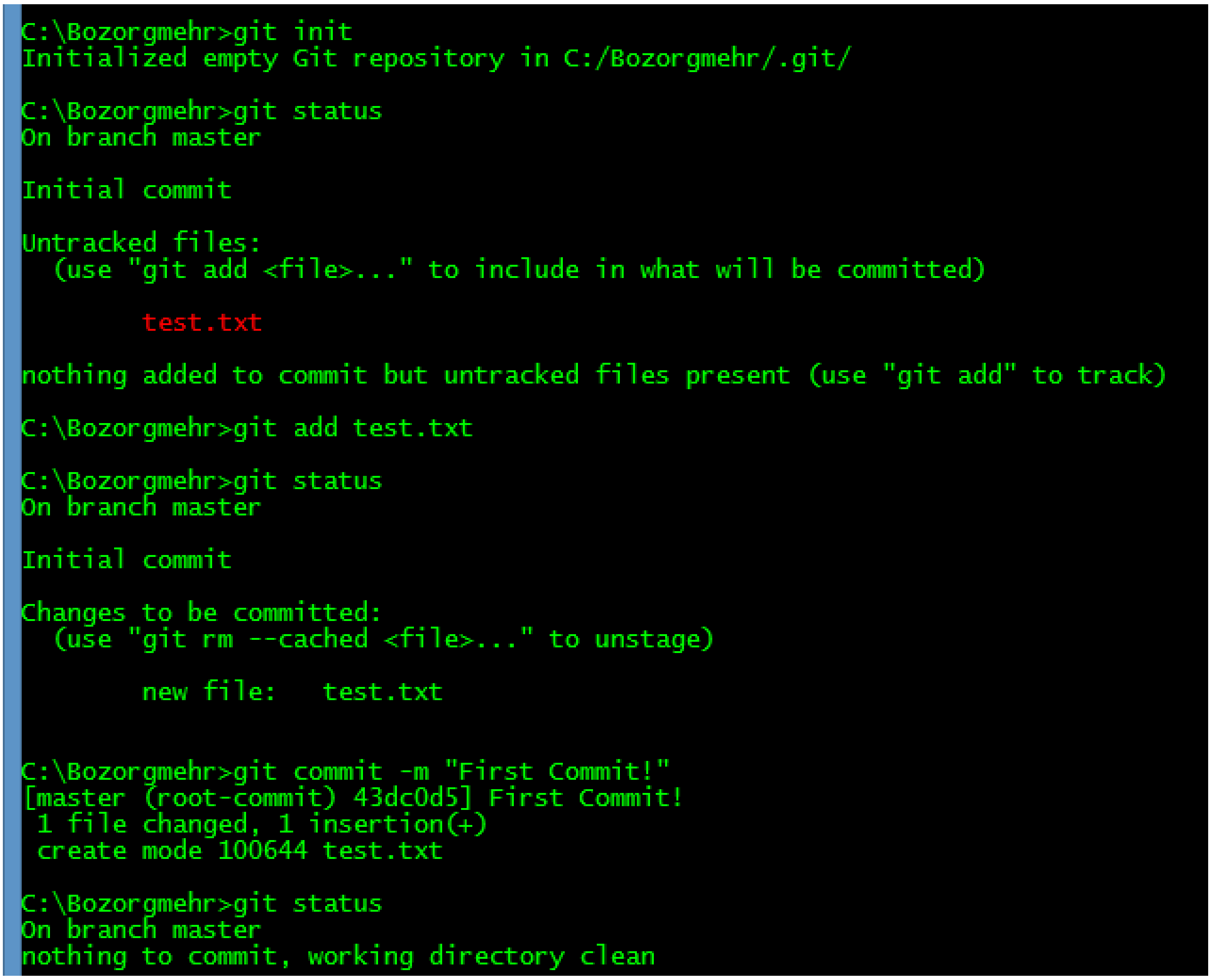
Git Homework:

Ali Ekramian (9012606007)

Git repository address:

`https://github.com/ekramian/Bozorgmehr.git`

A.



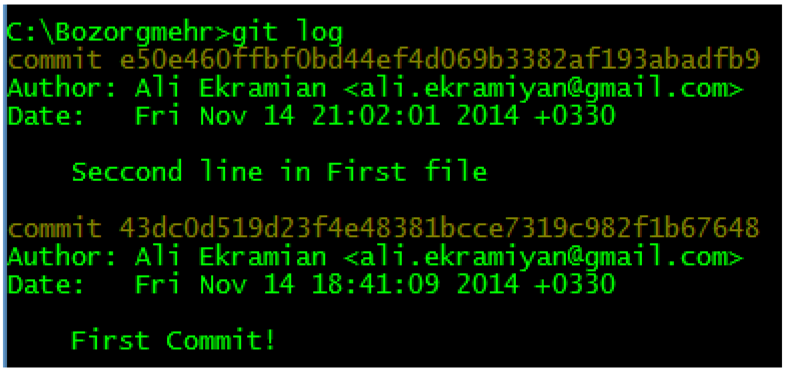
```
C:\Bozorgmehr>git init
Initialized empty Git repository in C:/Bozorgmehr/.git/

C:\Bozorgmehr>git status
On branch master

Initial commit

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

        test.txt

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

C:\Bozorgmehr>git add test.txt

C:\Bozorgmehr>git status
On branch master

Initial commit

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

        new file:   test.txt


C:\Bozorgmehr>git commit -m "First Commit!"
[master (root-commit) 43dc0d5] First Commit!
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt

C:\Bozorgmehr>git status
On branch master
nothing to commit, working directory clean
```

First when we initial an empty repository, the ‘test.txt’ is an untracked file which git doesn’t follow its changes.

When we run “git add test.txt” then we add ‘test.txt’ to staging area and we tell git that we want to include this file to our repository and git follow its changes.

After commit the staging area, git add a snapshot of added files to repository history and working directory would be clean of any uncommitted changes.

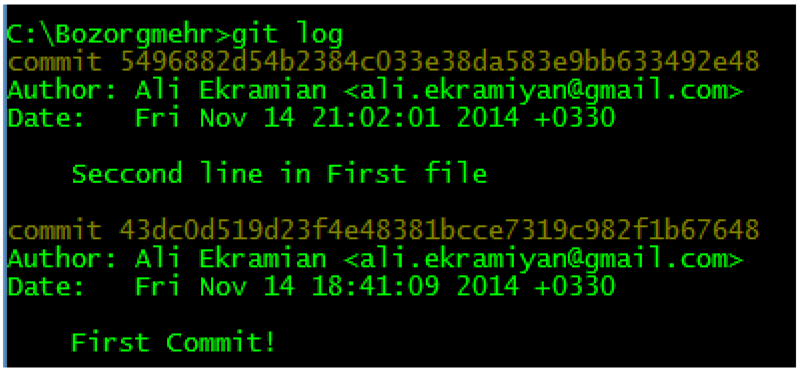
B.



```
C:\Bozorgmehr>git log
commit e50e460ffbf0bd44ef4d069b3382af193abadfb9
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:02:01 2014 +0330

    Seccond line in First file

commit 43dc0d519d23f4e48381bcce7319c982f1b67648
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 18:41:09 2014 +0330

    First Commit!
```

We have 4 line for each commit, display commit ID, Author, commit Date and commit message.

C.

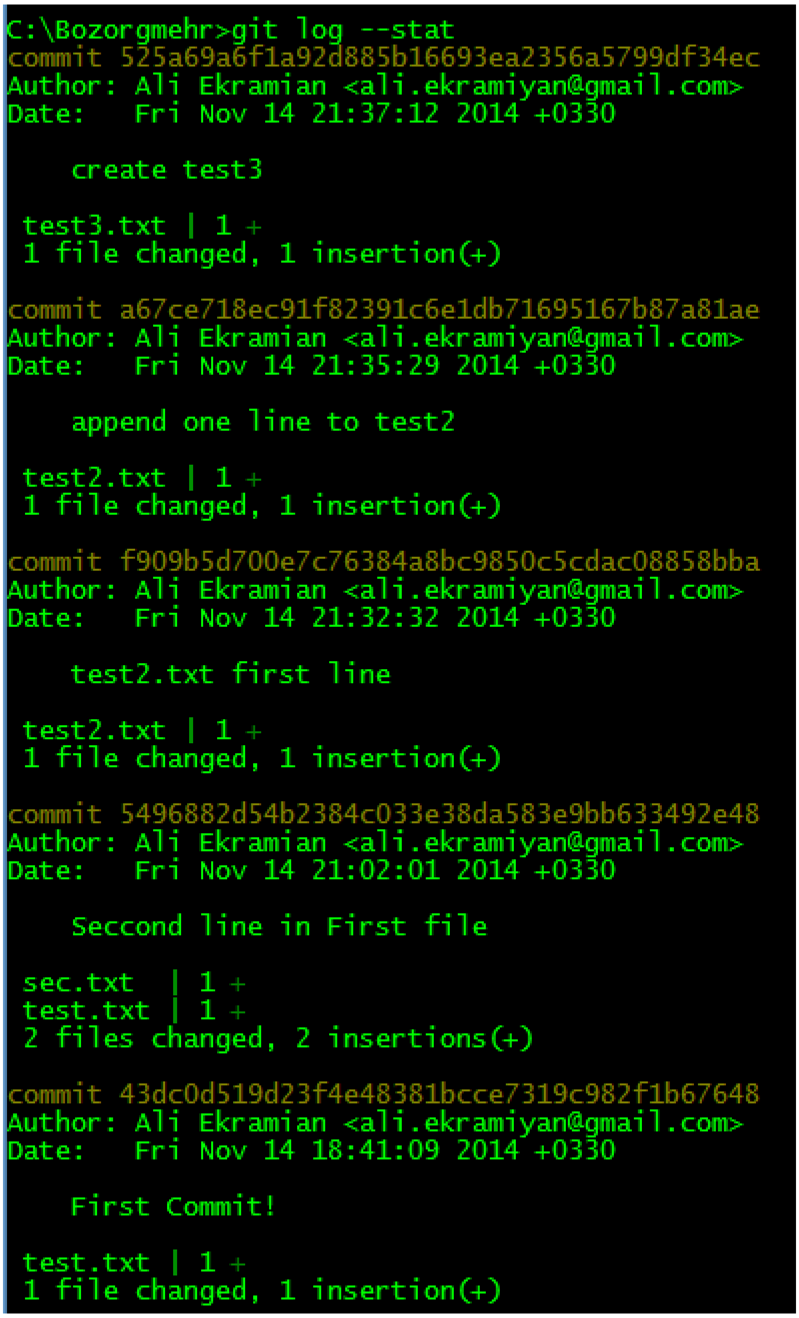


```
C:\Bozorgmehr>git log
commit 5496882d54b2384c033e38da583e9bb633492e48
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:02:01 2014 +0330

    Seccond line in First file

commit 43dc0d519d23f4e48381bcce7319c982f1b67648
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 18:41:09 2014 +0330

    First Commit!
```

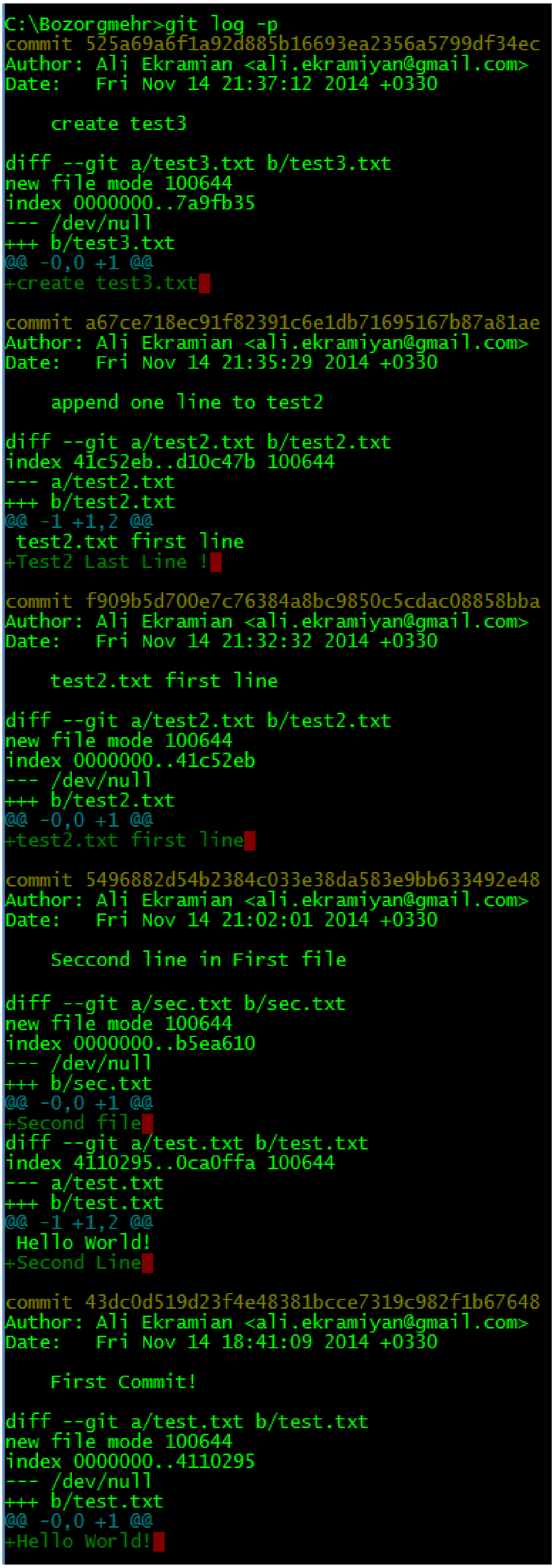
The last commit is replaced with a new commit that have the same snapshot of working directory of removed commit including ‘sec.txt’ file. This new commit have the same author, date and message (we can change message when we run ‘git commit --amend’) but it has a different commit ID (so it’s new commit and the last commit is deleted).

D.

We use ‘git log --stat’ or ‘git log -p’ for most detailed view.



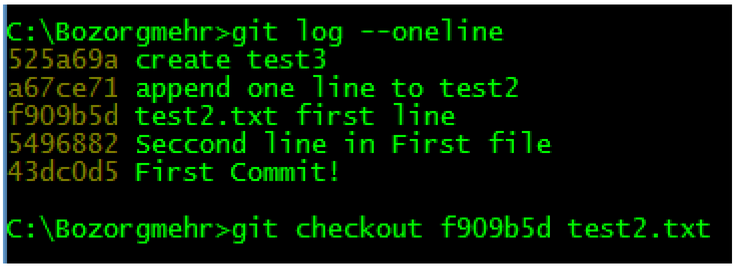
```
C:\Bozorgmehr>git log --stat
commit 525a69a6f1a92d885b16693ea2356a5799df34ec
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:37:12 2014 +0330

    create test3

 test3.txt | 1 +
 1 file changed, 1 insertion(+)

commit a67ce718ec91f82391c6e1db71695167b87a81ae
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:35:29 2014 +0330

    append one line to test2

 test2.txt | 1 +
 1 file changed, 1 insertion(+)

commit f909b5d700e7c76384a8bc9850c5cdac08858bba
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:32:32 2014 +0330

    test2.txt first line

 test2.txt | 1 +
 1 file changed, 1 insertion(+)

commit 5496882d54b2384c033e38da583e9bb633492e48
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:02:01 2014 +0330

    Seccond line in First file

 sec.txt  | 1 +
 test.txt | 1 +
 2 files changed, 2 insertions(+)

commit 43dc0d519d23f4e48381bcce7319c982f1b67648
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 18:41:09 2014 +0330

    First Commit!

 test.txt | 1 +
 1 file changed, 1 insertion(+)
```



```
C:\Bozorgmehr>git log -p
commit 525a69a6f1a92d885b16693ea2356a5799df34ec
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:37:12 2014 +0330

    create test3

diff --git a/test3.txt b/test3.txt
new file mode 100644
index 0000000..7a9fb35
--- /dev/null
+++ b/test3.txt
@@ -0,0 +1 @@
+create test3.txt

commit a67ce718ec91f82391c6e1db71695167b87a81ae
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:35:29 2014 +0330

    append one line to test2

diff --git a/test2.txt b/test2.txt
index 41c52eb..d10c47b 100644
--- a/test2.txt
+++ b/test2.txt
@@ -1 +1,2 @@
 test2.txt first line
+Test2 Last Line !

commit f909b5d700e7c76384a8bc9850c5cdac08858bba
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:32:32 2014 +0330

    test2.txt first line

diff --git a/test2.txt b/test2.txt
new file mode 100644
index 0000000..41c52eb
--- /dev/null
+++ b/test2.txt
@@ -0,0 +1 @@
+test2.txt first line

commit 5496882d54b2384c033e38da583e9bb633492e48
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:02:01 2014 +0330

    Seccond line in First file

diff --git a/sec.txt b/sec.txt
new file mode 100644
index 0000000..b5ea610
--- /dev/null
+++ b/sec.txt
@@ -0,0 +1 @@
+Second file
diff --git a/test.txt b/test.txt
index 4110295..0ca0ffa 100644
--- a/test.txt
+++ b/test.txt
@@ -1 +1,2 @@
 Hello World!
+Second Line

commit 43dc0d519d23f4e48381bcce7319c982f1b67648
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 18:41:09 2014 +0330

    First Commit!

diff --git a/test.txt b/test.txt
new file mode 100644
index 0000000..4110295
--- /dev/null
+++ b/test.txt
@@ -0,0 +1 @@
+Hello World!
```

E1.

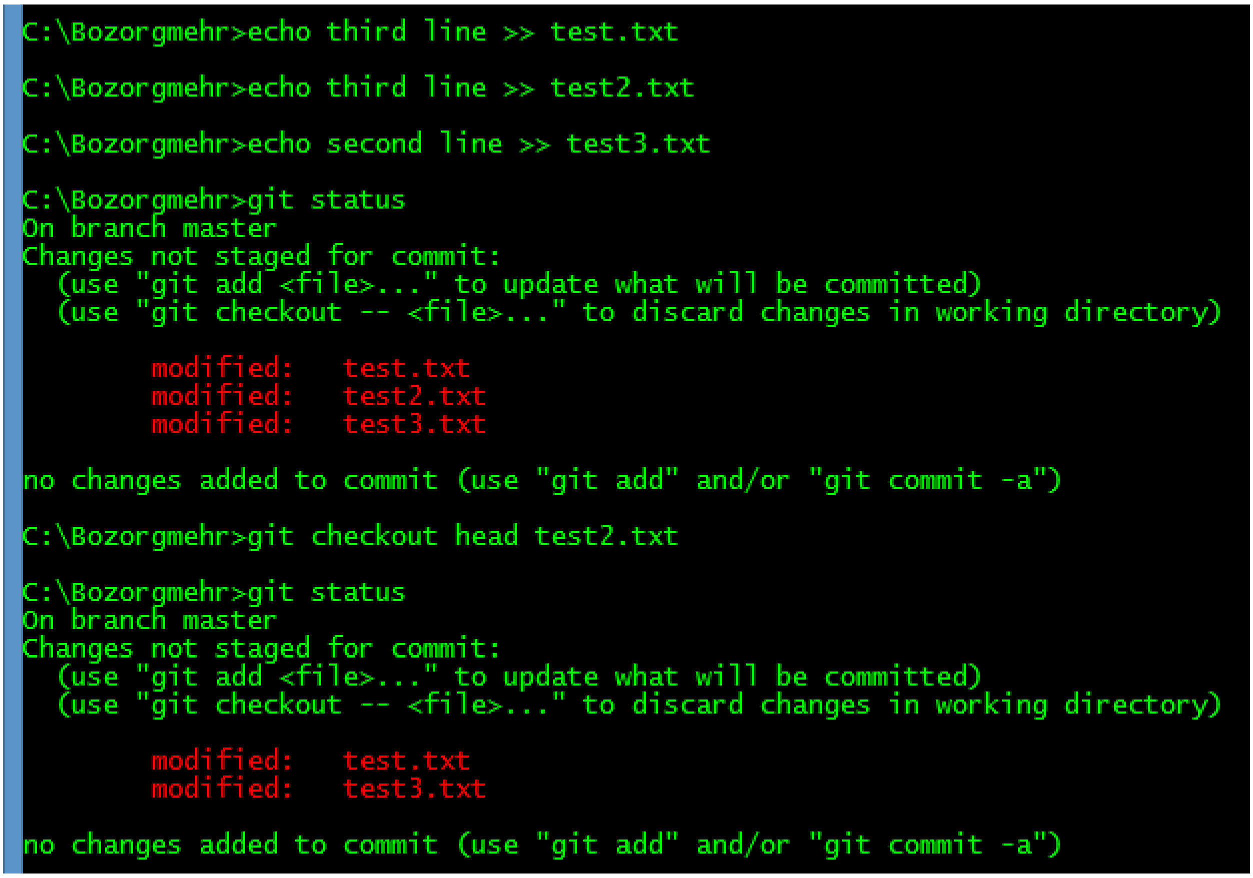
We use ‘git checkout’ command:



```
C:\Bozorgmehr>git log --oneline
525a69a create test3
a67ce71 append one line to test2
f909b5d test2.txt first line
5496882 Seccond line in First file
43dc0d5 First Commit!

C:\Bozorgmehr>git checkout f909b5d test2.txt
```

E2.

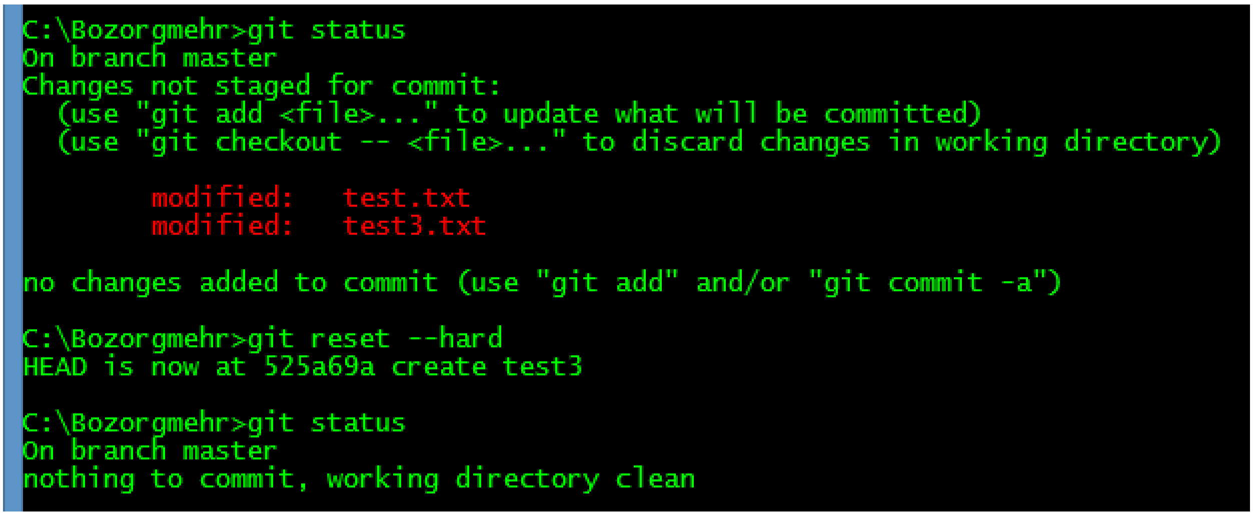
We use ‘git checkout head tets2.txt’



```
C:\Bozorgmehr>echo third line >> test.txt

C:\Bozorgmehr>echo third line >> test2.txt

C:\Bozorgmehr>echo second line >> test3.txt

C:\Bozorgmehr>git status
On branch 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:   test.txt
        modified:   test2.txt
        modified:   test3.txt

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

C:\Bozorgmehr>git checkout head test2.txt

C:\Bozorgmehr>git status
On branch 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:   test.txt
        modified:   test3.txt

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

F.

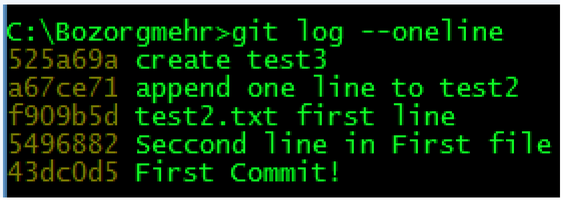
We run ‘git reset --hard’



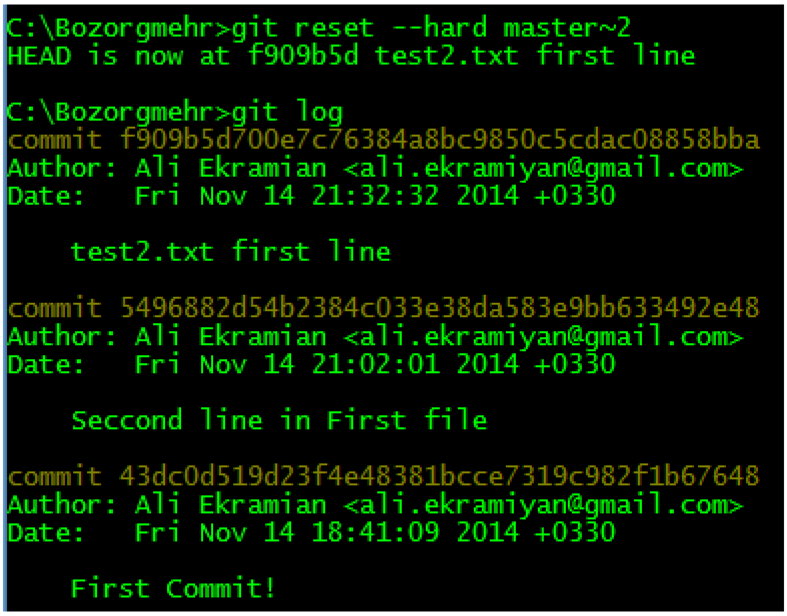
```
C:\Bozorgmehr>git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   test.txt
        modified:   test3.txt

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

C:\Bozorgmehr>git reset --hard
HEAD is now at 525a69a create test3

C:\Bozorgmehr>git status
On branch master
nothing to commit, working directory clean
```

G.

We run “git reset --hard master~2”. Thins remove two last commits.



```
C:\Bozorgmehr>git log --oneline
525a69a create test3
a67ce71 append one line to test2
f909b5d test2.txt first line
5496882 Seccond line in First file
43dc0d5 First Commit!
```

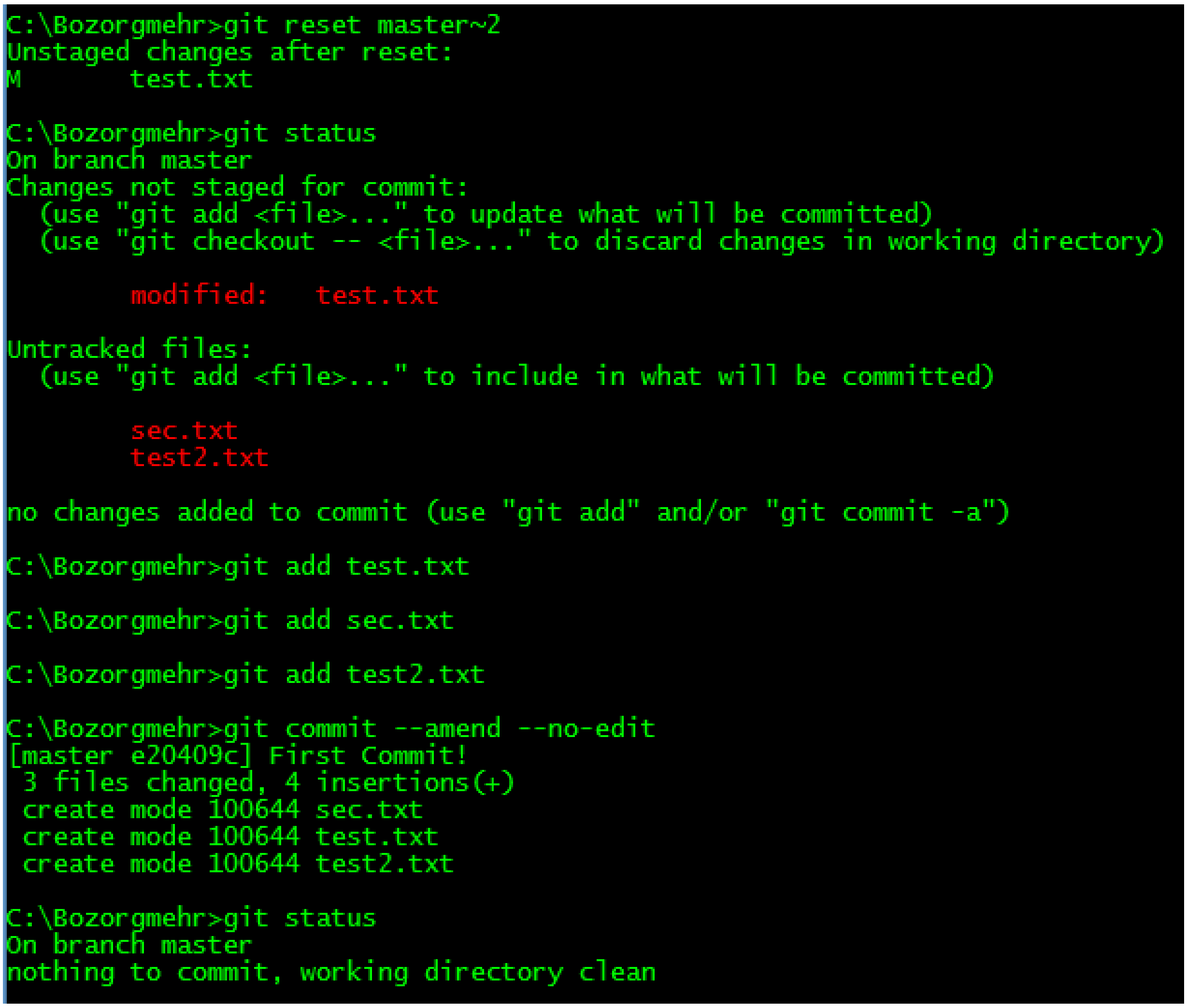


```
C:\Bozorgmehr>git reset --hard master~2
HEAD is now at f909b5d test2.txt first line

C:\Bozorgmehr>git log
commit f909b5d700e7c76384a8bc9850c5cdac08858bba
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:32:32 2014 +0330

    test2.txt first line

commit 5496882d54b2384c033e38da583e9bb633492e48
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 21:02:01 2014 +0330

    Seccond line in First file

commit 43dc0d519d23f4e48381bcce7319c982f1b67648
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Fri Nov 14 18:41:09 2014 +0330

    First Commit!
```

H.

Run:

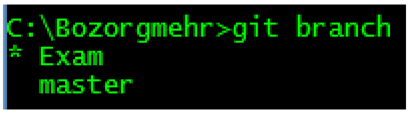
```
git reset master~2
git add test.txt
git add test2.txt
git add sec.txt
git commit –amend –no-edit
```



```
C:\Bozorgmehr>git reset master~2
Unstaged changes after reset:
M       test.txt

C:\Bozorgmehr>git status
On branch 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:   test.txt

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

        sec.txt
        test2.txt

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

C:\Bozorgmehr>git add test.txt

C:\Bozorgmehr>git add sec.txt

C:\Bozorgmehr>git add test2.txt

C:\Bozorgmehr>git commit --amend --no-edit
[master e20409c] First Commit!
 3 files changed, 4 insertions(+)
 create mode 100644 sec.txt
 create mode 100644 test.txt
 create mode 100644 test2.txt

C:\Bozorgmehr>git status
On branch master
nothing to commit, working directory clean
```

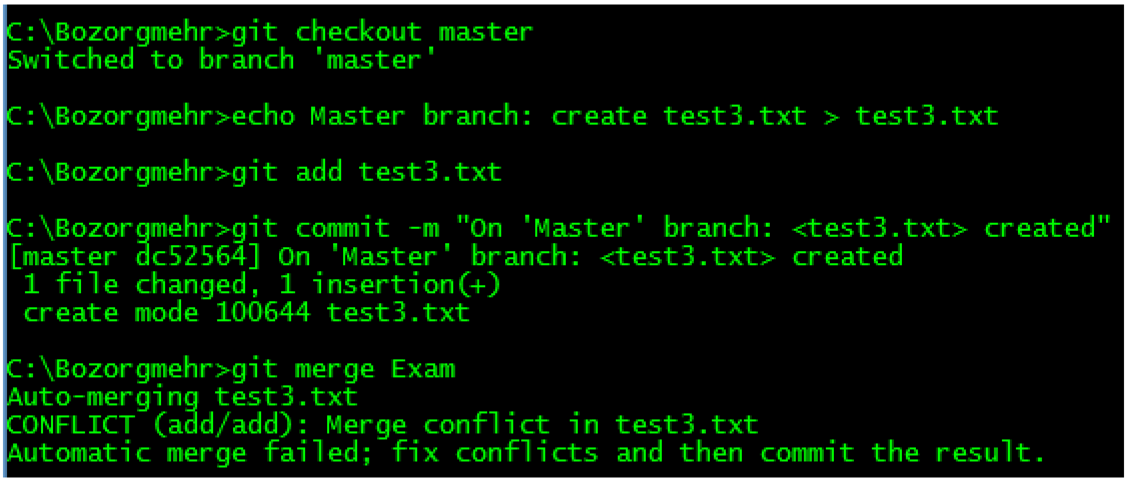
I.

```
git branch
```



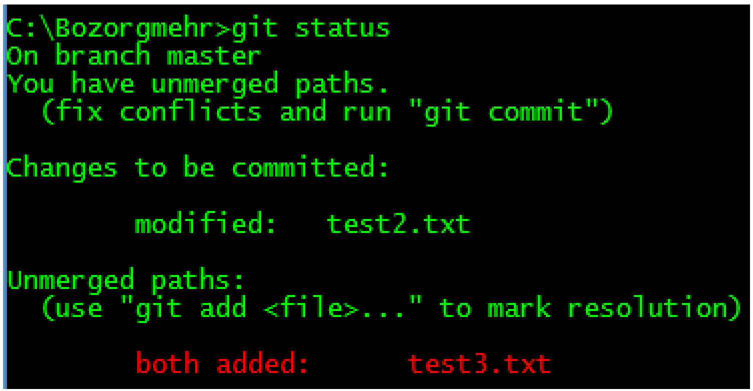
```
C:\Bozorgmehr>git branch
* Exam
  master
```

J.

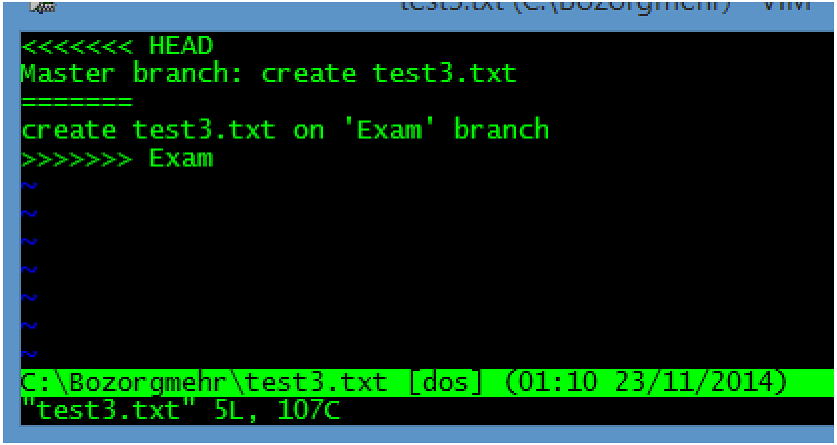


```
C:\Bozorgmehr>git checkout master
Switched to branch 'master'

C:\Bozorgmehr>echo Master branch: create test3.txt > test3.txt

C:\Bozorgmehr>git add test3.txt

C:\Bozorgmehr>git commit -m "On 'Master' branch: <test3.txt> created"
[master dc52564] On 'Master' branch: <test3.txt> created
 1 file changed, 1 insertion(+)
 create mode 100644 test3.txt

C:\Bozorgmehr>git merge Exam
Auto-merging test3.txt
CONFLICT (add/add): Merge conflict in test3.txt
Automatic merge failed; fix conflicts and then commit the result.
```

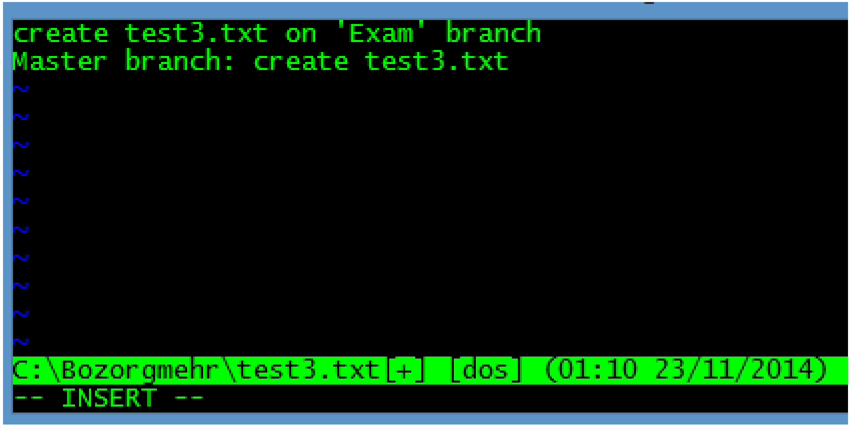
We have merge conflict in test3.txt. For continue merging, we should resolve conflicts: we just open files, edit it, and add it to staging area.



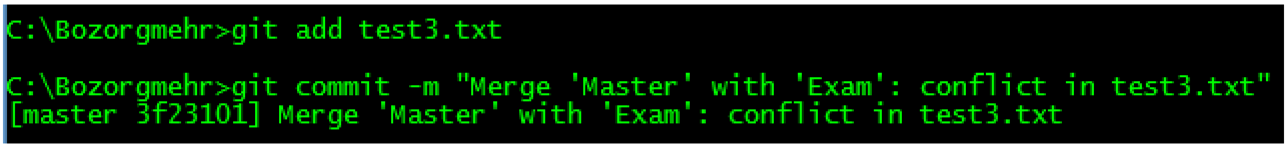
```
C:\Bozorgmehr>git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")

Changes to be committed:

        modified:   test2.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both added:      test3.txt
```



```
<<<<<<< HEAD
Master branch: create test3.txt
=======
create test3.txt on 'Exam' branch
>>>>>>> Exam
~
~
~
~
~
~
~
C:\Bozorgmehr\test3.txt [dos] (01:10 23/11/2014)
"test3.txt" 5L, 107C
```

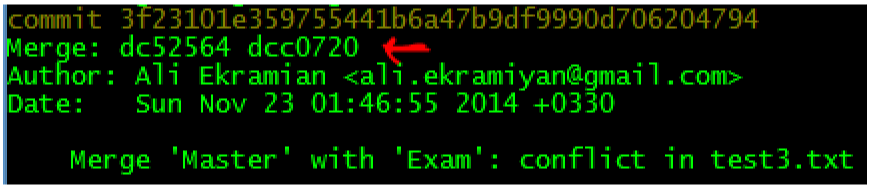


```
create test3.txt on 'Exam' branch
Master branch: create test3.txt
~
~
~
~
~
~
~
~
~
~
C:\Bozorgmehr\test3.txt[+] [dos] (01:10 23/11/2014)
-- INSERT --
```



```
C:\Bozorgmehr>git add test3.txt

C:\Bozorgmehr>git commit -m "Merge 'Master' with 'Exam': conflict in test3.txt"
[master 3f23101] Merge 'Master' with 'Exam': conflict in test3.txt
```

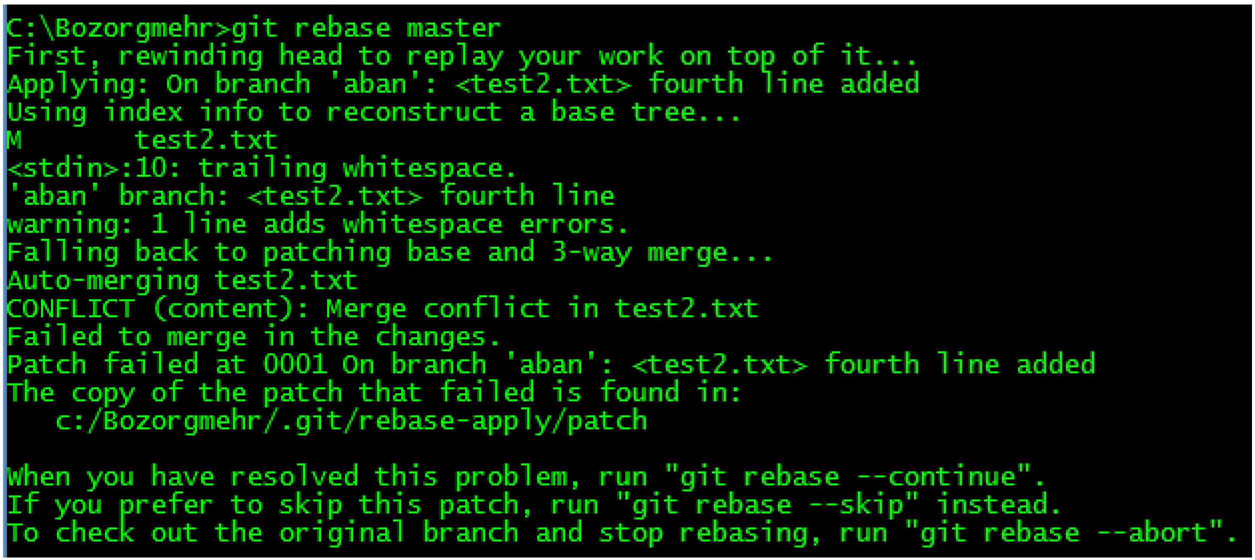
After merging, when we check commits log, we see that last commit is the merge commit:



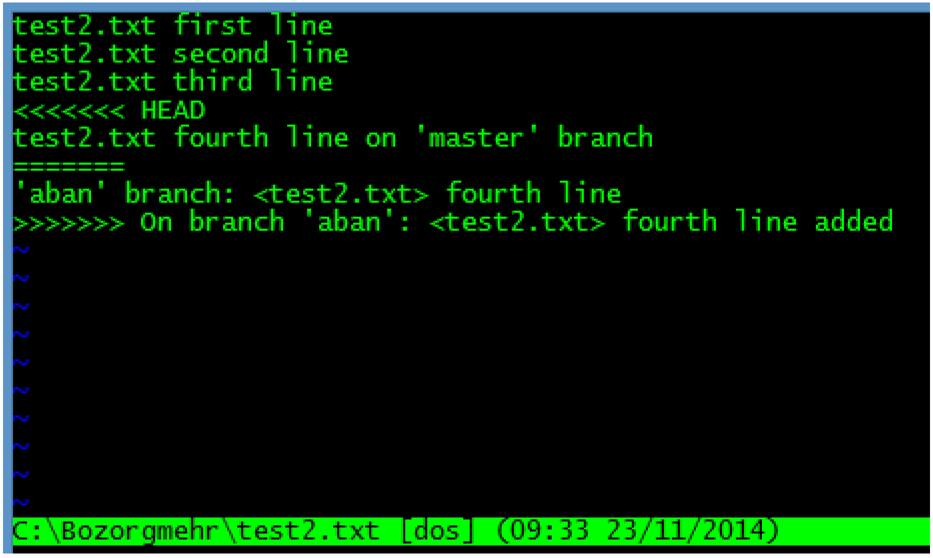
```
commit 3f23101e359755441b6a47b9df9990d706204794
Merge: dc52564 dcc0720
Author: Ali Ekramian <ali.ekramiyan@gmail.com>
Date:   Sun Nov 23 01:46:55 2014 +0330

    Merge 'Master' with 'Exam': conflict in test3.txt
```

K.

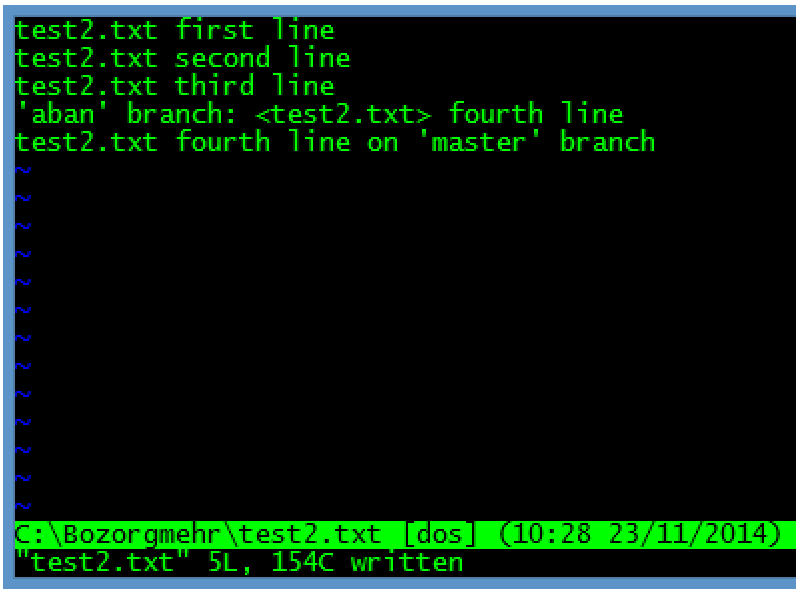
We get conflict message: “CONFLICT (content): Merge conflict in test2.txt”. We edit ‘test2.txt’ to resolve conflict, add it to staging area and run ‘git rebase --continue’



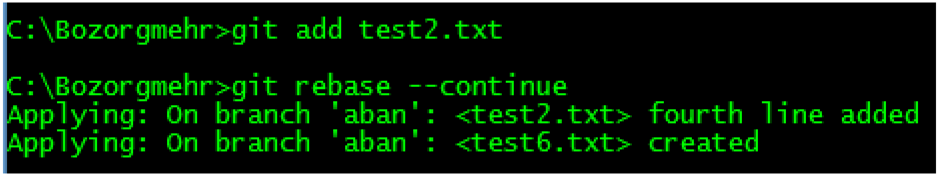
```
C:\Bozorgmehr>git rebase master
First, rewinding head to replay your work on top of it...
Applying: On branch 'aban': <test2.txt> fourth line added
Using index info to reconstruct a base tree...
M       test2.txt
<stdin>:10: trailing whitespace.
'aban' branch: <test2.txt> fourth line
warning: 1 line adds whitespace errors.
Falling back to patching base and 3-way merge...
Auto-merging test2.txt
CONFLICT (content): Merge conflict in test2.txt
Failed to merge in the changes.
Patch failed at 0001 On branch 'aban': <test2.txt> fourth line added
The copy of the patch that failed is found in:
   c:/Bozorgmehr/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".
```



```
test2.txt first line
test2.txt second line
test2.txt third line
<<<<<<< HEAD
test2.txt fourth line on 'master' branch
=======
'aban' branch: <test2.txt> fourth line
>>>>>>> On branch 'aban': <test2.txt> fourth line added
~
~
~
~
~
~
~
~
~
~
C:\Bozorgmehr\test2.txt [dos] (09:33 23/11/2014)
```



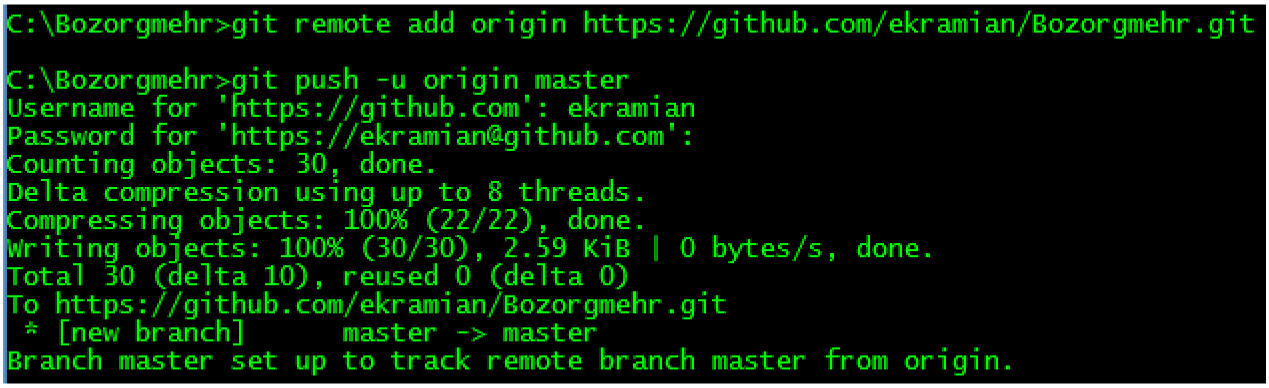
```
test2.txt first line
test2.txt second line
test2.txt third line
'aban' branch: <test2.txt> fourth line
test2.txt fourth line on 'master' branch
~
~
~
~
~
~
~
~
~
~
~
~
~
C:\Bozorgmehr\test2.txt [dos] (10:28 23/11/2014)
"test2.txt" 5L, 154C written
```



```
C:\Bozorgmehr>git add test2.txt

C:\Bozorgmehr>git rebase --continue
Applying: On branch 'aban': <test2.txt> fourth line added
Applying: On branch 'aban': <test6.txt> created
```

L.

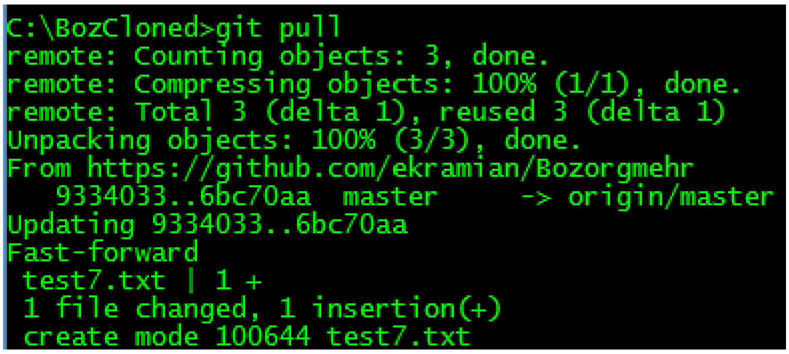
We use ‘git remote …’ command to add connection to github repository, and we use ‘git push …’ to save changes on github server.



```
C:\Bozorgmehr>git remote add origin https://github.com/ekramian/Bozorgmehr.git

C:\Bozorgmehr>git push -u origin master
Username for 'https://github.com': ekramian
Password for 'https://ekramian@github.com':
Counting objects: 30, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (22/22), done.
Writing objects: 100% (30/30), 2.59 KiB | 0 bytes/s, done.
Total 30 (delta 10), reused 0 (delta 0)
To https://github.com/ekramian/Bozorgmehr.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

M.

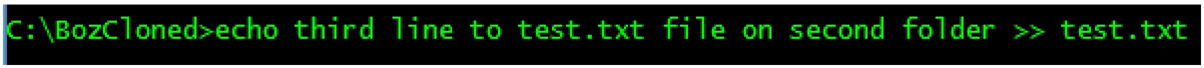
After running ‘git pull’, test7.txt will create.



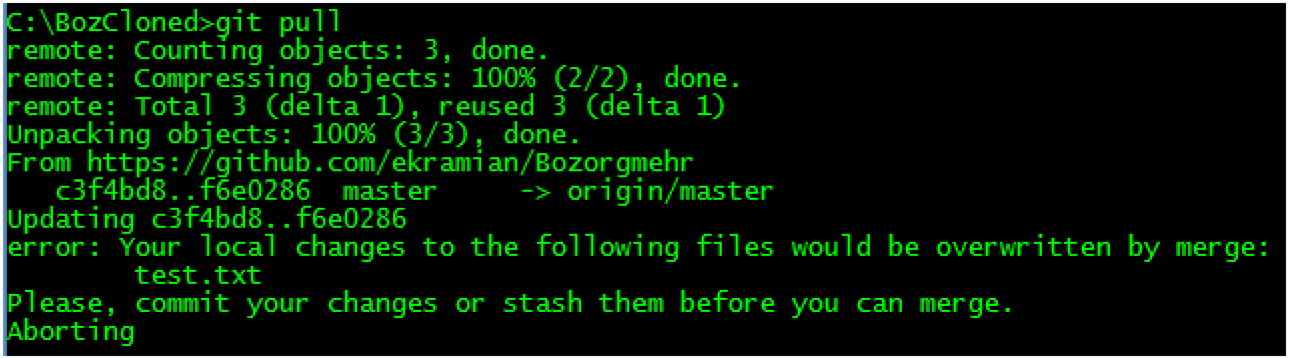
```
C:\BozCloned>git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1)
Unpacking objects: 100% (3/3), done.
From https://github.com/ekramian/Bozorgmehr
   9334033..6bc70aa  master     -> origin/master
Updating 9334033..6bc70aa
Fast-forward
 test7.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 test7.txt
```

N.

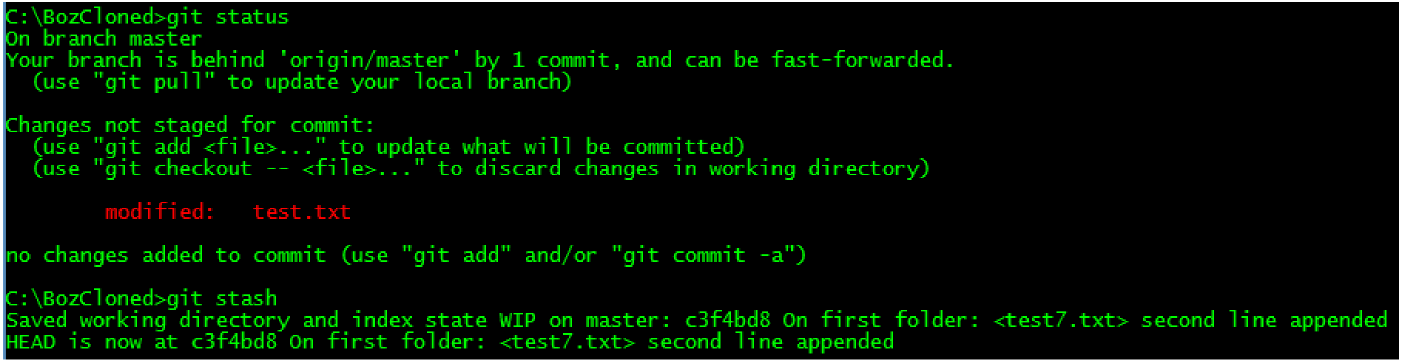
On first run on git pull, git abort pulling because of same change on same file on different repositories.
After run ‘git stash apply’, again we get merge conflict message in test.txt. we should solve conflicts and commit it.



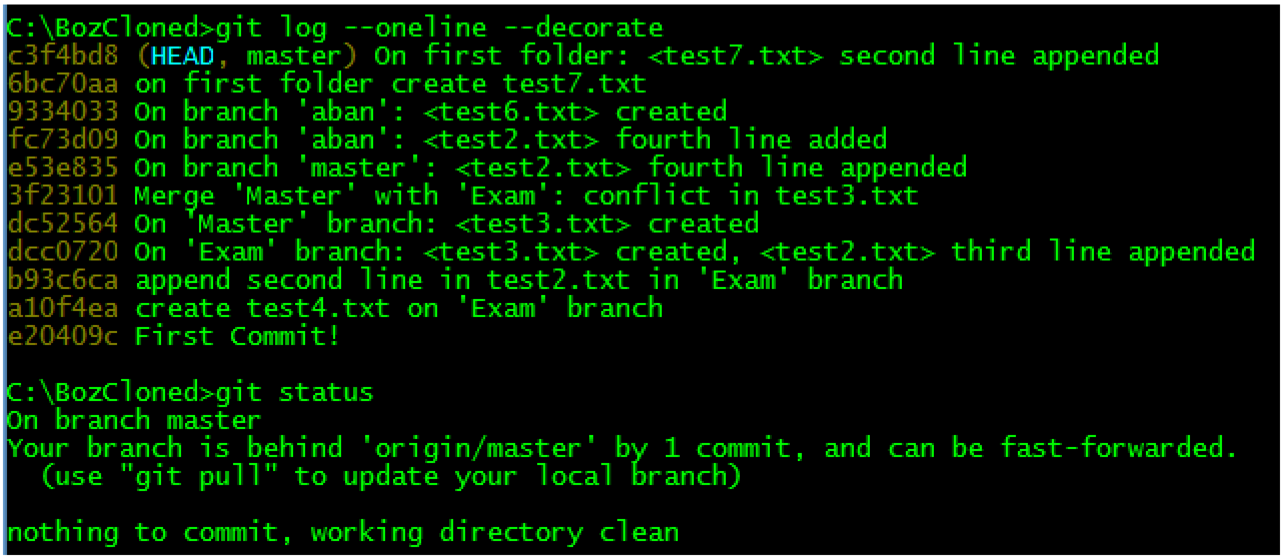
```
C:\BozCloned>echo third line to test.txt file on second folder >> test.txt
```



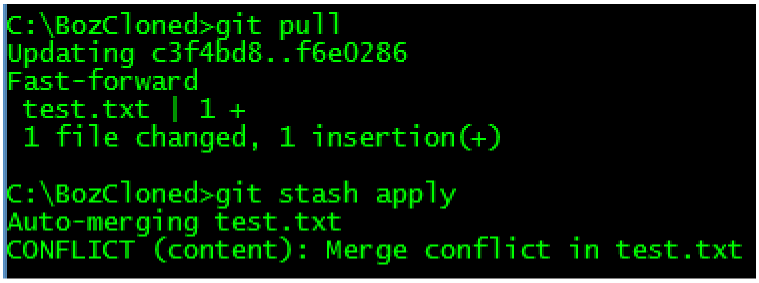
```
C:\BozCloned>git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1)
Unpacking objects: 100% (3/3), done.
From https://github.com/ekramian/Bozorgmehr
   c3f4bd8..f6e0286  master     -> origin/master
Updating c3f4bd8..f6e0286
error: Your local changes to the following files would be overwritten by merge:
        test.txt
Please, commit your changes or stash them before you can merge.
Aborting
```



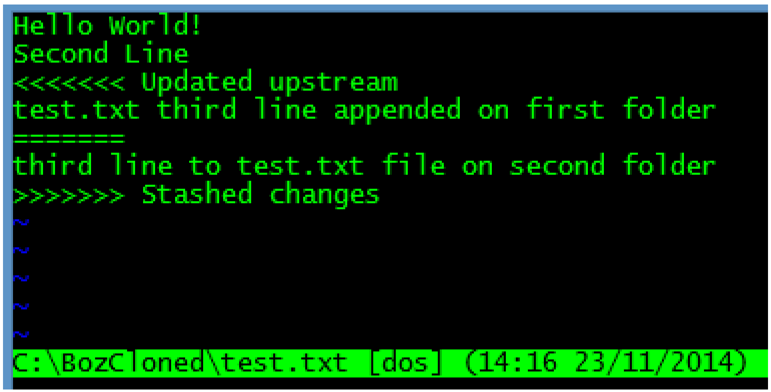
```
C:\BozCloned>git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

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:   test.txt

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

C:\BozCloned>git stash
Saved working directory and index state WIP on master: c3f4bd8 On first folder: <test7.txt> second line appended
HEAD is now at c3f4bd8 On first folder: <test7.txt> second line appended
```



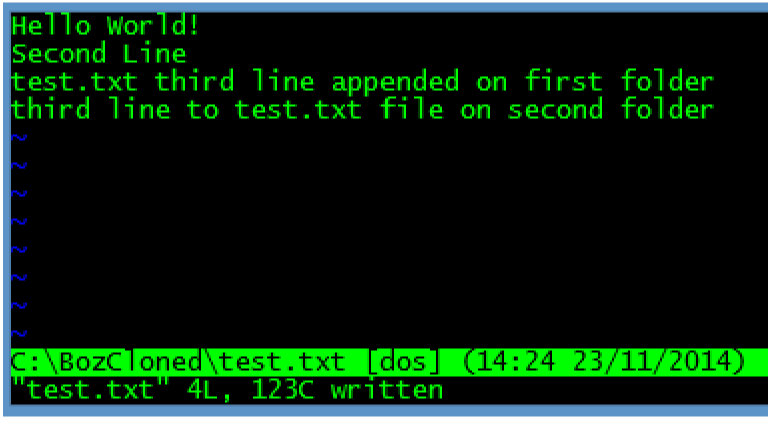
```
C:\BozCloned>git log --oneline --decorate
c3f4bd8 (HEAD, master) On first folder: <test7.txt> second line appended
6bc70aa on first folder create test7.txt
9334033 On branch 'aban': <test6.txt> created
fc73d09 On branch 'aban': <test2.txt> fourth line added
e53e835 On branch 'master': <test2.txt> fourth line appended
3f23101 Merge 'Master' with 'Exam': conflict in test3.txt
dc52564 On 'Master' branch: <test3.txt> created
dcc0720 On 'Exam' branch: <test3.txt> created, <test2.txt> third line appended
b93c6ca append second line in test2.txt in 'Exam' branch
a10f4ea create test4.txt on 'Exam' branch
e20409c First Commit!

C:\BozCloned>git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working directory clean
```



```
C:\BozCloned>git pull
Updating c3f4bd8..f6e0286
Fast-forward
 test.txt | 1 +
 1 file changed, 1 insertion(+)

C:\BozCloned>git stash apply
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
```

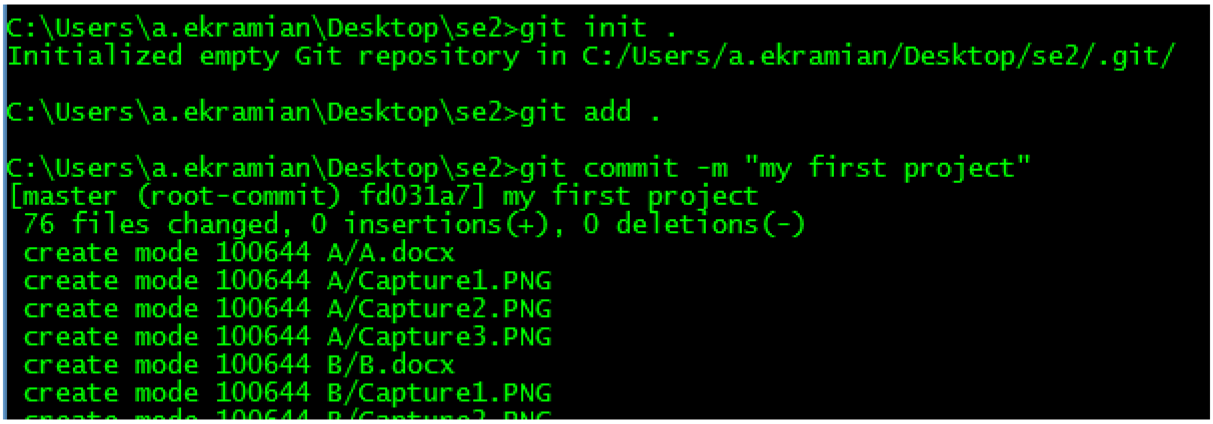


```
Hello World!
Second Line
<<<<<<< Updated upstream
test.txt third line appended on first folder
=======
third line to test.txt file on second folder
>>>>>>> Stashed changes
~
~
~
~
~
C:\BozCloned\test.txt [dos] (14:16 23/11/2014)
```

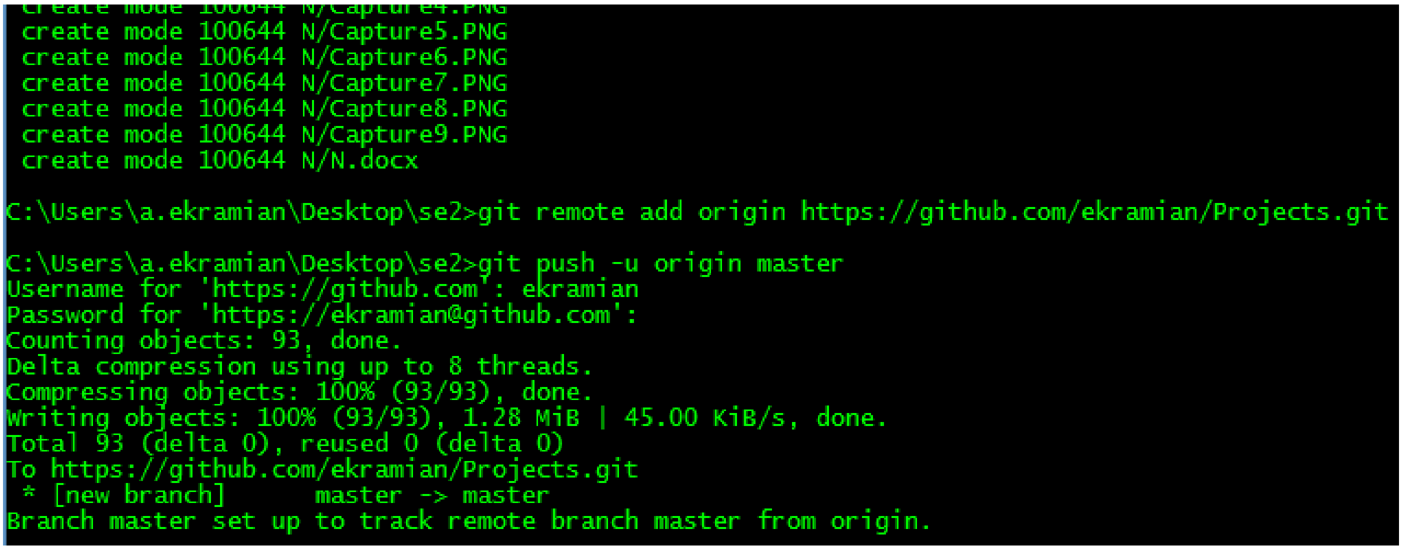


```
Hello World!
Second Line
test.txt third line appended on first folder
third line to test.txt file on second folder
~
~
~
~
~
~
~
~
C:\BozCloned\test.txt [dos] (14:24 23/11/2014)
"test.txt" 4L, 123C written
```

O.



```
C:\Users\a.ekramian\Desktop\se2>git init .
Initialized empty Git repository in C:/Users/a.ekramian/Desktop/se2/.git/

C:\Users\a.ekramian\Desktop\se2>git add .

C:\Users\a.ekramian\Desktop\se2>git commit -m "my first project"
[master (root-commit) fd031a7] my first project
 76 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 A/A.docx
 create mode 100644 A/Capture1.PNG
 create mode 100644 A/Capture2.PNG
 create mode 100644 A/Capture3.PNG
 create mode 100644 B/B.docx
 create mode 100644 B/Capture1.PNG
 create mode 100644 B/Capture2.PNG
```



```
 create mode 100644 N/Capture4.PNG
 create mode 100644 N/Capture5.PNG
 create mode 100644 N/Capture6.PNG
 create mode 100644 N/Capture7.PNG
 create mode 100644 N/Capture8.PNG
 create mode 100644 N/Capture9.PNG
 create mode 100644 N/N.docx

C:\Users\a.ekramian\Desktop\se2>git remote add origin https://github.com/ekramian/Projects.git

C:\Users\a.ekramian\Desktop\se2>git push -u origin master
Username for 'https://github.com': ekramian
Password for 'https://ekramian@github.com':
Counting objects: 93, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (93/93), done.
Writing objects: 100% (93/93), 1.28 MiB | 45.00 KiB/s, done.
Total 93 (delta 0), reused 0 (delta 0)
To https://github.com/ekramian/Projects.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```