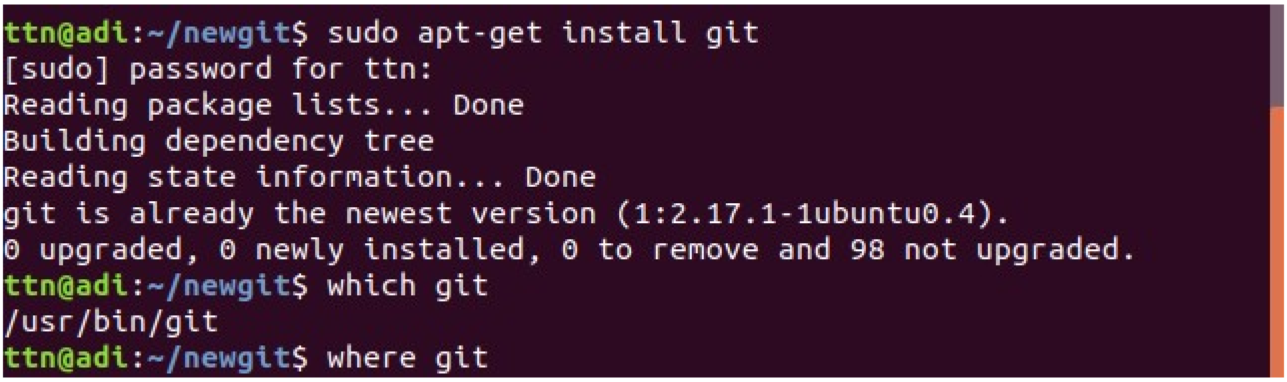
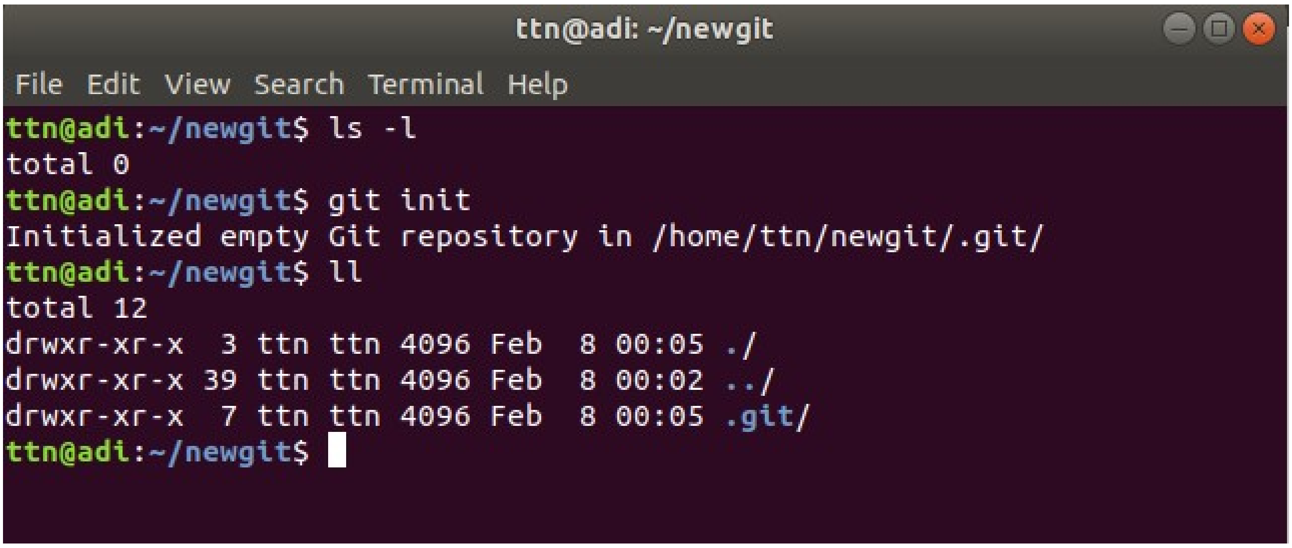
Introduction to version control and git assignment

Question 1:
Git setup



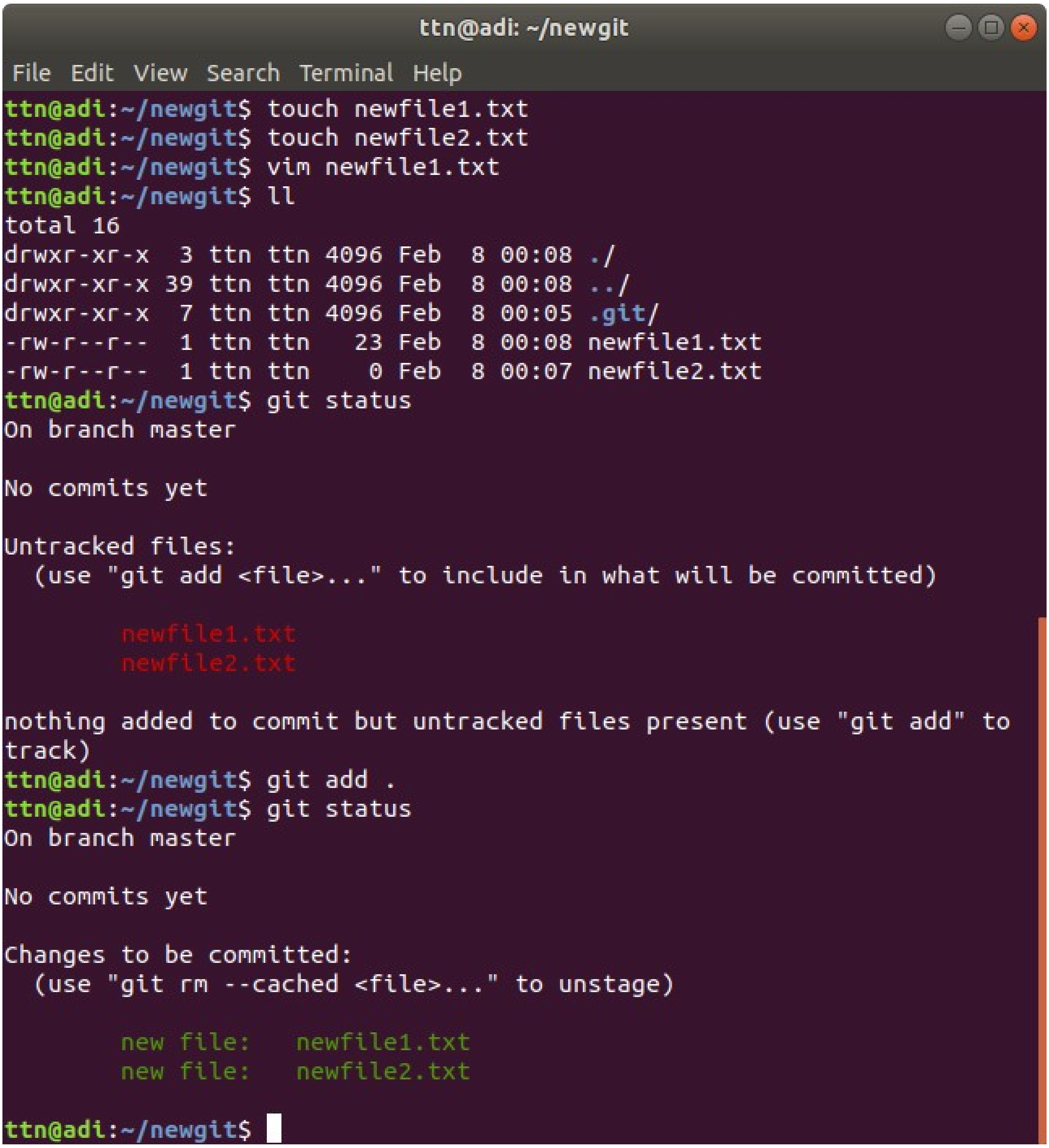
```
ttn@adi:~/newgit$ sudo apt-get install git
[sudo] password for ttn:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 98 not upgraded.
ttn@adi:~/newgit$ which git
/usr/bin/git
ttn@adi:~/newgit$ where git
```

Question 2:
Initialize a git repository



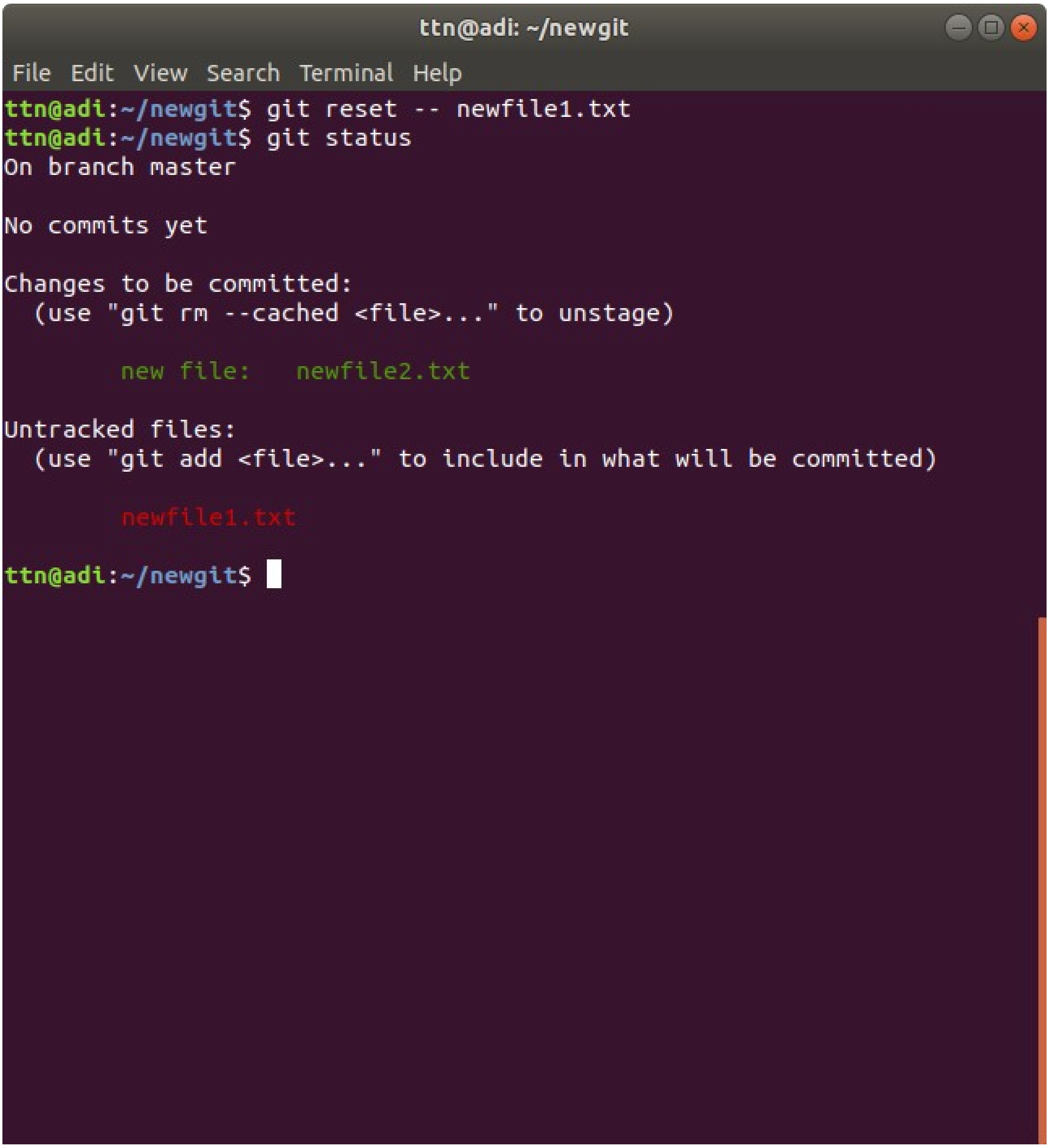
```
ttn@adi:~/newgit$ ls -l
total 0
ttn@adi:~/newgit$ git init
Initialized empty Git repository in /home/ttn/newgit/.git/
ttn@adi:~/newgit$ ll
total 12
drwxr-xr-x  3 ttn ttn 4096 Feb  8 00:05 ./
drwxr-xr-x 39 ttn ttn 4096 Feb  8 00:02 ../
drwxr-xr-x  7 ttn ttn 4096 Feb  8 00:05 .git/
ttn@adi:~/newgit$ 
```

Question 3:
Add files to the repository



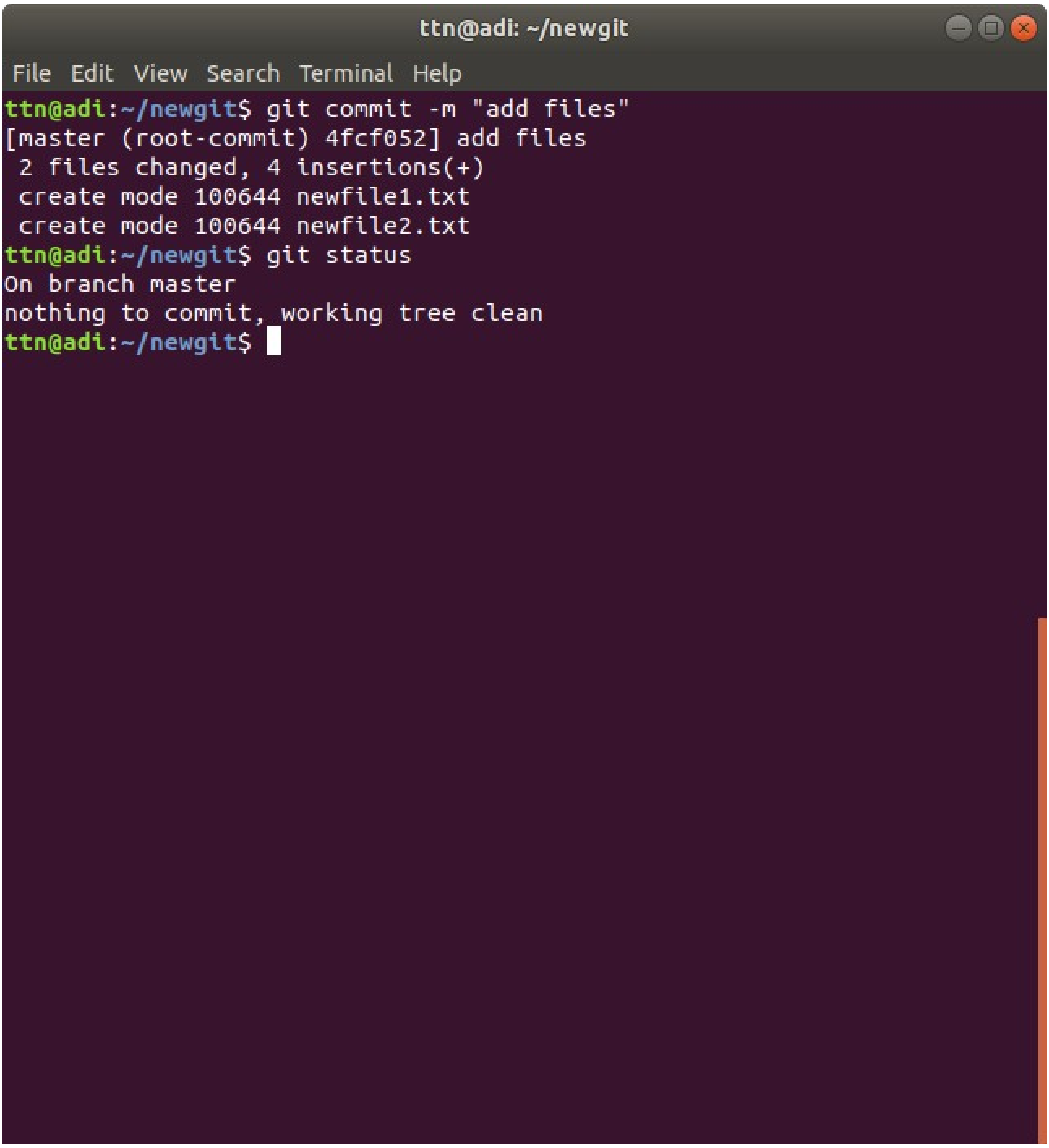
```
ttn@adi:~/newgit$ touch newfile1.txt
ttn@adi:~/newgit$ touch newfile2.txt
ttn@adi:~/newgit$ vim newfile1.txt
ttn@adi:~/newgit$ ll
total 16
drwxr-xr-x  3 ttn ttn 4096 Feb  8 00:08 ./
drwxr-xr-x 39 ttn ttn 4096 Feb  8 00:08 ../
drwxr-xr-x  7 ttn ttn 4096 Feb  8 00:05 .git/
-rw-r--r--  1 ttn ttn   23 Feb  8 00:08 newfile1.txt
-rw-r--r--  1 ttn ttn    0 Feb  8 00:07 newfile2.txt
ttn@adi:~/newgit$ git status
On branch master

No commits yet

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

        newfile1.txt
        newfile2.txt

nothing added to commit but untracked files present (use "git add" to
track)
ttn@adi:~/newgit$ git add .
ttn@adi:~/newgit$ git status
On branch master

No commits yet

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

        new file:   newfile1.txt
        new file:   newfile2.txt

ttn@adi:~/newgit$ 
```

Question 4:
Unstage one file



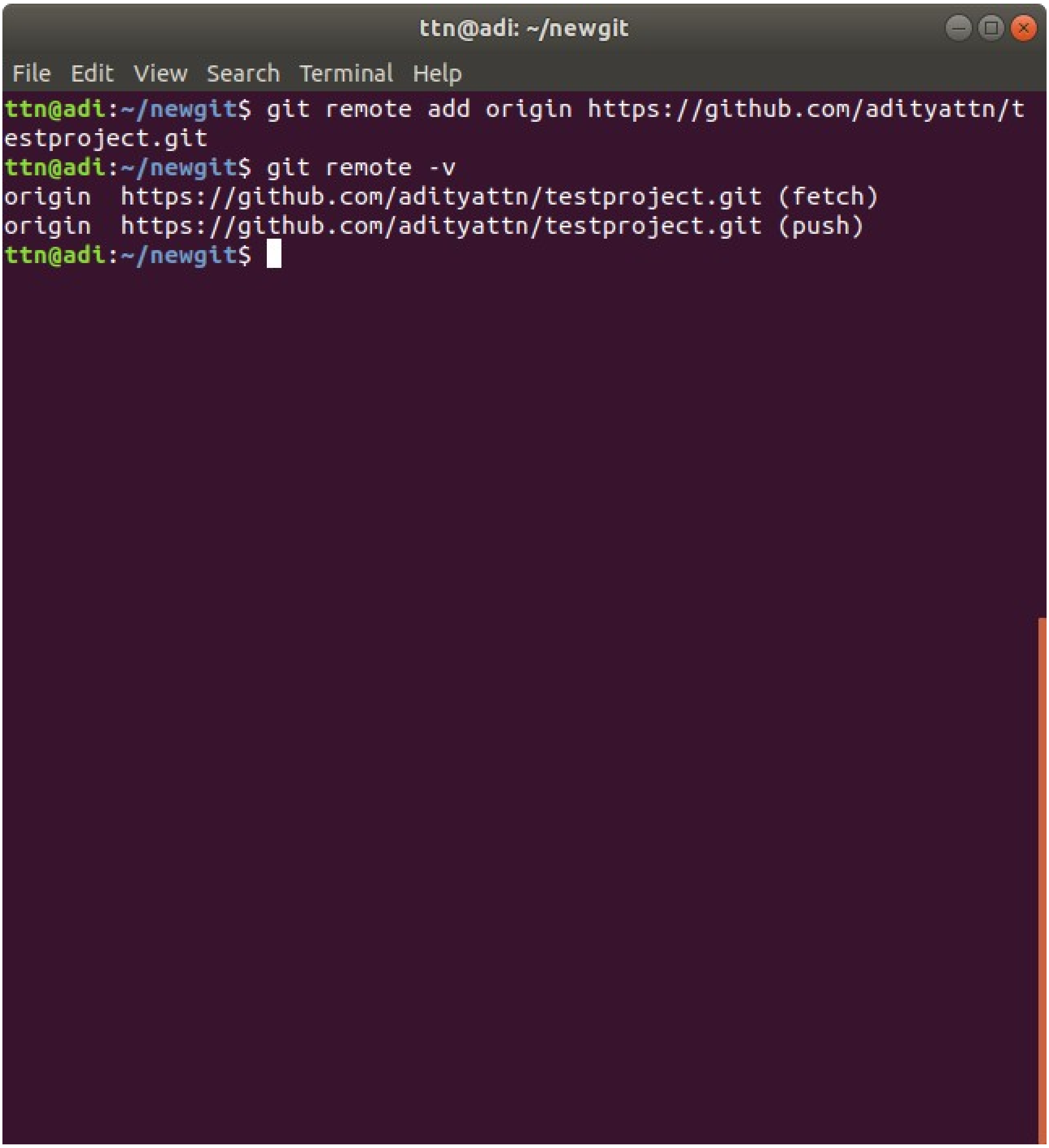
```
ttn@adi:~/newgit$ git reset -- newfile1.txt
ttn@adi:~/newgit$ git status
On branch master

No commits yet

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

        new file:   newfile2.txt

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

        newfile1.txt

ttn@adi:~/newgit$ 
```

Question 5:
Commit the file



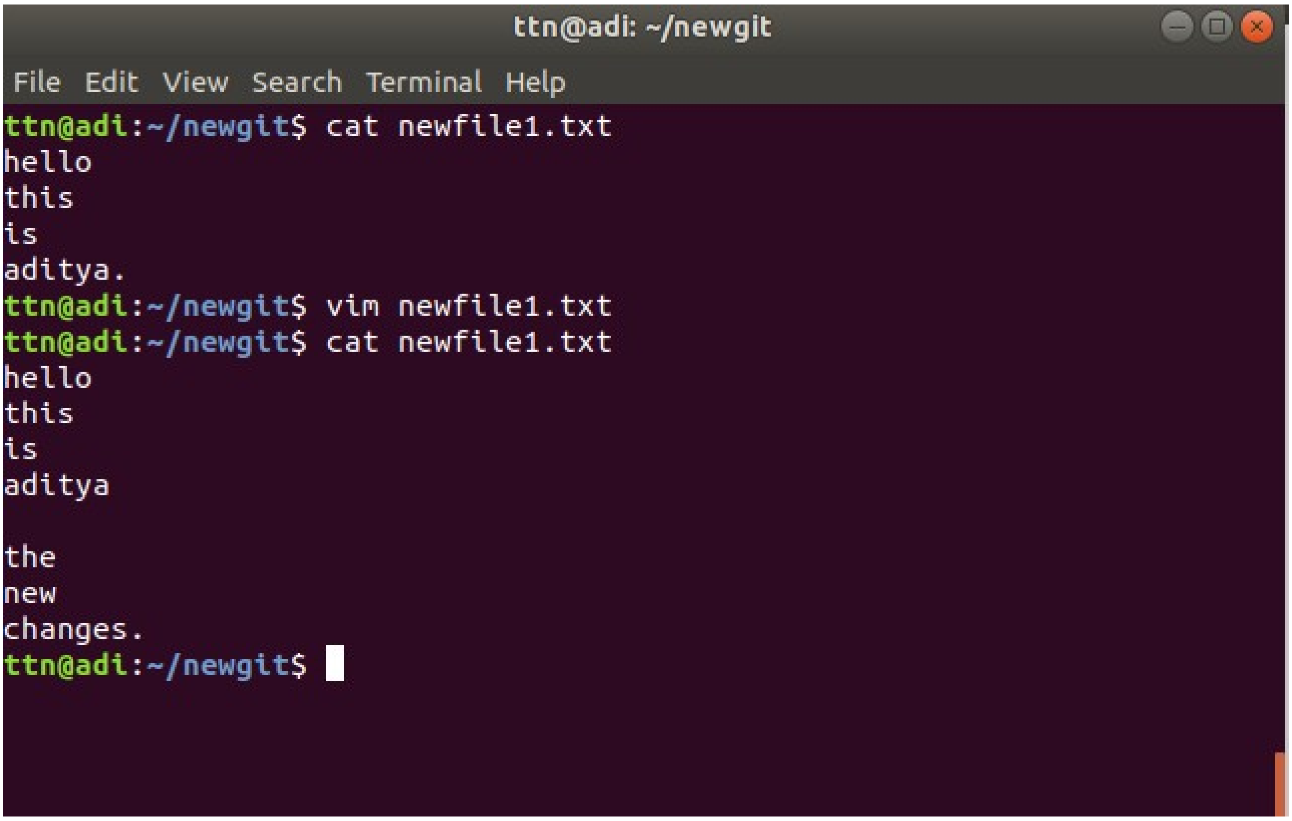
```
ttn@adi:~/newgit$ git commit -m "add files"
[master (root-commit) 4fcf052] add files
 2 files changed, 4 insertions(+)
 create mode 100644 newfile1.txt
 create mode 100644 newfile2.txt
ttn@adi:~/newgit$ git status
On branch master
nothing to commit, working tree clean
ttn@adi:~/newgit$
```

6. Add a remote



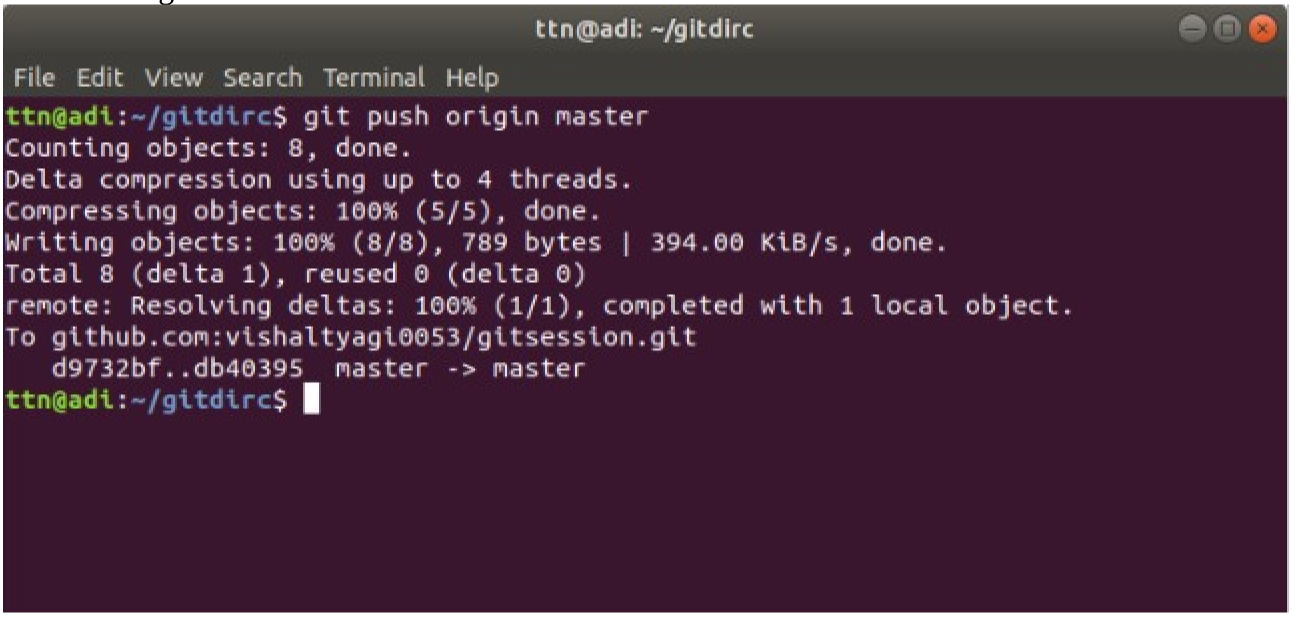
```
ttn@adi:~/newgit$ git remote add origin https://github.com/adityattn/t
estproject.git
ttn@adi:~/newgit$ git remote -v
origin  https://github.com/adityattn/testproject.git (fetch)
origin  https://github.com/adityattn/testproject.git (push)
ttn@adi:~/newgit$ 
```

7.Undo changes to a particular file



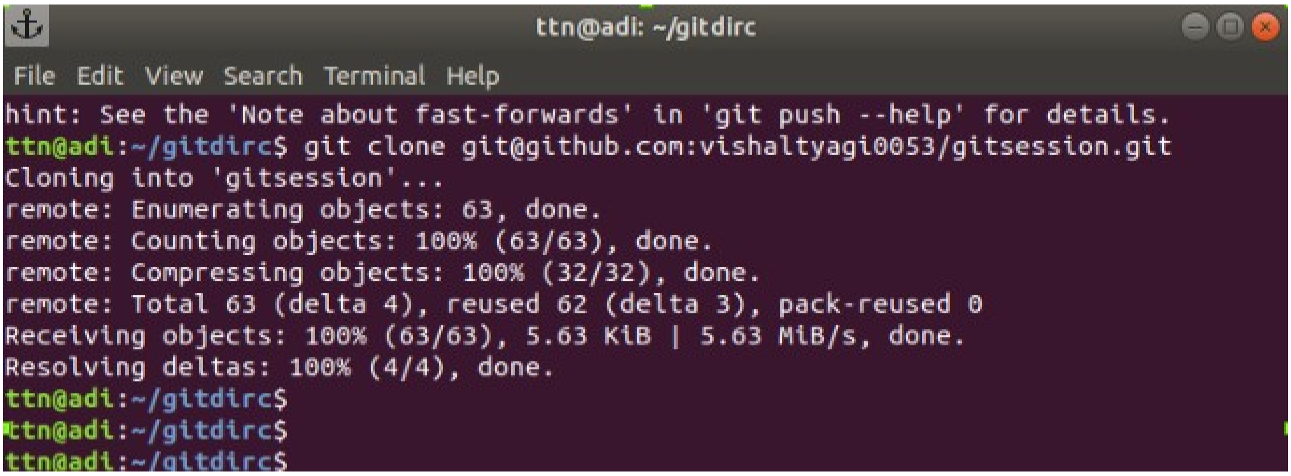
```
ttn@adi:~/newgit$ cat newfile1.txt
hello
this
is
aditya.
ttn@adi:~/newgit$ vim newfile1.txt
ttn@adi:~/newgit$ cat newfile1.txt
hello
this
is
aditya

the
new
changes.
ttn@adi:~/newgit$ 
```

8.Push changes to Github



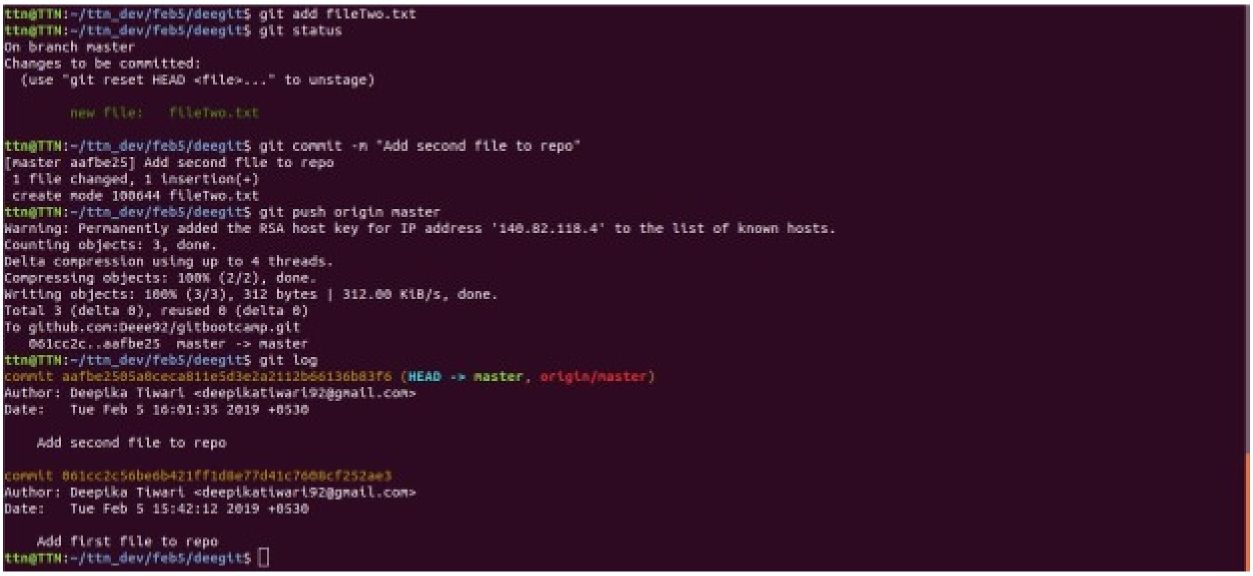
```
ttn@adi:~/gitdirc$ git push origin master
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 789 bytes | 394.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:vishaltyagi0053/gitsession.git
   d9732bf..db40395  master -> master
ttn@adi:~/gitdirc$ 
```

9.Clone the repository

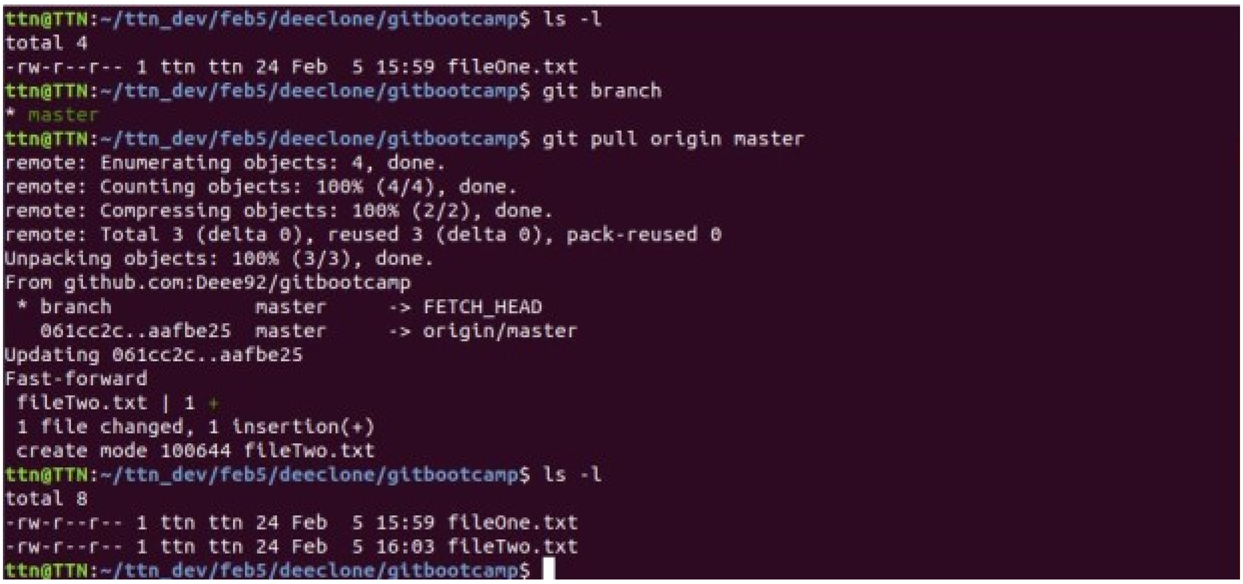


```
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
ttn@adi:~/gitdirc$ git clone git@github.com:vishaltyagi0053/gitsession.git
Cloning into 'gitsession'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 63 (delta 4), reused 62 (delta 3), pack-reused 0
Receiving objects: 100% (63/63), 5.63 KiB | 5.63 MiB/s, done.
Resolving deltas: 100% (4/4), done.
ttn@adi:~/gitdirc$
ttn@adi:~/gitdirc$
ttn@adi:~/gitdirc$
```

10.Add changes to one of the copies and pull the changes in the other.

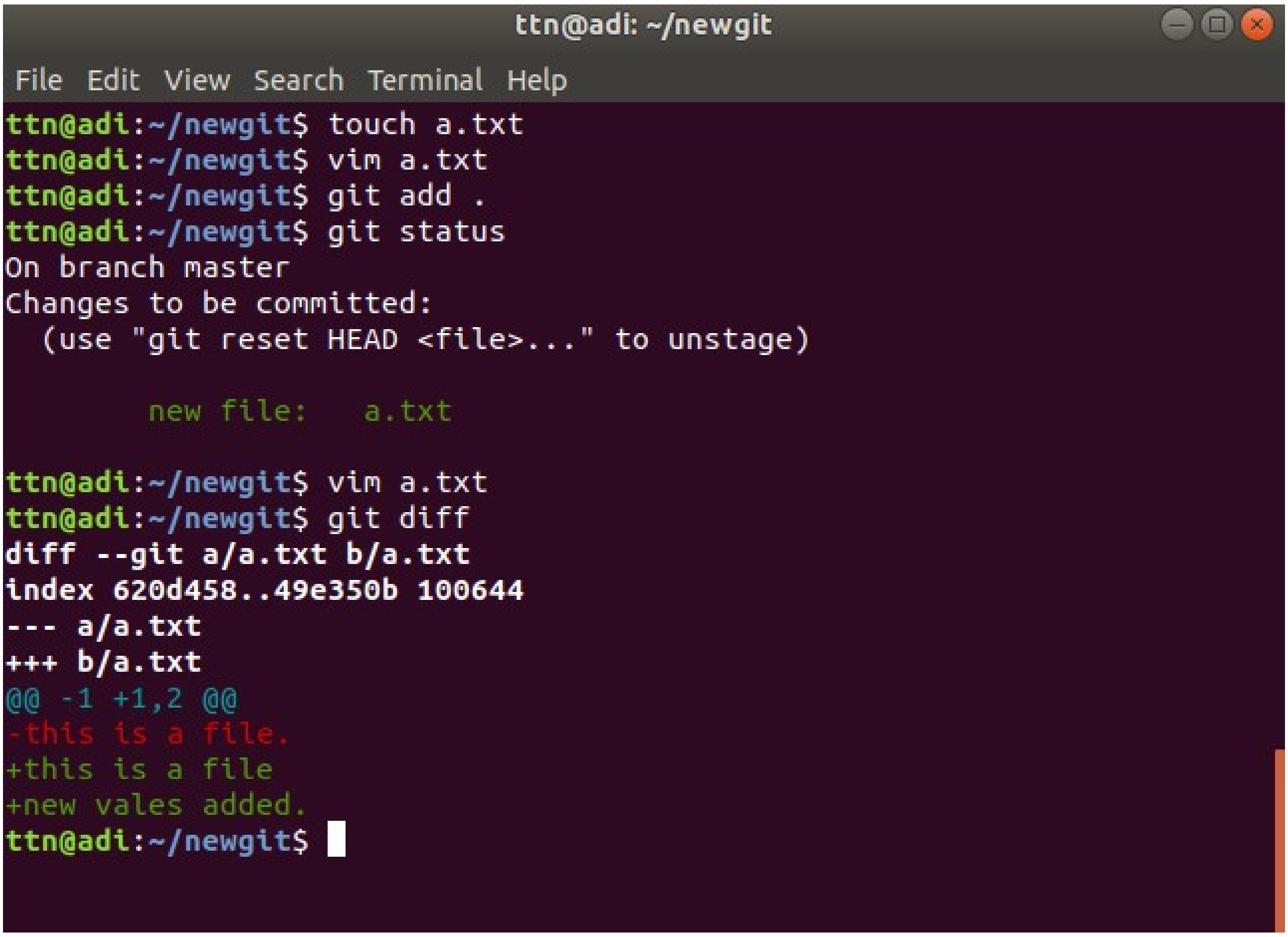


```
ttn@TTN:~/ttn_dev/feb5/deegit$ git add fileTwo.txt
ttn@TTN:~/ttn_dev/feb5/deegit$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   fileTwo.txt

ttn@TTN:~/ttn_dev/feb5/deegit$ git commit -m "Add second file to repo"
[master aafbe25] Add second file to repo
 1 file changed, 1 insertion(+)
 create mode 100644 fileTwo.txt
ttn@TTN:~/ttn_dev/feb5/deegit$ git push origin master
Warning: Permanently added the RSA host key for IP address '140.82.118.4' to the list of known hosts.
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:Deee92/gitbootcamp.git
   061cc2c..aafbe25  master -> master
ttn@TTN:~/ttn_dev/feb5/deegit$ git log
commit aafbe2505a8ceca811e5d3e2a2112b66136b83f6 (HEAD -> master, origin/master)
Author: Deepika Tiwari <deepikatiwari92@gmail.com>
Date:   Tue Feb 5 16:01:35 2019 +0530

    Add second file to repo

commit 061cc2c56be6b421ff1d8e77d41c7608cf252ae3
Author: Deepika Tiwari <deepikatiwari92@gmail.com>
Date:   Tue Feb 5 15:42:12 2019 +0530

    Add first file to repo
ttn@TTN:~/ttn_dev/feb5/deegit$ 
```



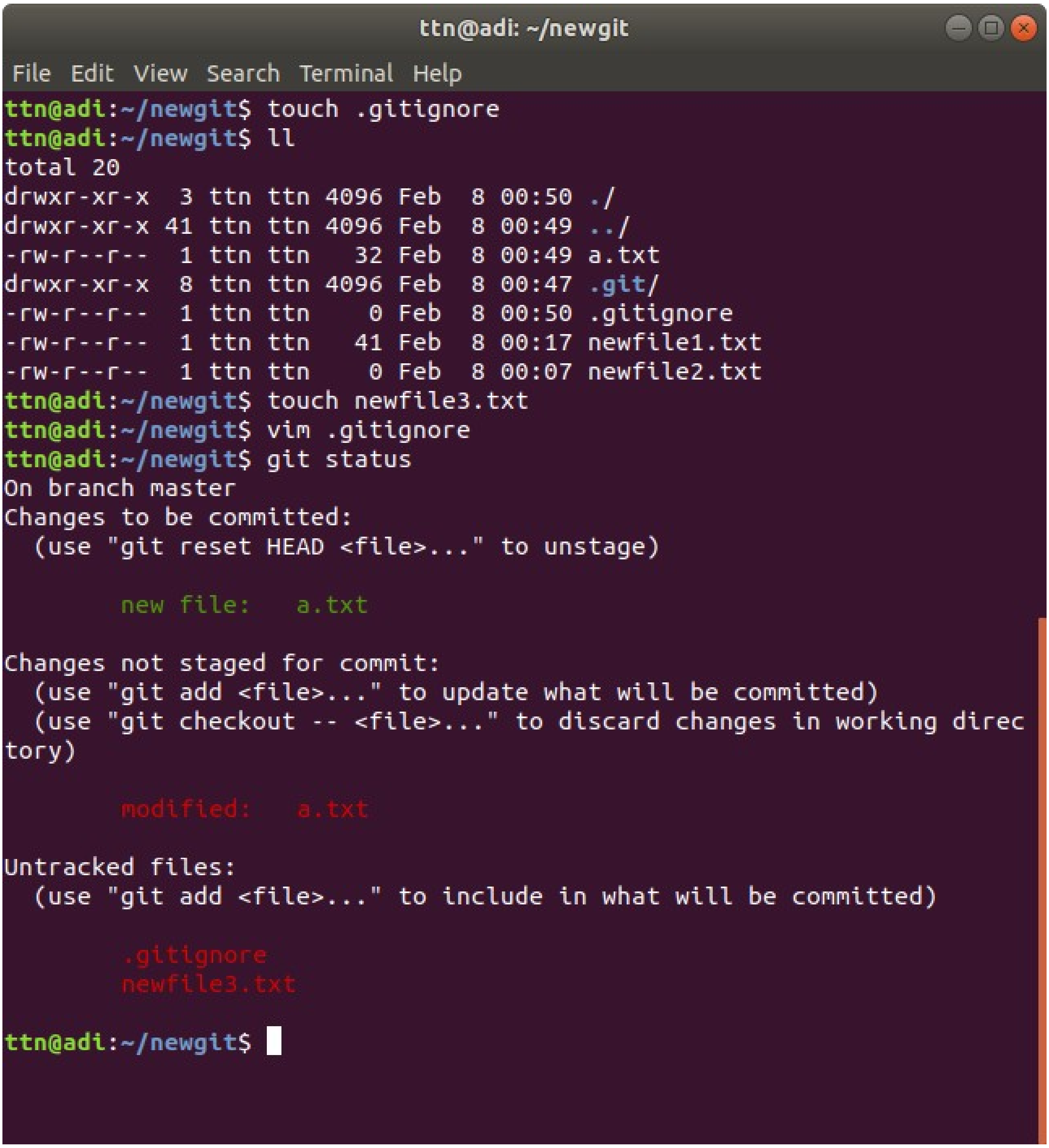
```
ttn@TTN:~/ttn_dev/feb5/deeclone/gitbootcamp$ ls -l
total 4
-rw-r--r-- 1 ttn ttn 24 Feb  5 15:59 fileOne.txt
ttn@TTN:~/ttn_dev/feb5/deeclone/gitbootcamp$ git branch
* master
ttn@TTN:~/ttn_dev/feb5/deeclone/gitbootcamp$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:Deee92/gitbootcamp
 * branch            master     -> FETCH_HEAD
   061cc2c..aafbe25  master     -> origin/master
Updating 061cc2c..aafbe25
Fast-forward
 fileTwo.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fileTwo.txt
ttn@TTN:~/ttn_dev/feb5/deeclone/gitbootcamp$ ls -l
total 8
-rw-r--r-- 1 ttn ttn 24 Feb  5 15:59 fileOne.txt
-rw-r--r-- 1 ttn ttn 24 Feb  5 16:03 fileTwo.txt
ttn@TTN:~/ttn_dev/feb5/deeclone/gitbootcamp$
```

11.Check differences between a file and its staged version



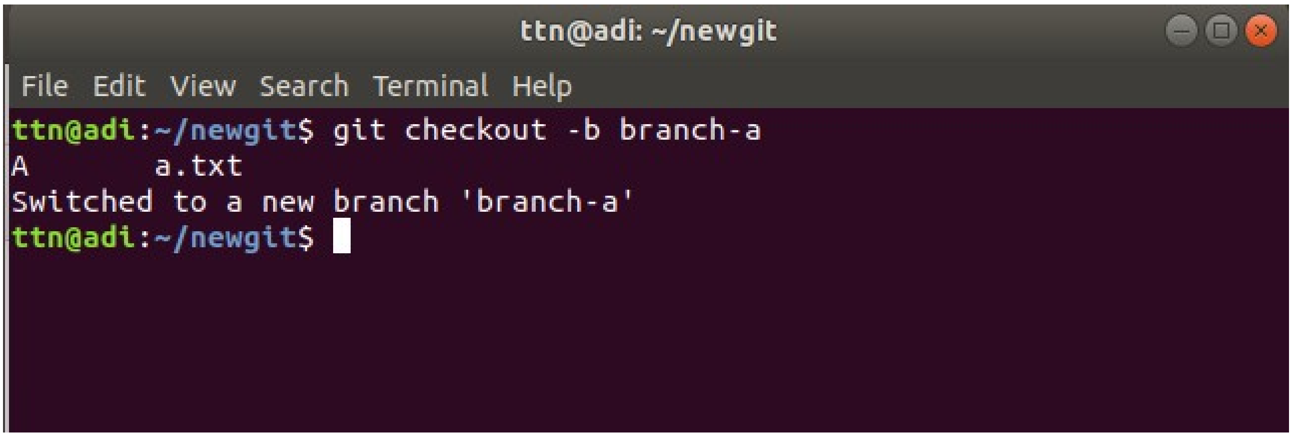
```
ttn@adi:~/newgit$ touch a.txt
ttn@adi:~/newgit$ vim a.txt
ttn@adi:~/newgit$ git add .
ttn@adi:~/newgit$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   a.txt

ttn@adi:~/newgit$ vim a.txt
ttn@adi:~/newgit$ git diff
diff --git a/a.txt b/a.txt
index 620d458..49e350b 100644
--- a/a.txt
+++ b/a.txt
@@ -1 +1,2 @@
-this is a file.
+this is a file
+new vales added.
ttn@adi:~/newgit$
```

12.Ignore a few files to be checked in



```
ttn@adi:~/newgit$ touch .gitignore
ttn@adi:~/newgit$ ll
total 20
drwxr-xr-x  3 ttn ttn 4096 Feb  8 00:50 ./
drwxr-xr-x 41 ttn ttn 4096 Feb  8 00:49 ../
-rw-r--r--  1 ttn ttn   32 Feb  8 00:49 a.txt
drwxr-xr-x  8 ttn ttn 4096 Feb  8 00:47 .git/
-rw-r--r--  1 ttn ttn    0 Feb  8 00:50 .gitignore
-rw-r--r--  1 ttn ttn   41 Feb  8 00:17 newfile1.txt
-rw-r--r--  1 ttn ttn    0 Feb  8 00:07 newfile2.txt
ttn@adi:~/newgit$ touch newfile3.txt
ttn@adi:~/newgit$ vim .gitignore
ttn@adi:~/newgit$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   a.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working direc
tory)

        modified:   a.txt

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

        .gitignore
        newfile3.txt

ttn@adi:~/newgit$
```

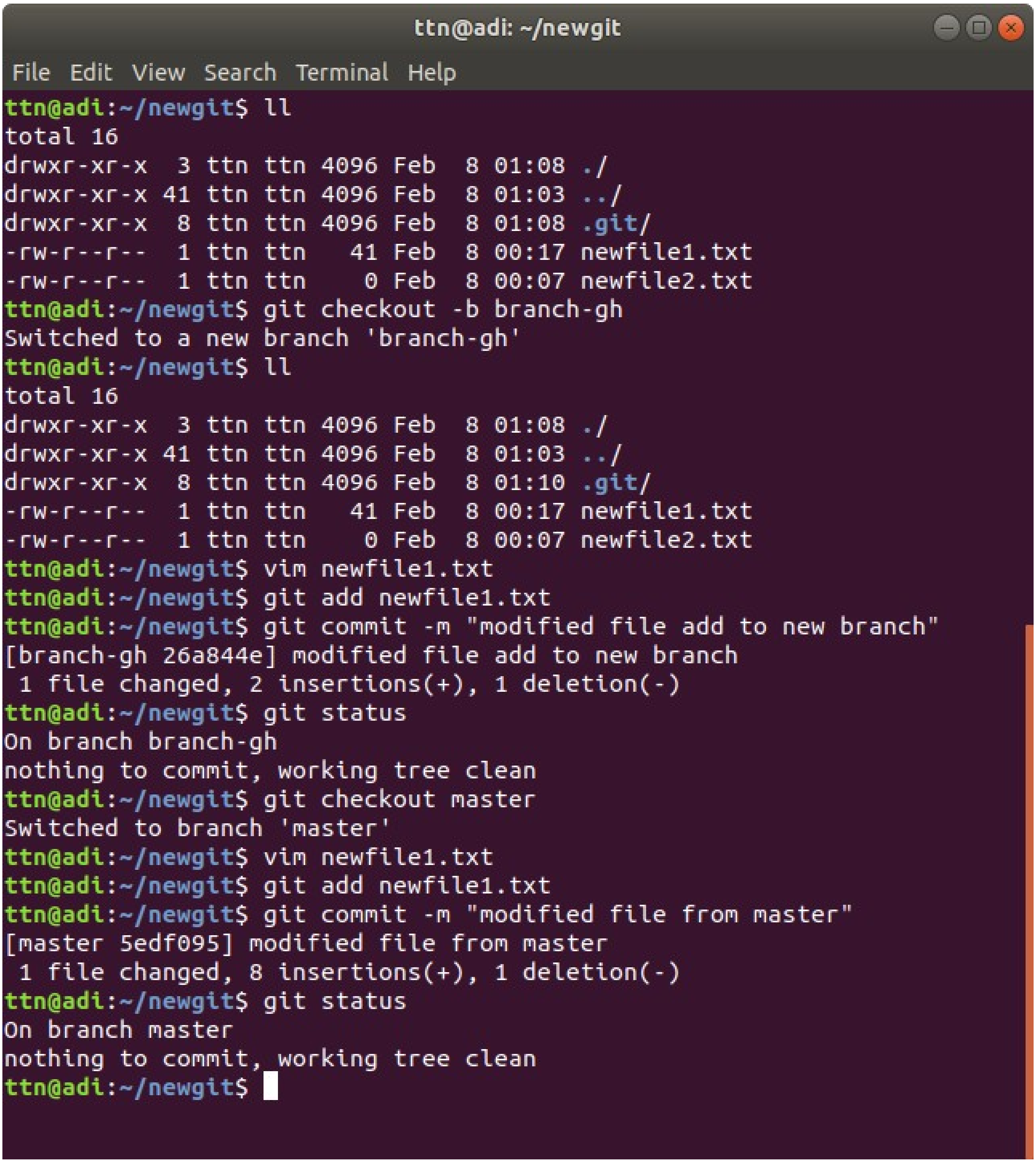
13.Create a new branch.



```
ttn@adi:~/newgit$ git checkout -b branch-a
A       a.txt
Switched to a new branch 'branch-a'
ttn@adi:~/newgit$ 
```

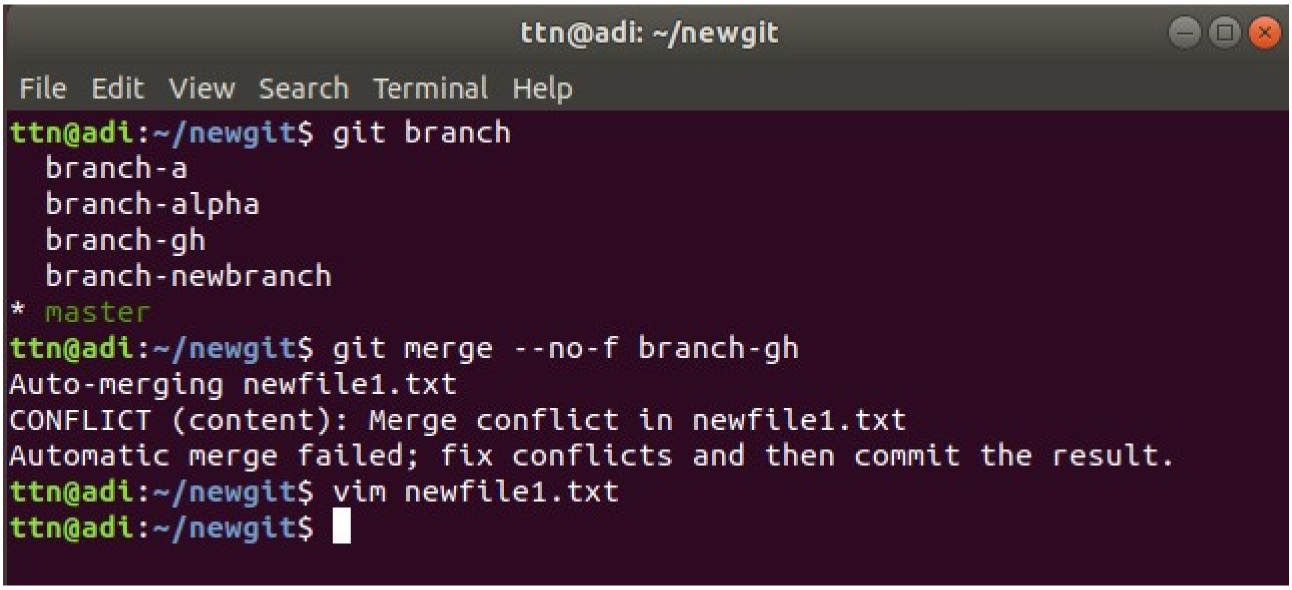
14Diverge them with commits

15.Edit the same file at the same line on both branches and commit

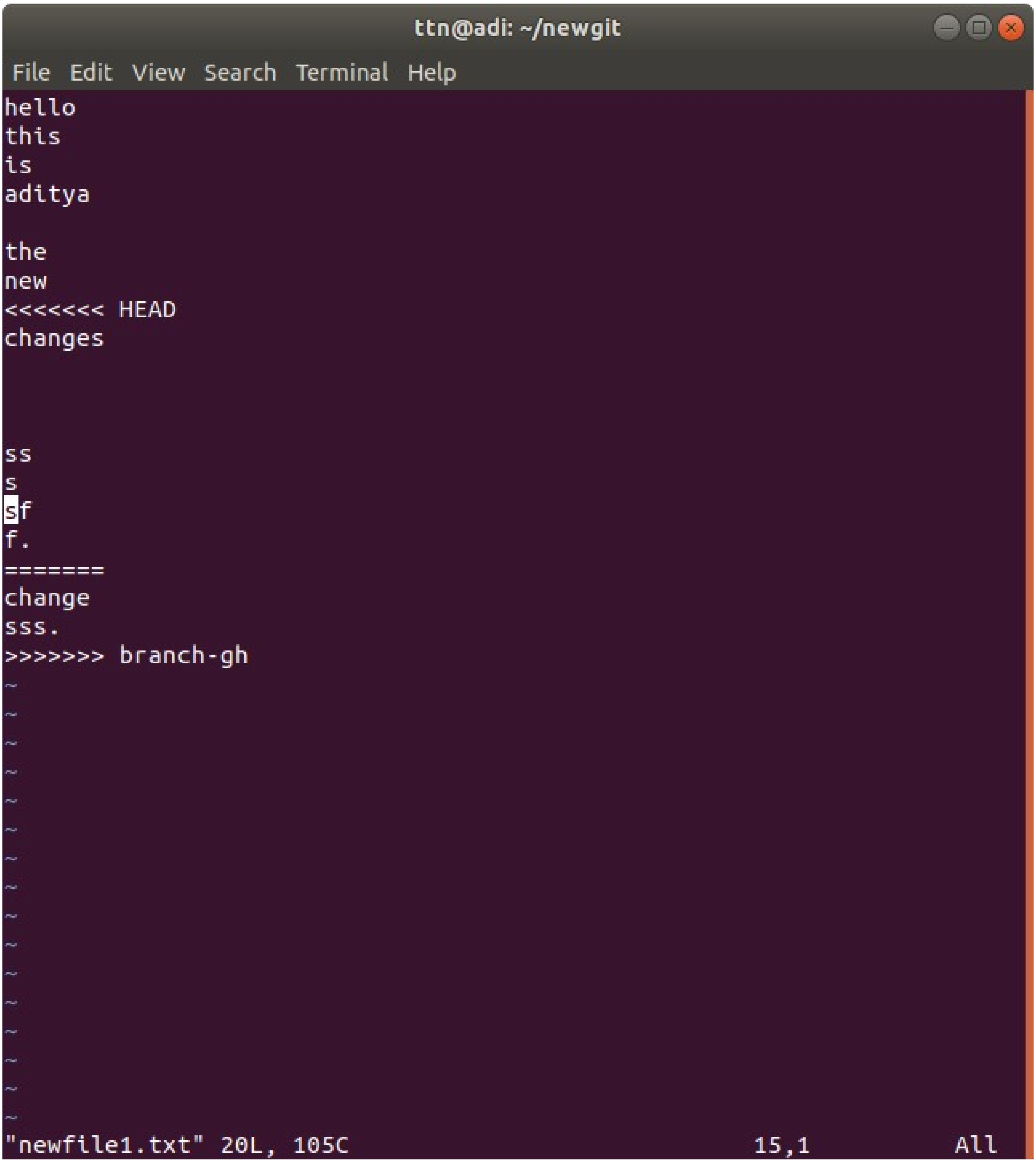


```
ttn@adi:~/newgit$ ll
total 16
drwxr-xr-x  3 ttn ttn 4096 Feb  8 01:08 ./
drwxr-xr-x 41 ttn ttn 4096 Feb  8 01:03 ../
drwxr-xr-x  8 ttn ttn 4096 Feb  8 01:08 .git/
-rw-r--r--  1 ttn ttn   41 Feb  8 00:17 newfile1.txt
-rw-r--r--  1 ttn ttn    0 Feb  8 00:07 newfile2.txt
ttn@adi:~/newgit$ git checkout -b branch-gh
Switched to a new branch 'branch-gh'
ttn@adi:~/newgit$ ll
total 16
drwxr-xr-x  3 ttn ttn 4096 Feb  8 01:08 ./
drwxr-xr-x 41 ttn ttn 4096 Feb  8 01:03 ../
drwxr-xr-x  8 ttn ttn 4096 Feb  8 01:10 .git/
-rw-r--r--  1 ttn ttn   41 Feb  8 00:17 newfile1.txt
-rw-r--r--  1 ttn ttn    0 Feb  8 00:07 newfile2.txt
ttn@adi:~/newgit$ vim newfile1.txt
ttn@adi:~/newgit$ git add newfile1.txt
ttn@adi:~/newgit$ git commit -m "modified file add to new branch"
[branch-gh 26a844e] modified file add to new branch
 1 file changed, 2 insertions(+), 1 deletion(-)
ttn@adi:~/newgit$ git status
On branch branch-gh
nothing to commit, working tree clean
ttn@adi:~/newgit$ git checkout master
Switched to branch 'master'
ttn@adi:~/newgit$ vim newfile1.txt
ttn@adi:~/newgit$ git add newfile1.txt
ttn@adi:~/newgit$ git commit -m "modified file from master"
[master 5edf095] modified file from master
 1 file changed, 8 insertions(+), 1 deletion(-)
ttn@adi:~/newgit$ git status
On branch master
nothing to commit, working tree clean
ttn@adi:~/newgit$ 
```

16.Try merging and resolve merge conflicts

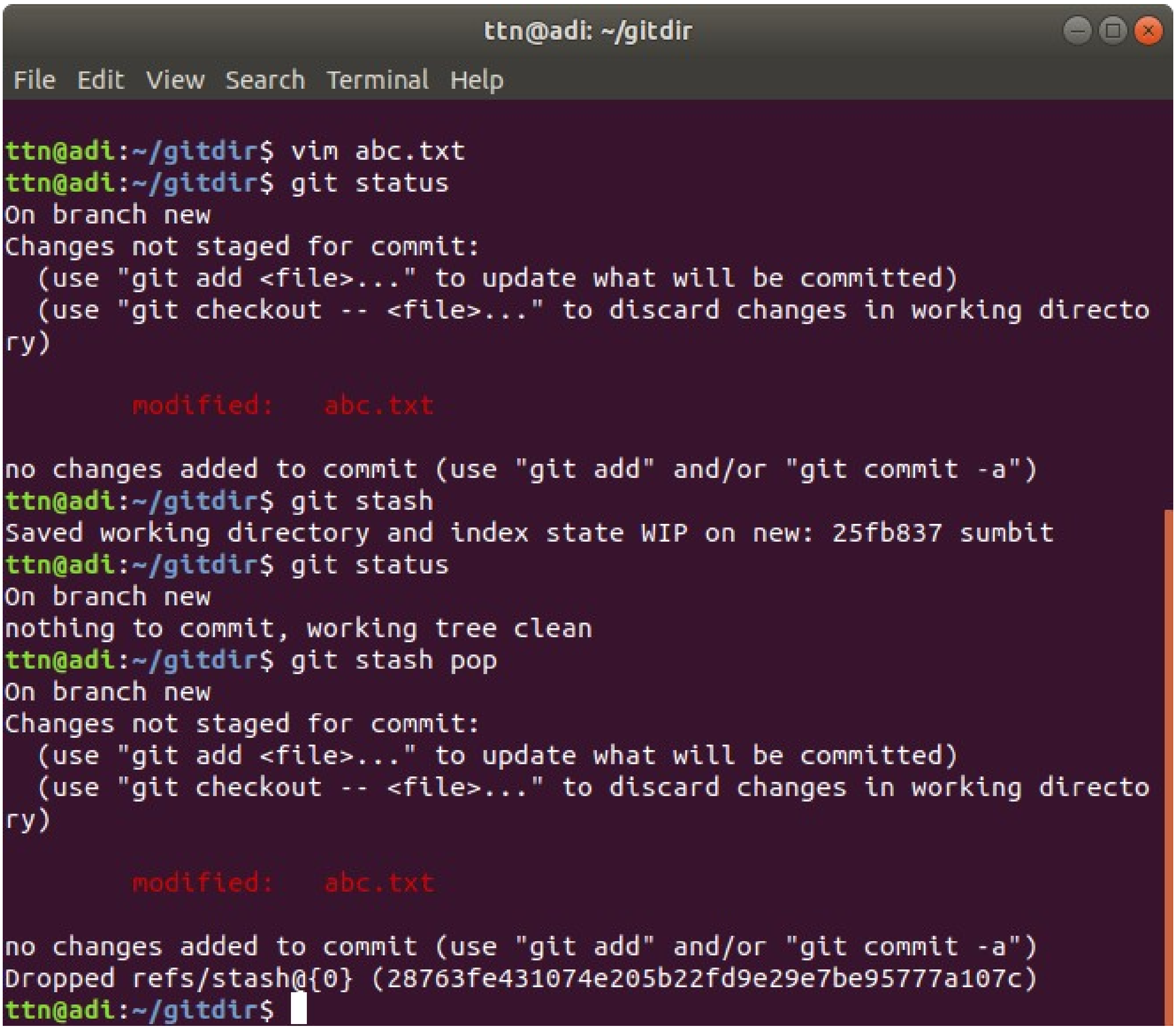


```
ttn@adi:~/newgit$ git branch
  branch-a
  branch-alpha
  branch-gh
  branch-newbranch
* master
ttn@adi:~/newgit$ git merge --no-f branch-gh
Auto-merging newfile1.txt
CONFLICT (content): Merge conflict in newfile1.txt
Automatic merge failed; fix conflicts and then commit the result.
ttn@adi:~/newgit$ vim newfile1.txt
ttn@adi:~/newgit$ 
```



```
hello
this
is
aditya

the
new
<<<<<<< HEAD
changes



ss
s
sf
f.
=======
change
sss.
>>>>>>> branch-gh
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"newfile1.txt" 20L, 105C                          15,1          All
```

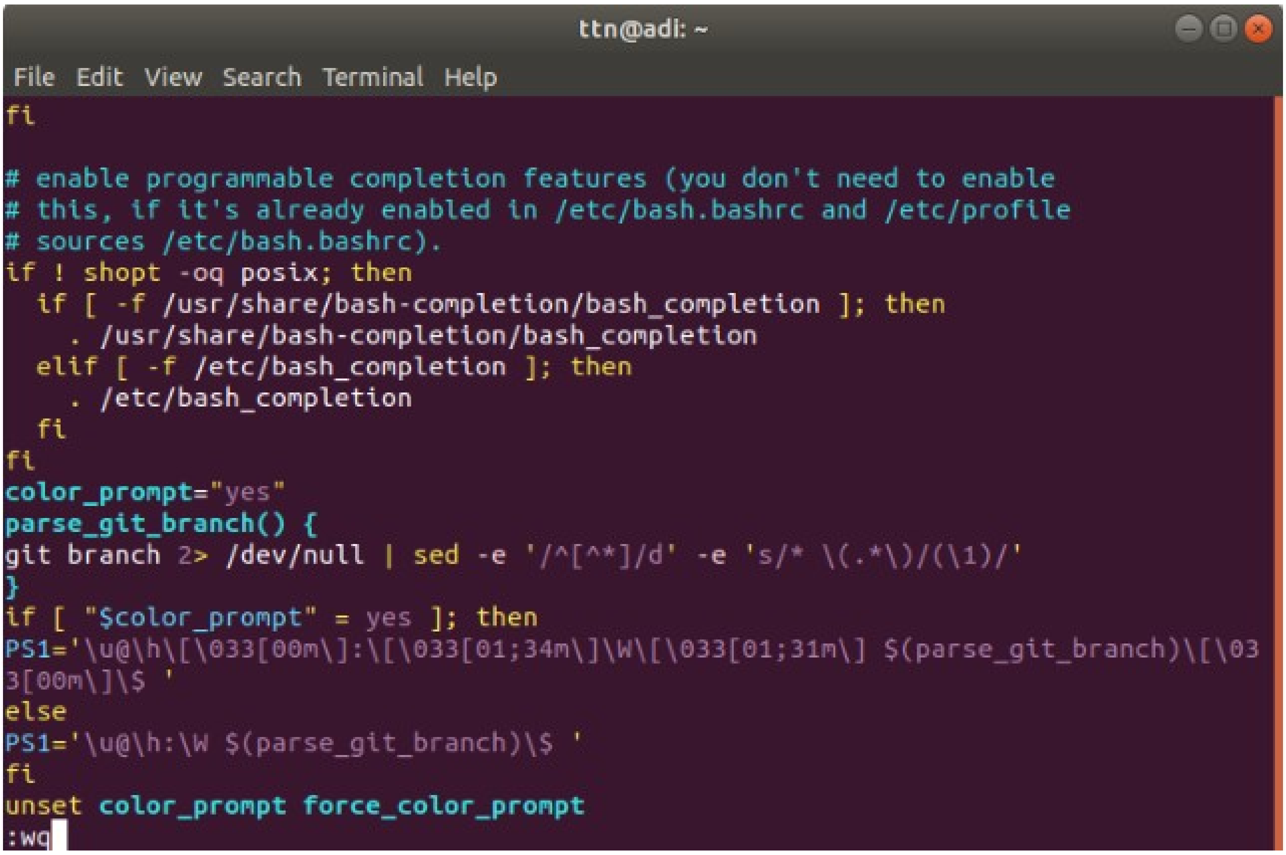
17. Stash the changes and pop them



```
ttn@adi:~/gitdir$ vim abc.txt
ttn@adi:~/gitdir$ git status
On branch new
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directo
ry)

        modified:   abc.txt

no changes added to commit (use "git add" and/or "git commit -a")
ttn@adi:~/gitdir$ git stash
Saved working directory and index state WIP on new: 25fb837 sumbit
ttn@adi:~/gitdir$ git status
On branch new
nothing to commit, working tree clean
ttn@adi:~/gitdir$ git stash pop
On branch new
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directo
ry)

        modified:   abc.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (28763fe431074e205b22fd9e29e7be95777a107c)
ttn@adi:~/gitdir$
```

18. Add the following code to your .bashrc file : color_prompt="yes"

```
parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h\[\033[00m\]:\[\033[
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
```



```
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
color_prompt="yes"
parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\03
3[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
:wq
```