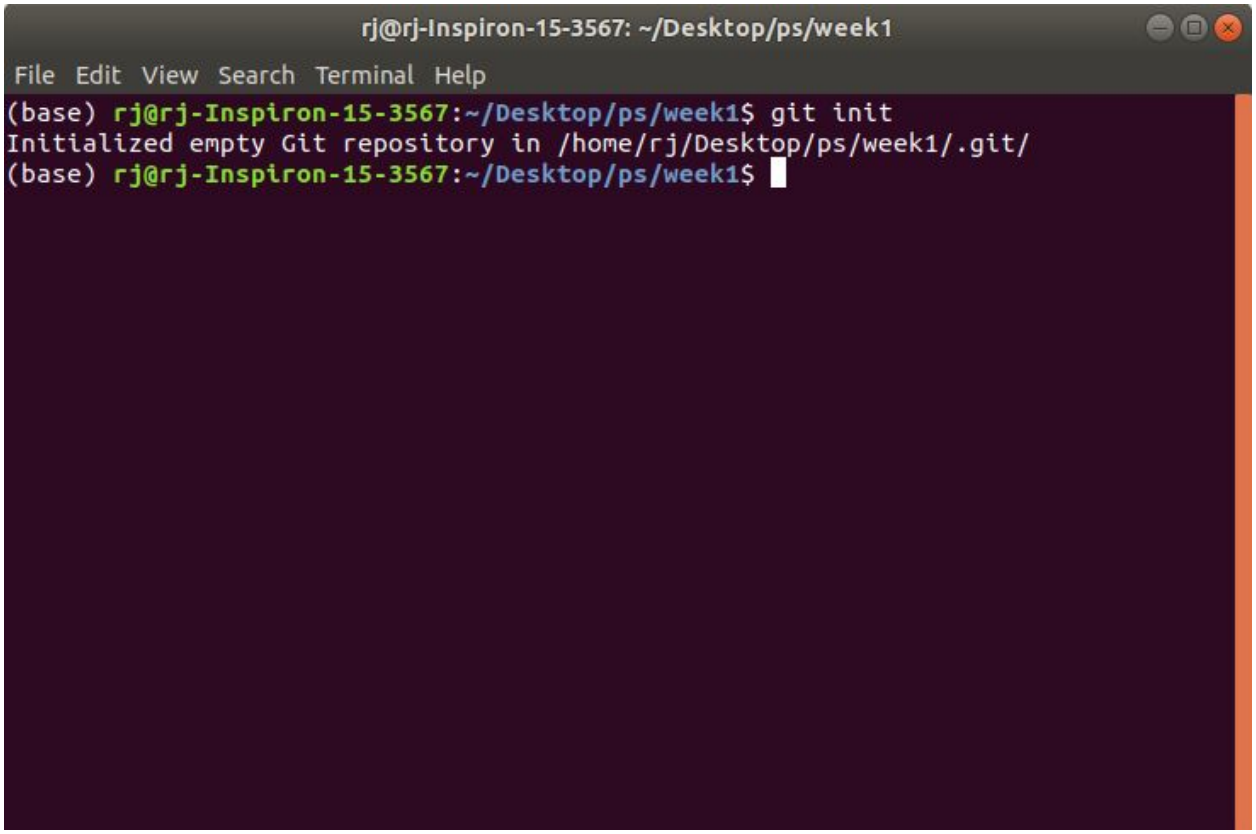
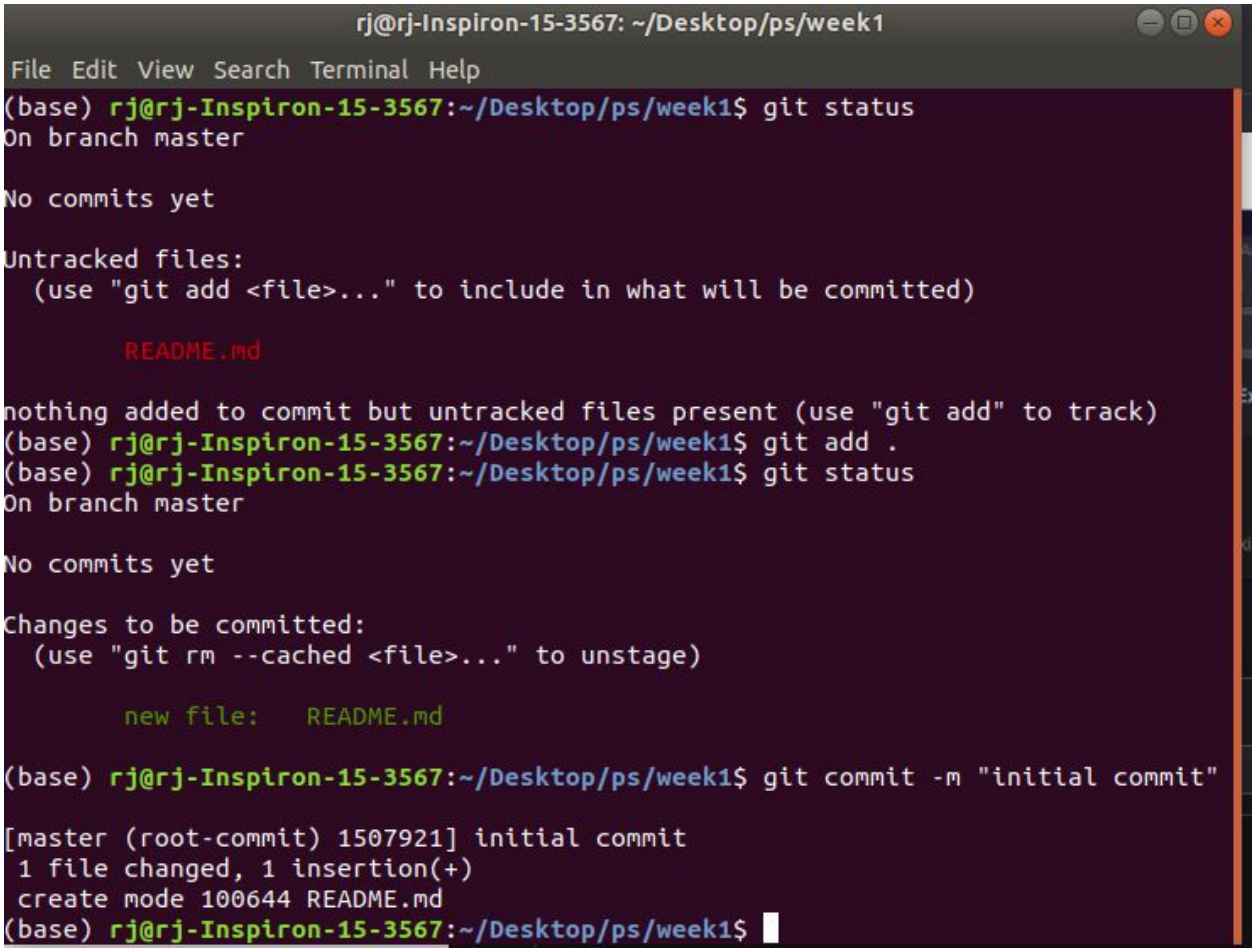
Rohit Kumar Singh (rohitjadaun3@gmail.com)

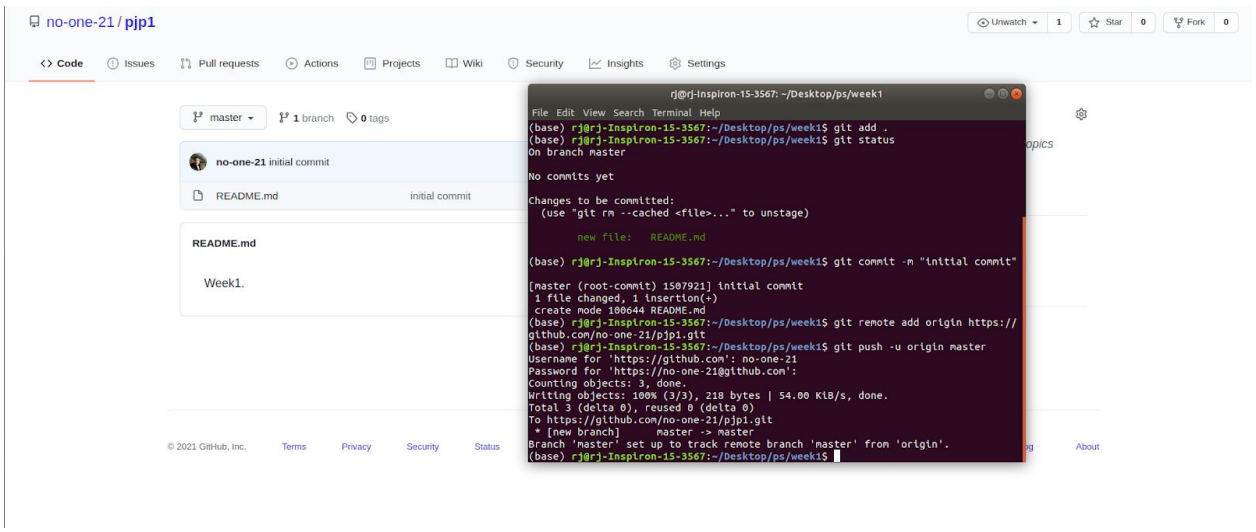
1. **Basic usage of cli**

a. **Setup a Local Repository**

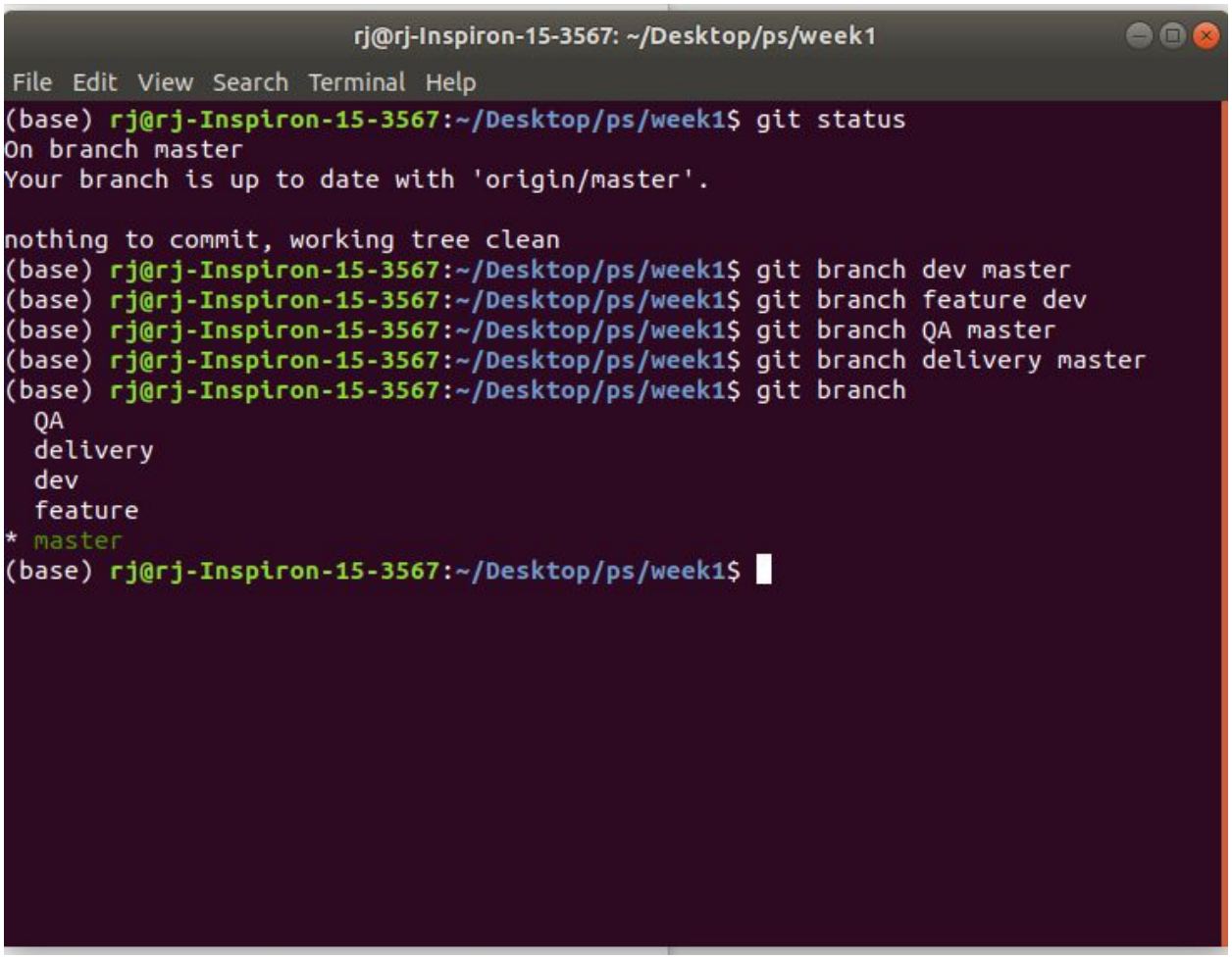


b. **Setup a remote repository**

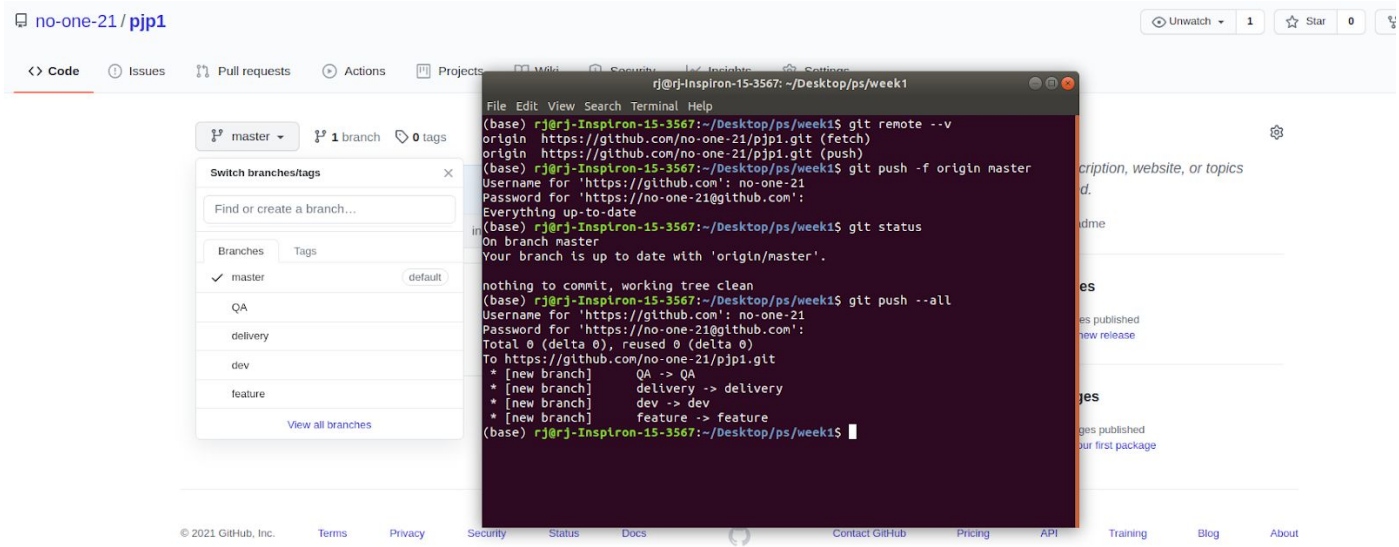




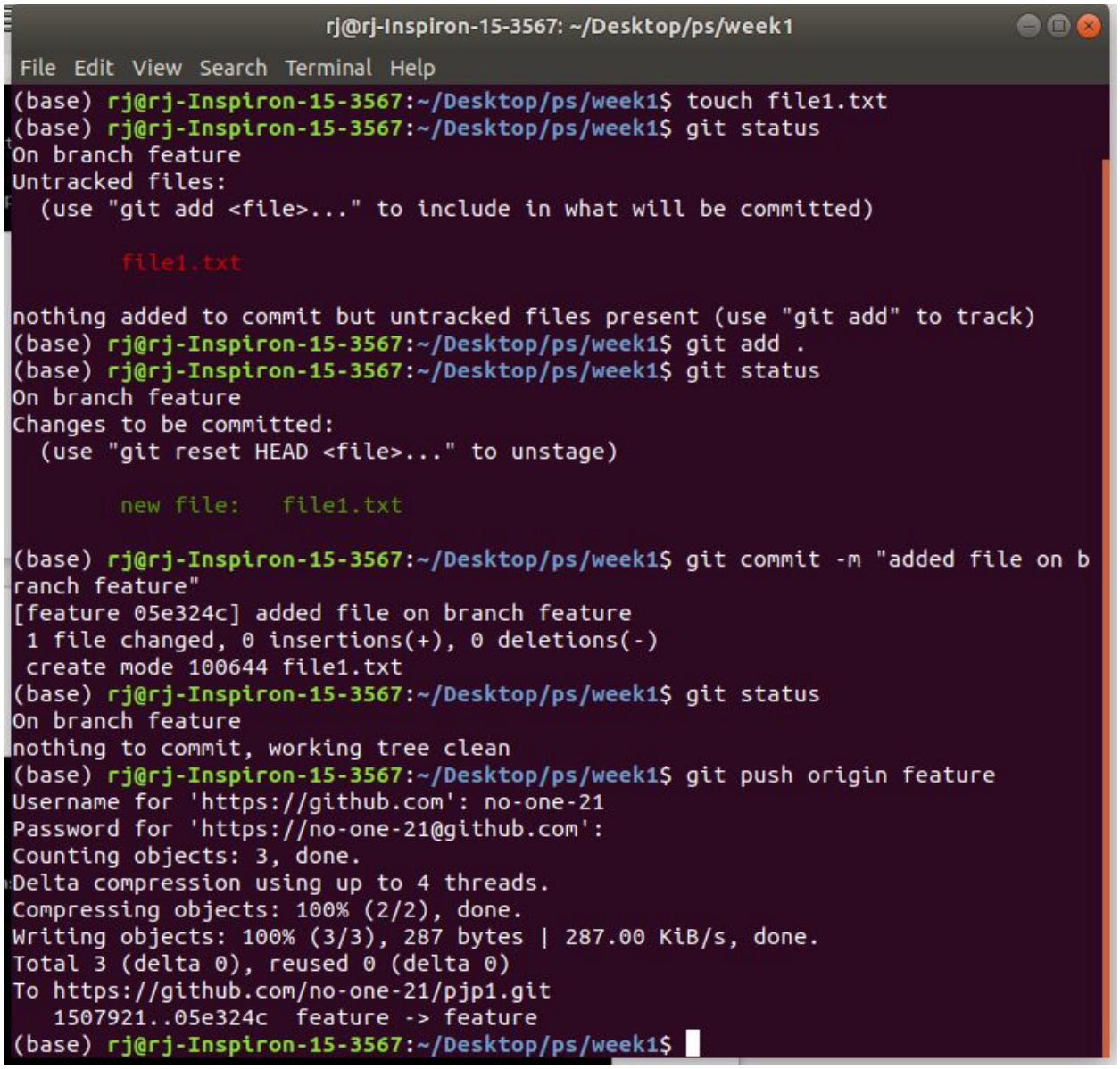
C. Create local branches



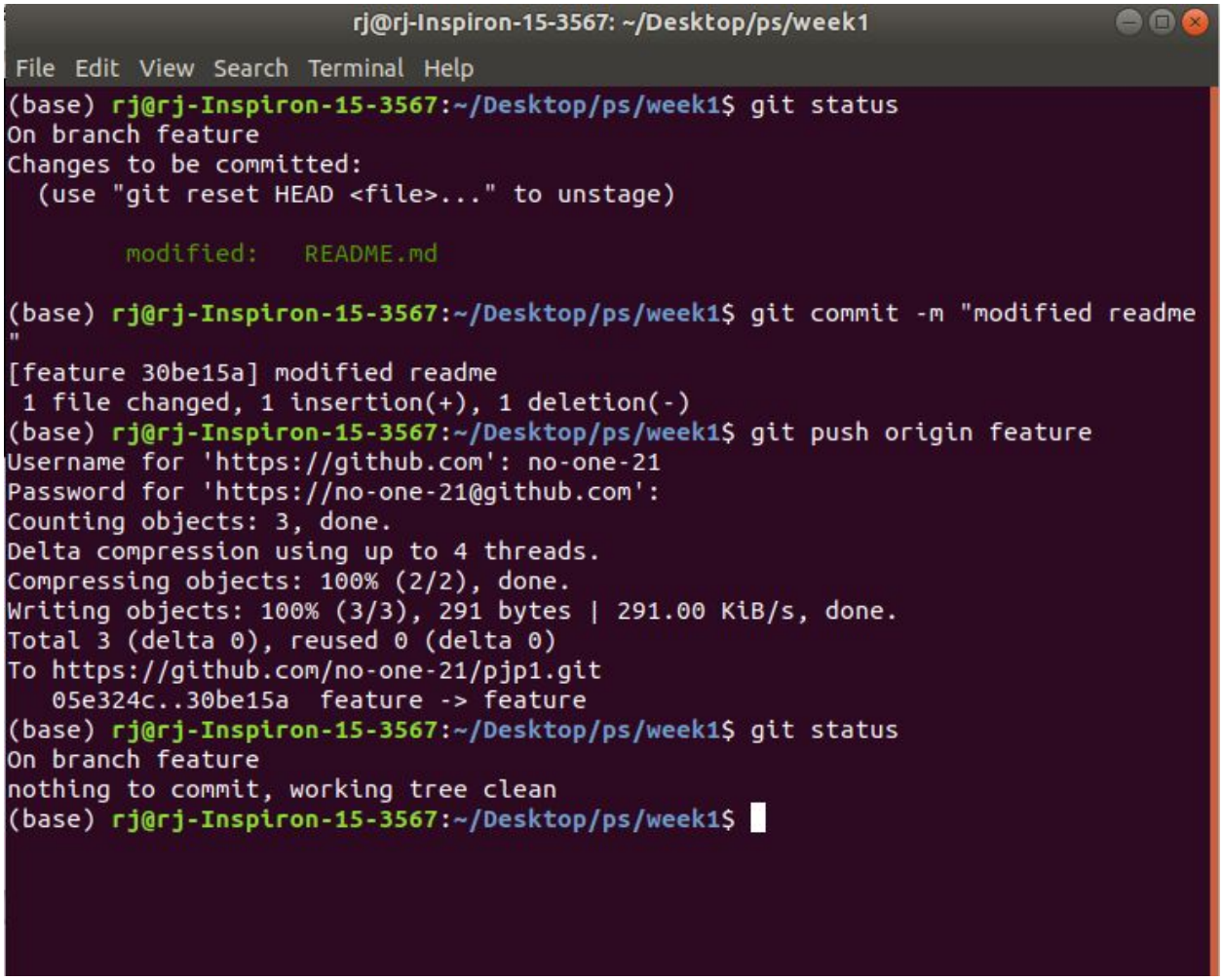
D. Create remote branches



E. Add files/folders, make changes to files, remove files/folders



F. Check in, stage, commit, push changes into feature branch



G. Promote code from feature branch to dev branch via pull requests.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: dev ← compare: feature ✓ **Able to merge.** These branches can be automatically merged.

modified files

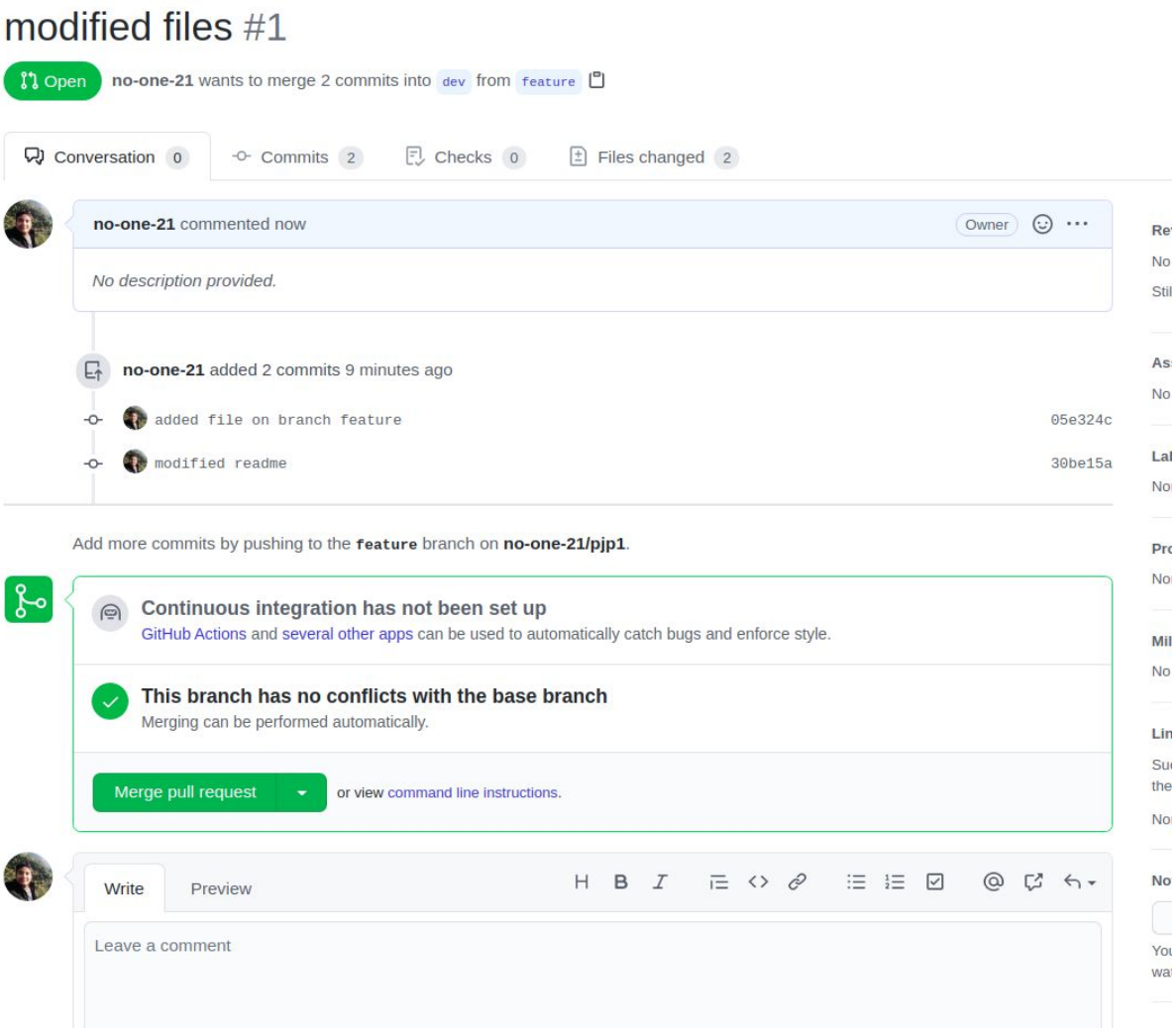
Write Preview

Leave a comment

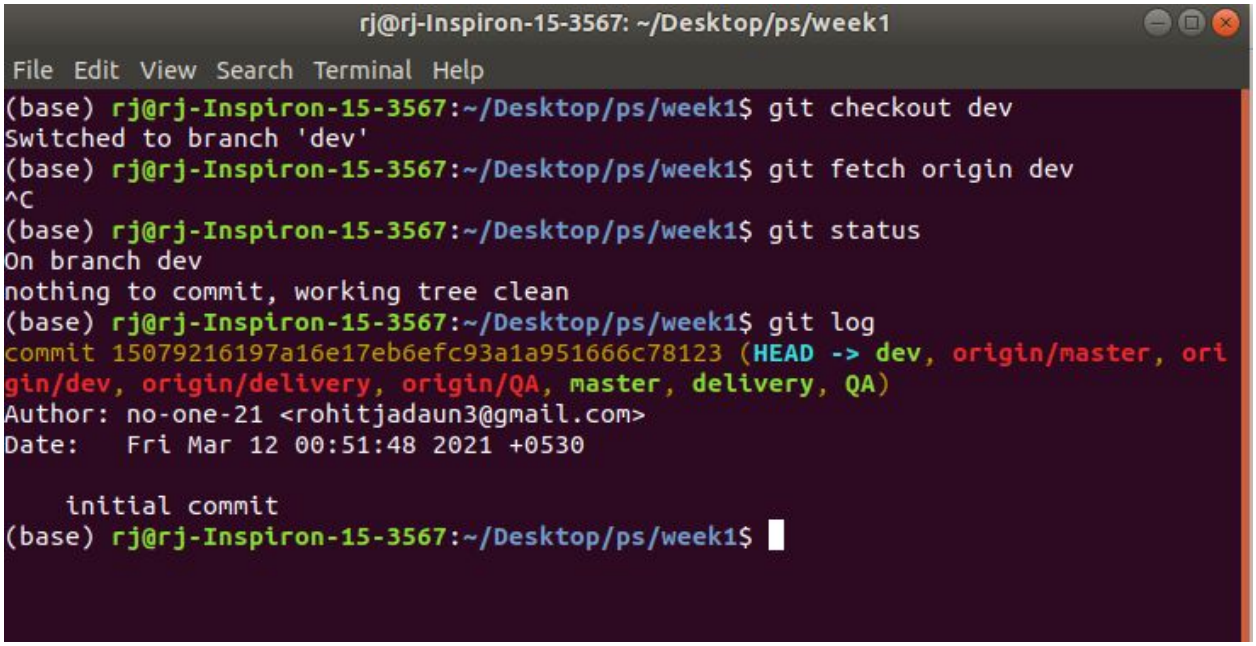
Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.

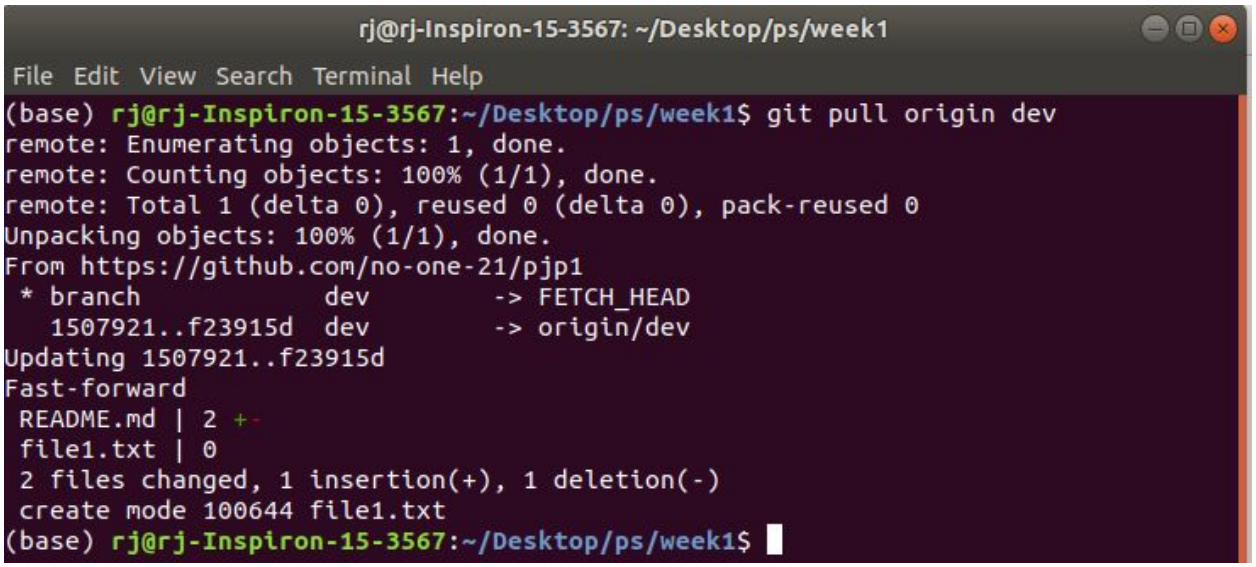


I. Checkout latest code from remote to local



H. Explore difference between checkout and pull

Ans.

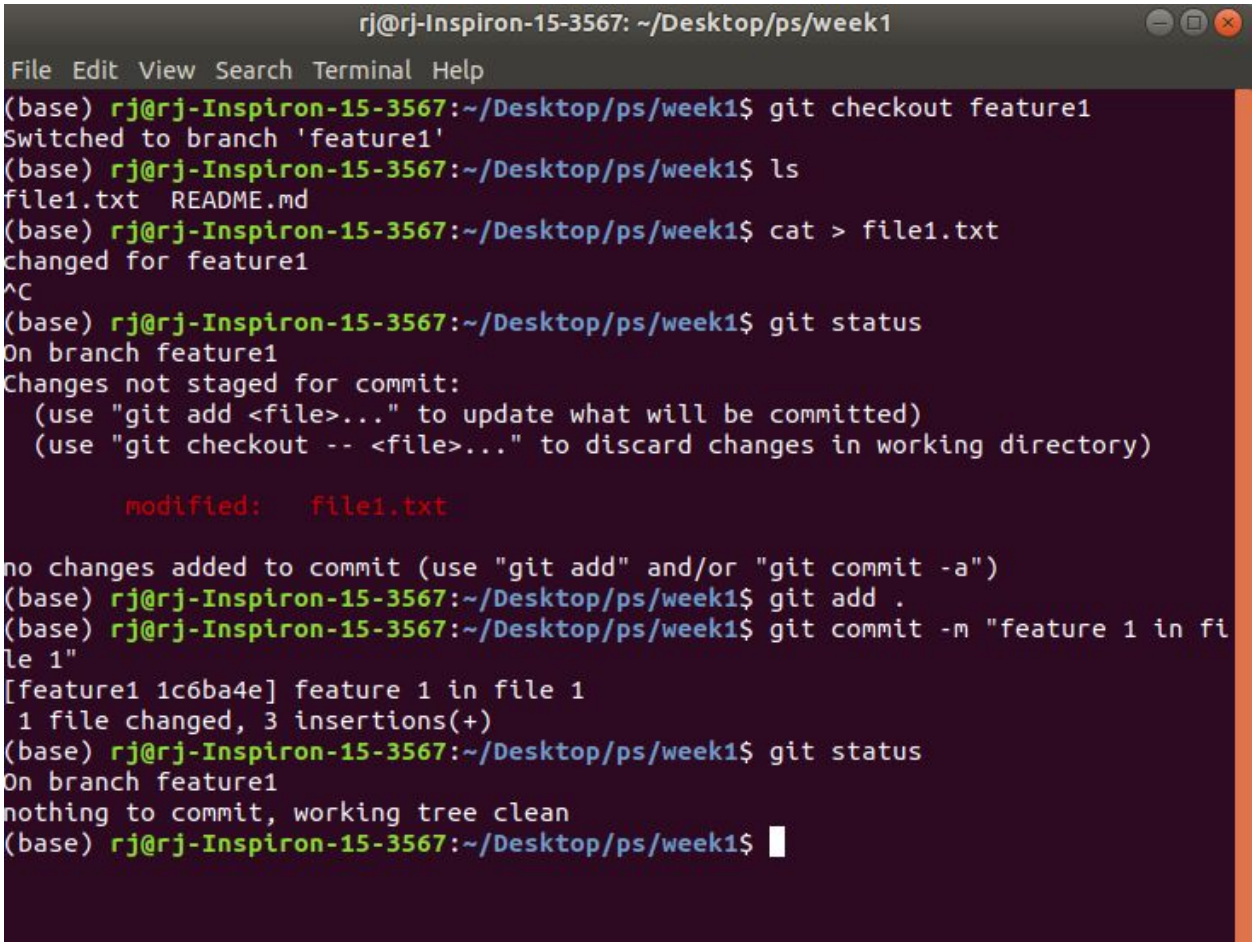


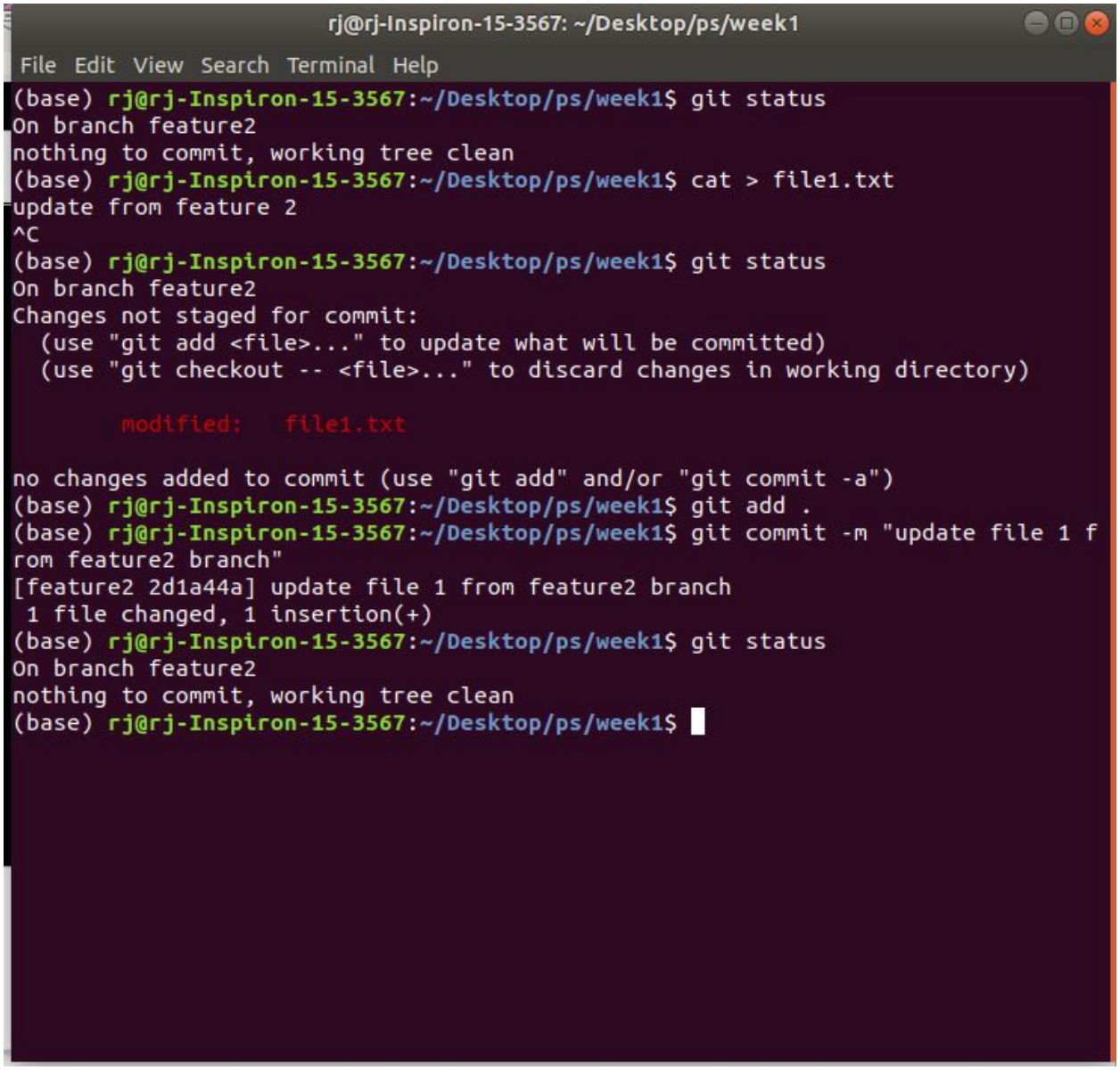
checkout-> Fetches the latest changes. You should already have this repo downloaded. It **does not merge** those new changes but makes your working directory reflect them. You can merge them at your leisure later.

pull-> fetches the latest changes from remote and merges them into the local branch of the same name.

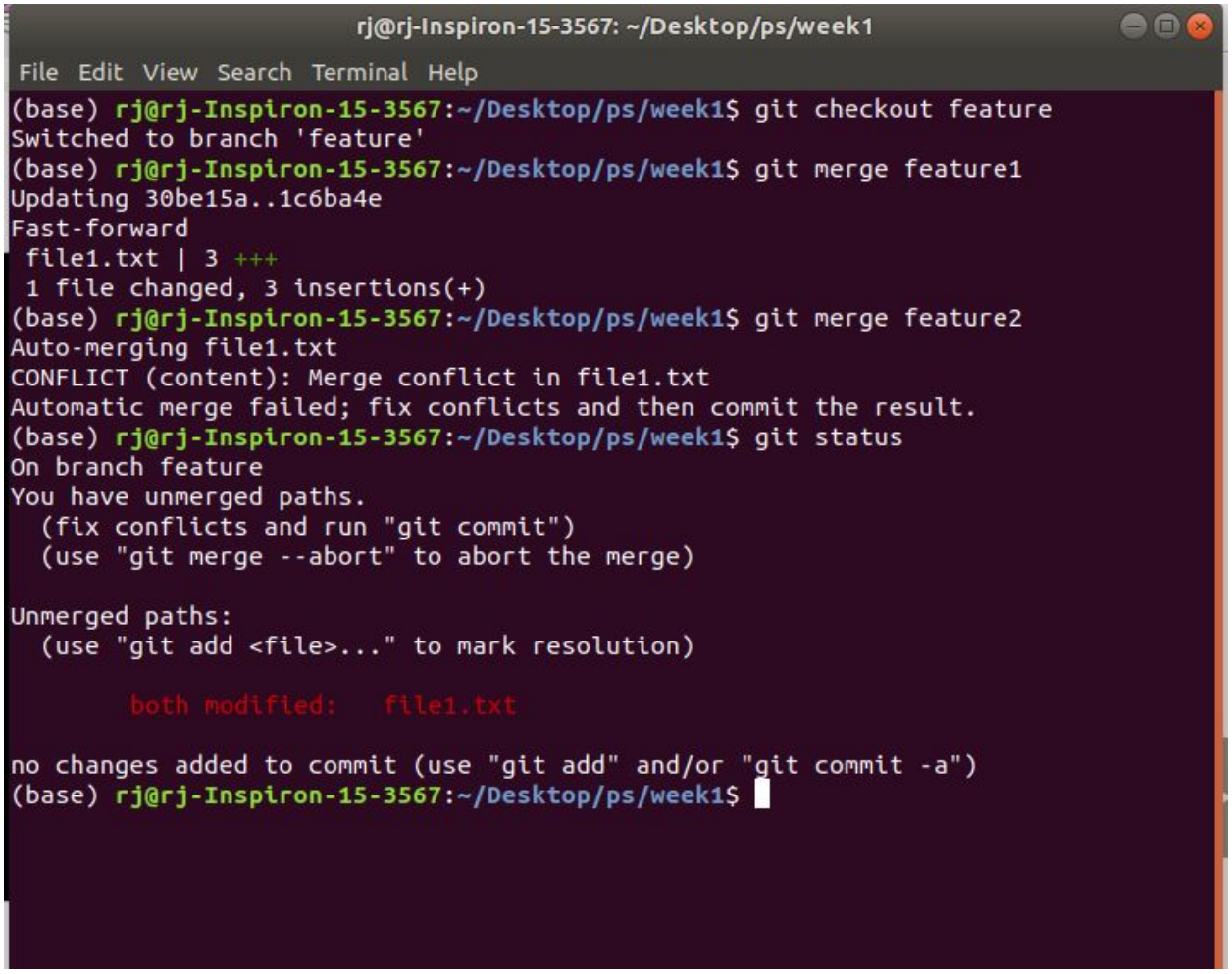
I. Get two people make changes to same file, checkin & handle merge conflicts

a. Let's first create 2 new branches and make changes to the test.txt file.

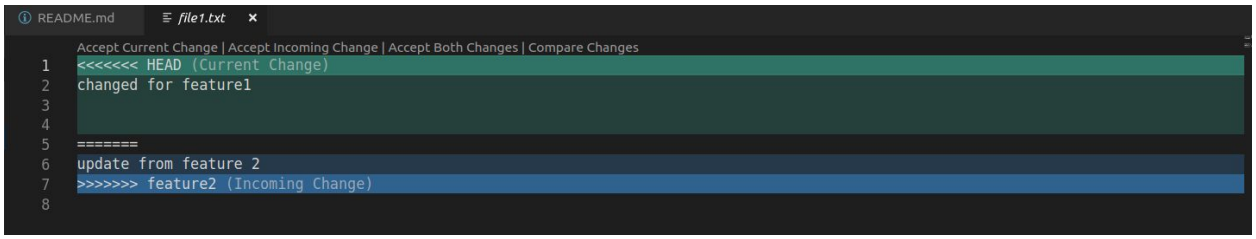

rj@rj-Inspiron-15-3567: ~/Desktop/ps/week1
File Edit View Search Terminal Help
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git checkout feature1
Switched to branch 'feature1'
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ ls
file1.txt README.md
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ cat > file1.txt
changed for feature1
^C
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git status
On branch feature1
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: file1.txt
no changes added to commit (use "git add" and/or "git commit -a")
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git add .
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git commit -m "feature 1 in fi
le 1"
[feature1 1c6ba4e] feature 1 in file 1
 1 file changed, 3 insertions(+)
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git status
On branch feature1
nothing to commit, working tree clean
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$

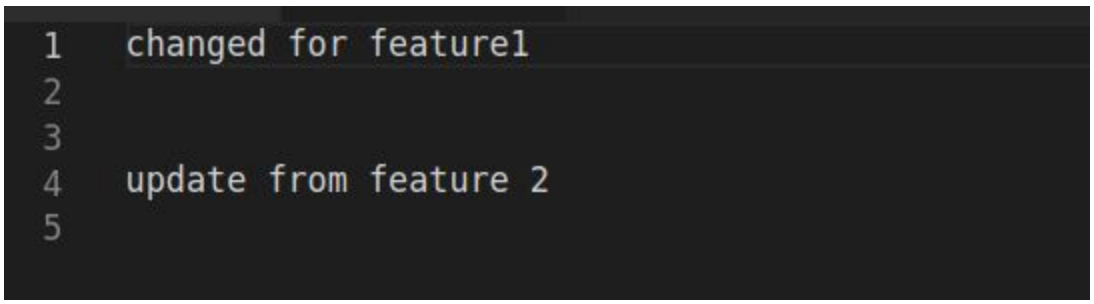


Now let’s try to merge changes from both the branches to feature branch

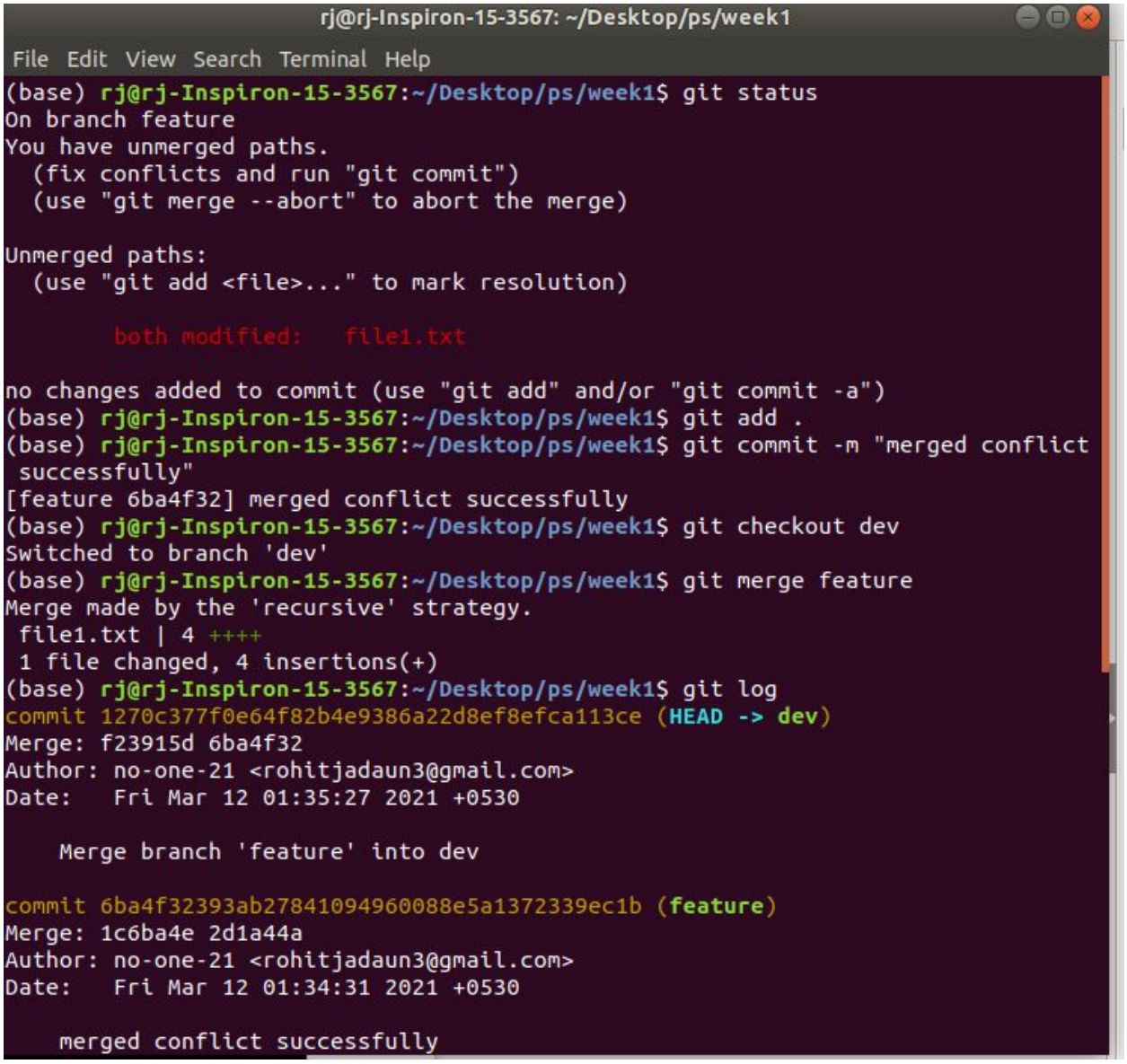


Now try to fix changes manually

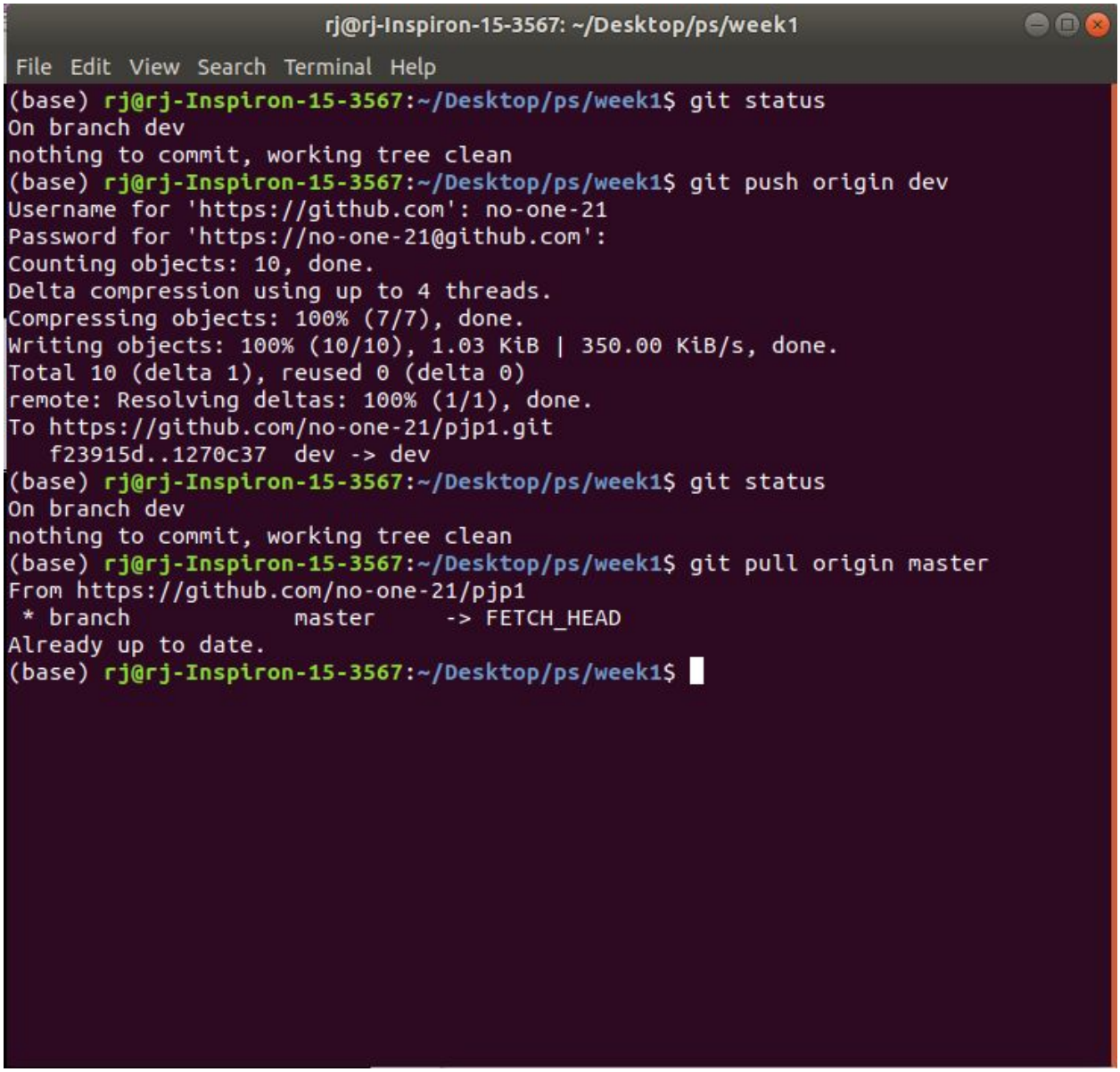




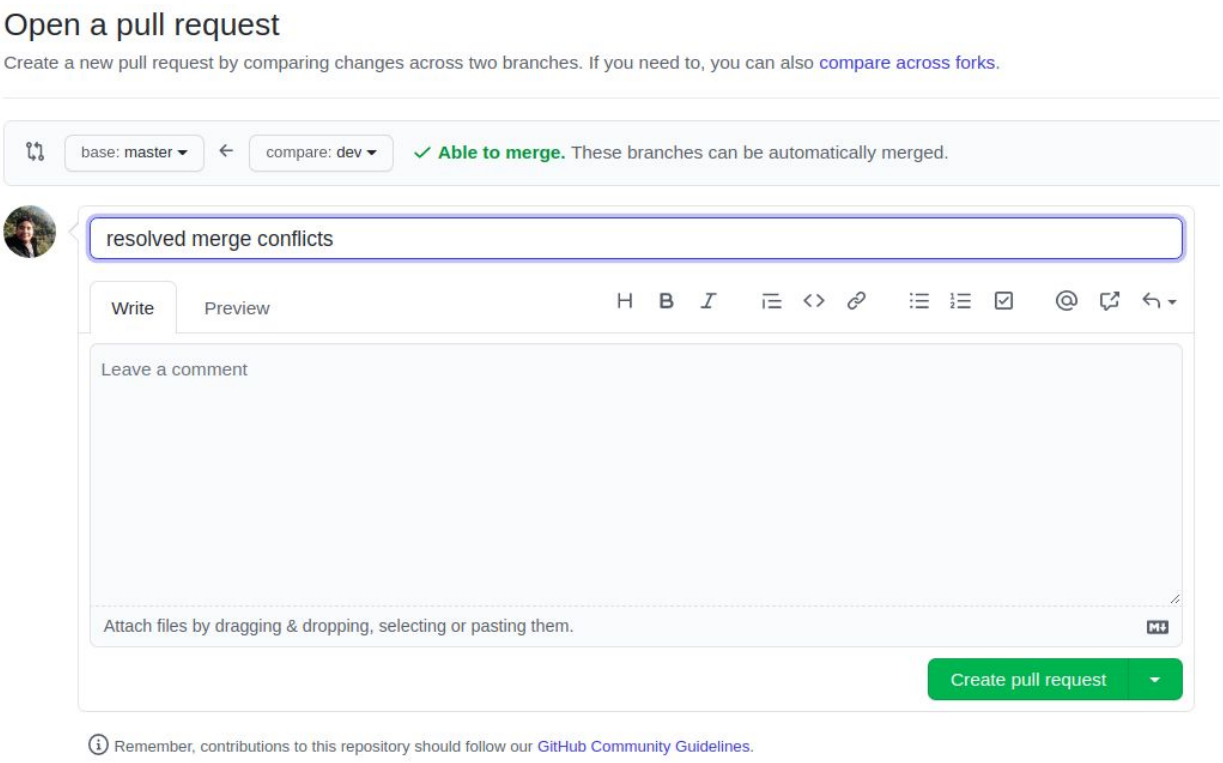
Now commit changes and merge them

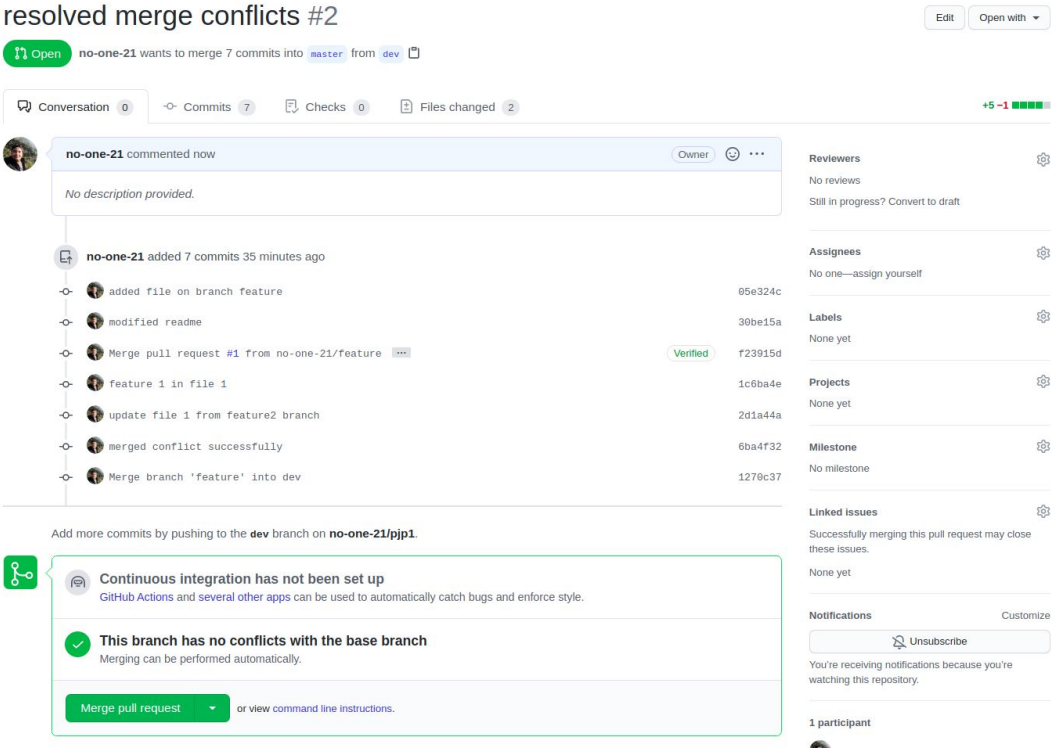


J. Ensure code is sync across all branches

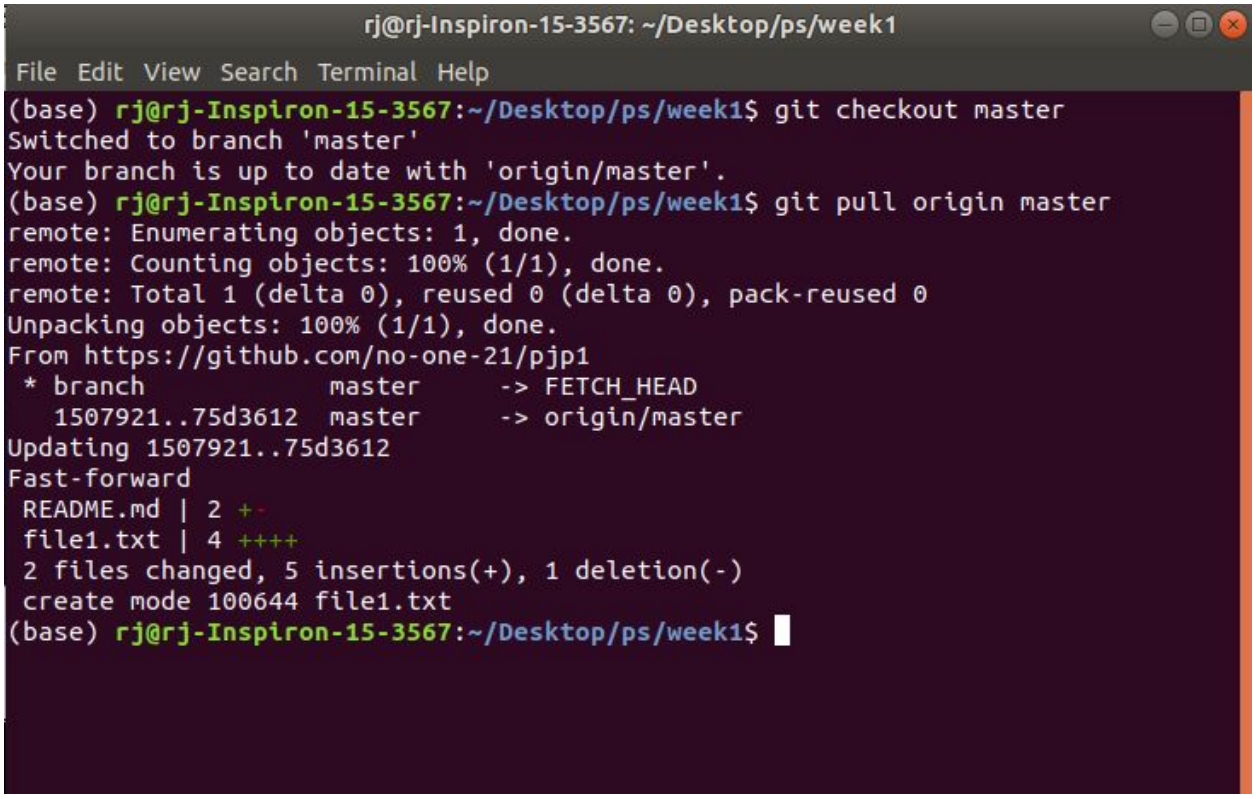


Opening a pull request





Pull master from origin



2. Perform above actions using vscode/git gui

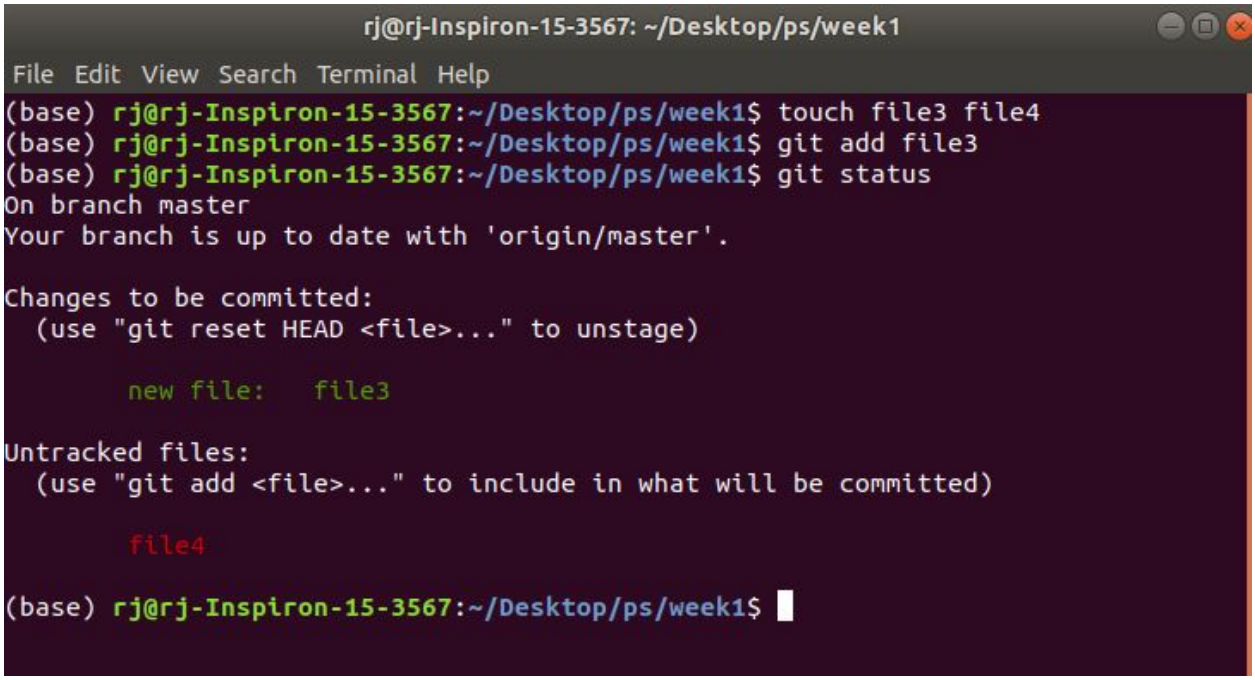
In vs code we can use different extensions like git lens for action like staging, commit, rebase, merging, pushing etc. and git blame to see changes added by each person.

3. On ground day to day scenarios

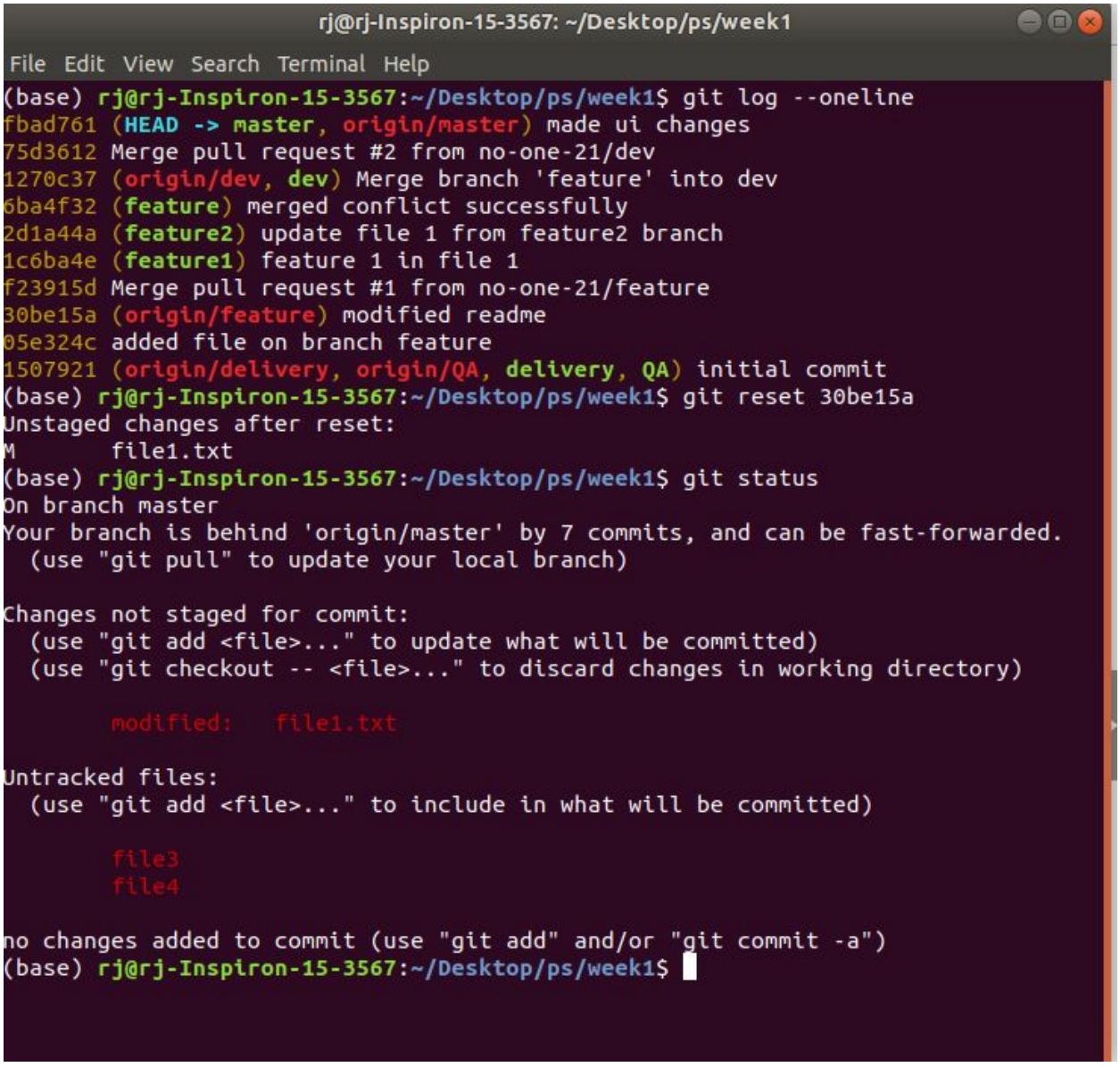
a. Git reset/revert changes (soft/hard reset)

--soft reset

Does not touch the index file or the working tree at all (but resets the head to <commit>, just like all modes do). This leaves all your changed files "Changes to be committed", as git status would put it.

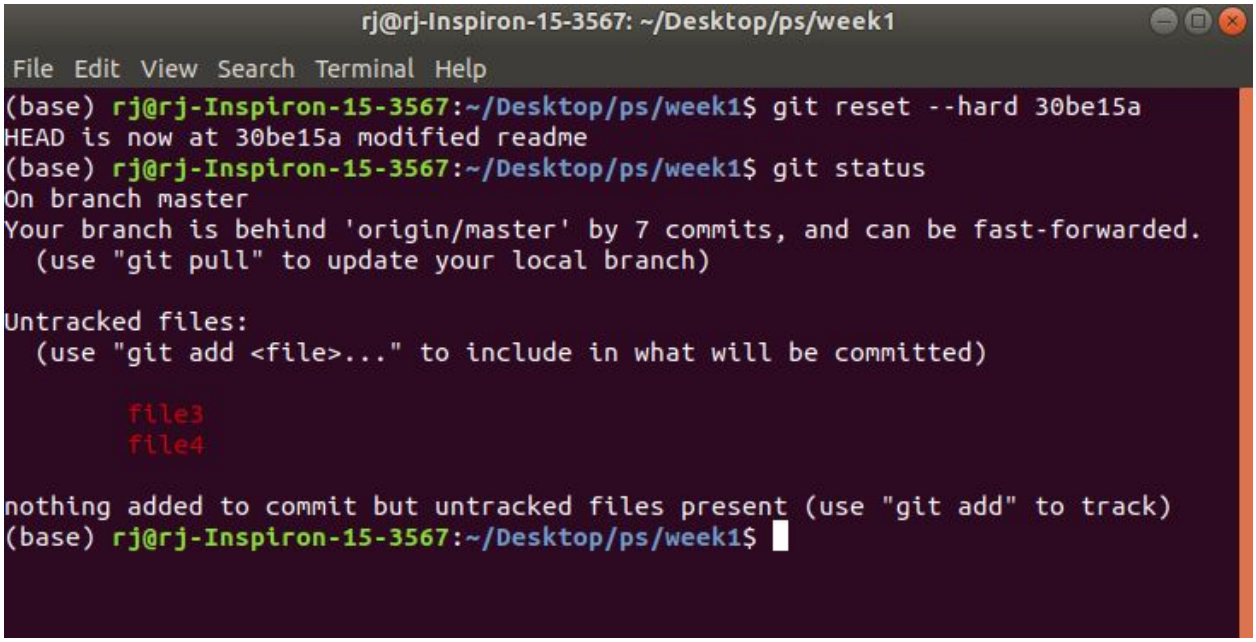


Git log after soft reset



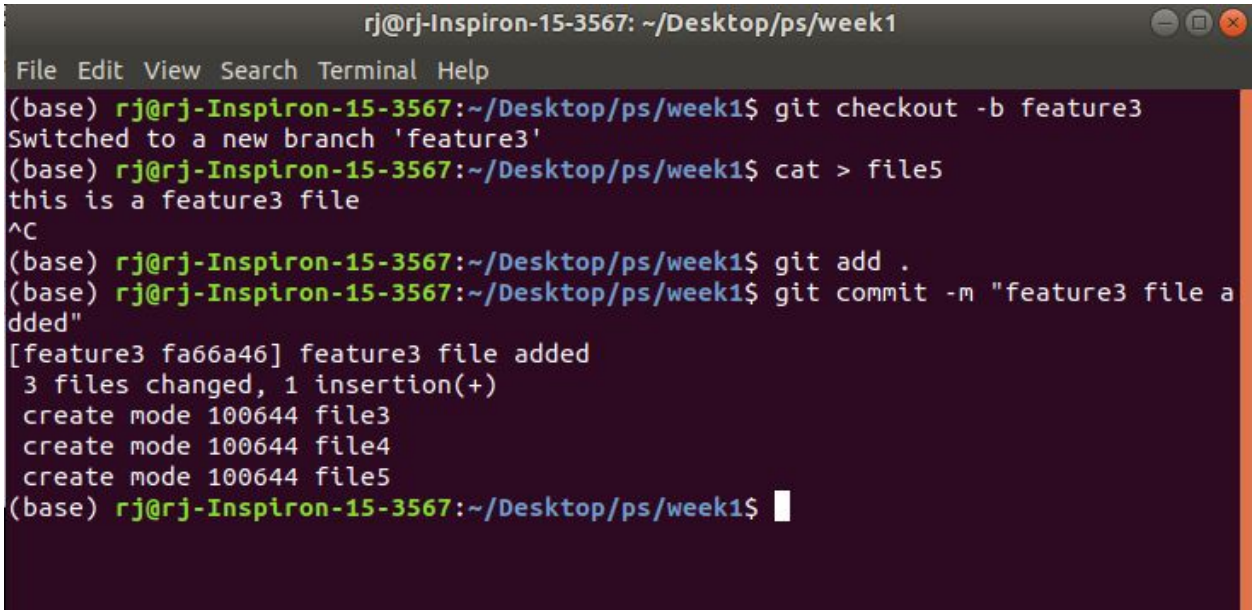
--hard reset example

Git log after hard reset

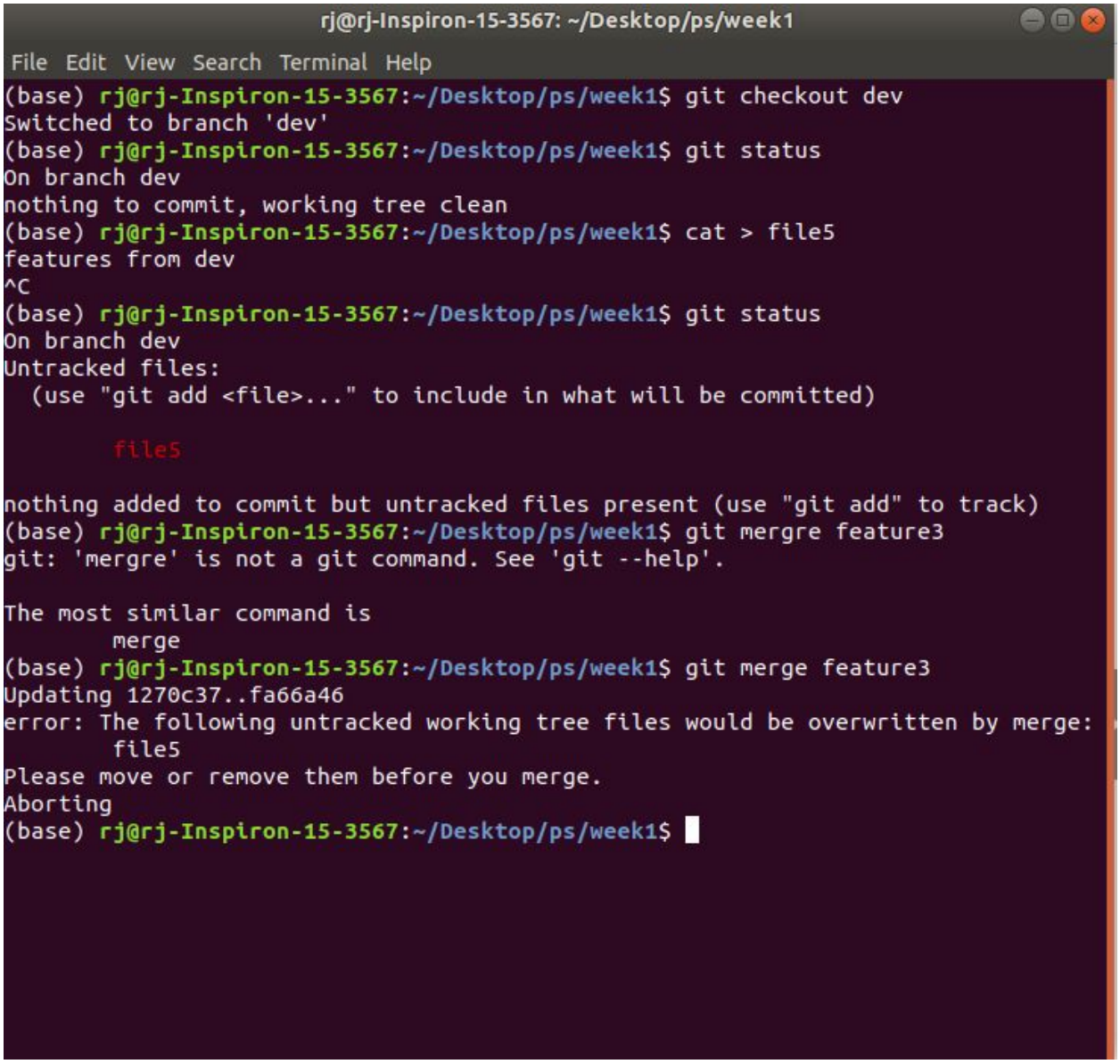


b. Git stash

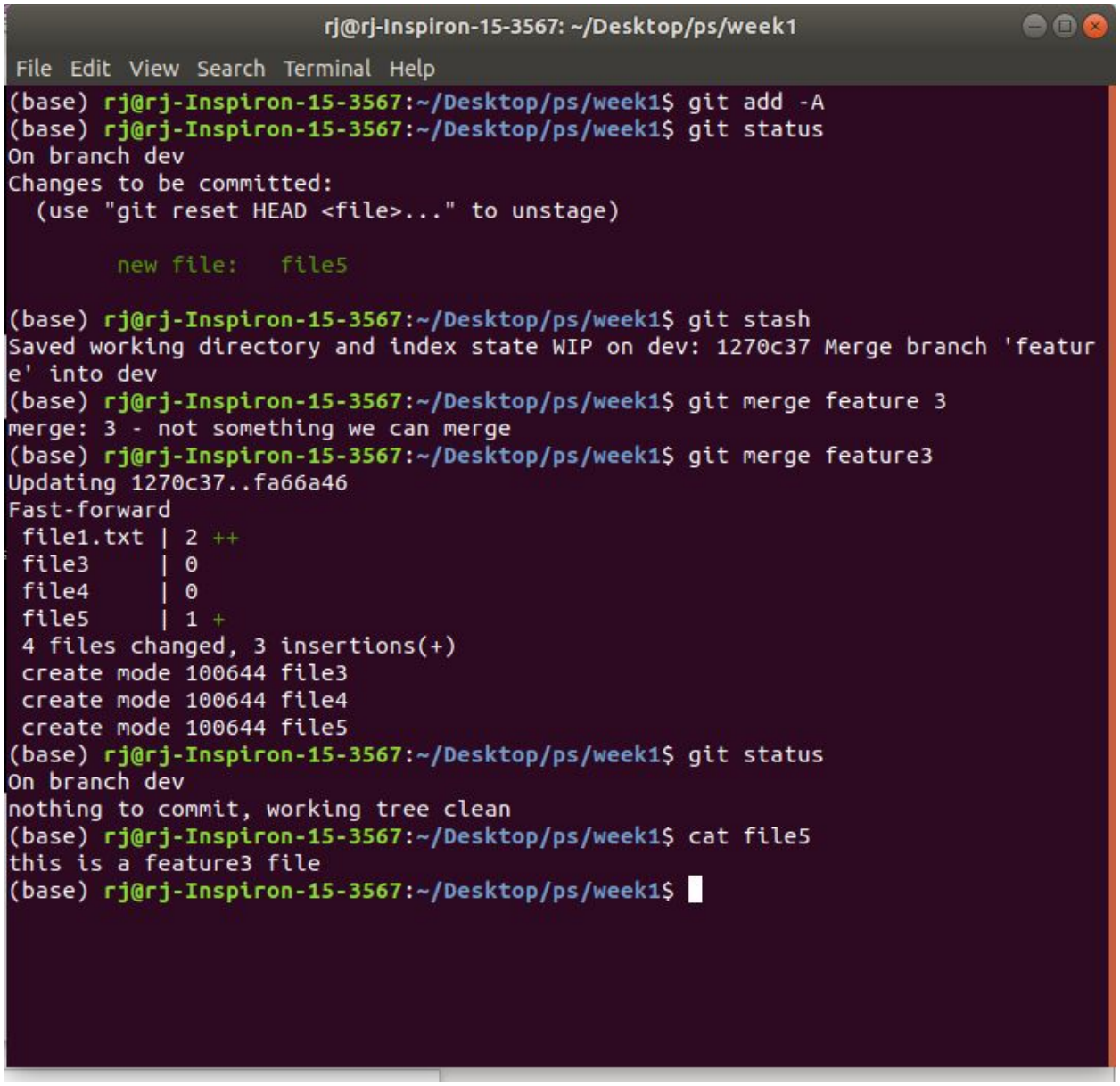
Let person 1 make changes to test.txt file and commit it



Now let master make changes and it tries to merge person after making changes.

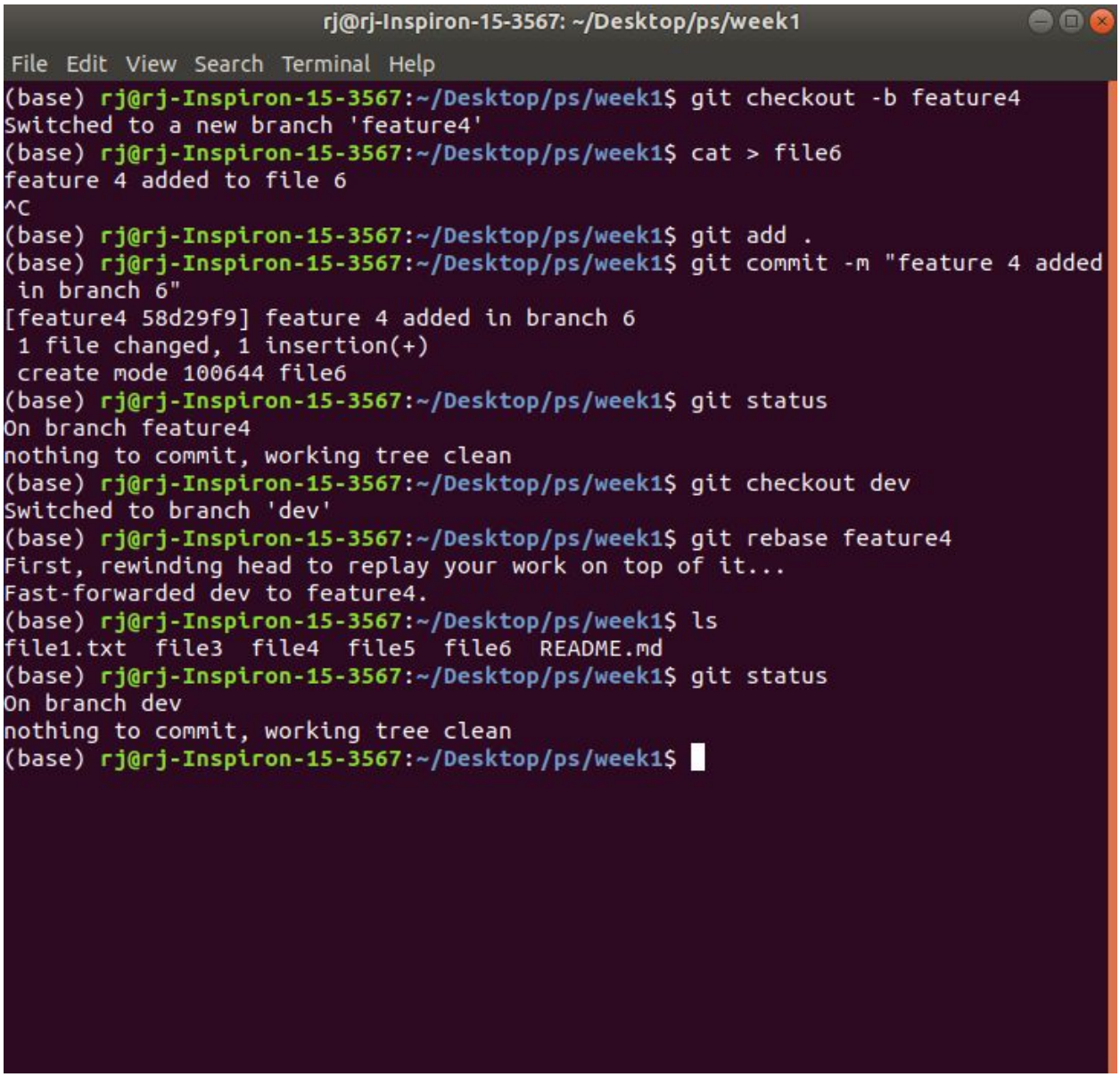


Performing git stash



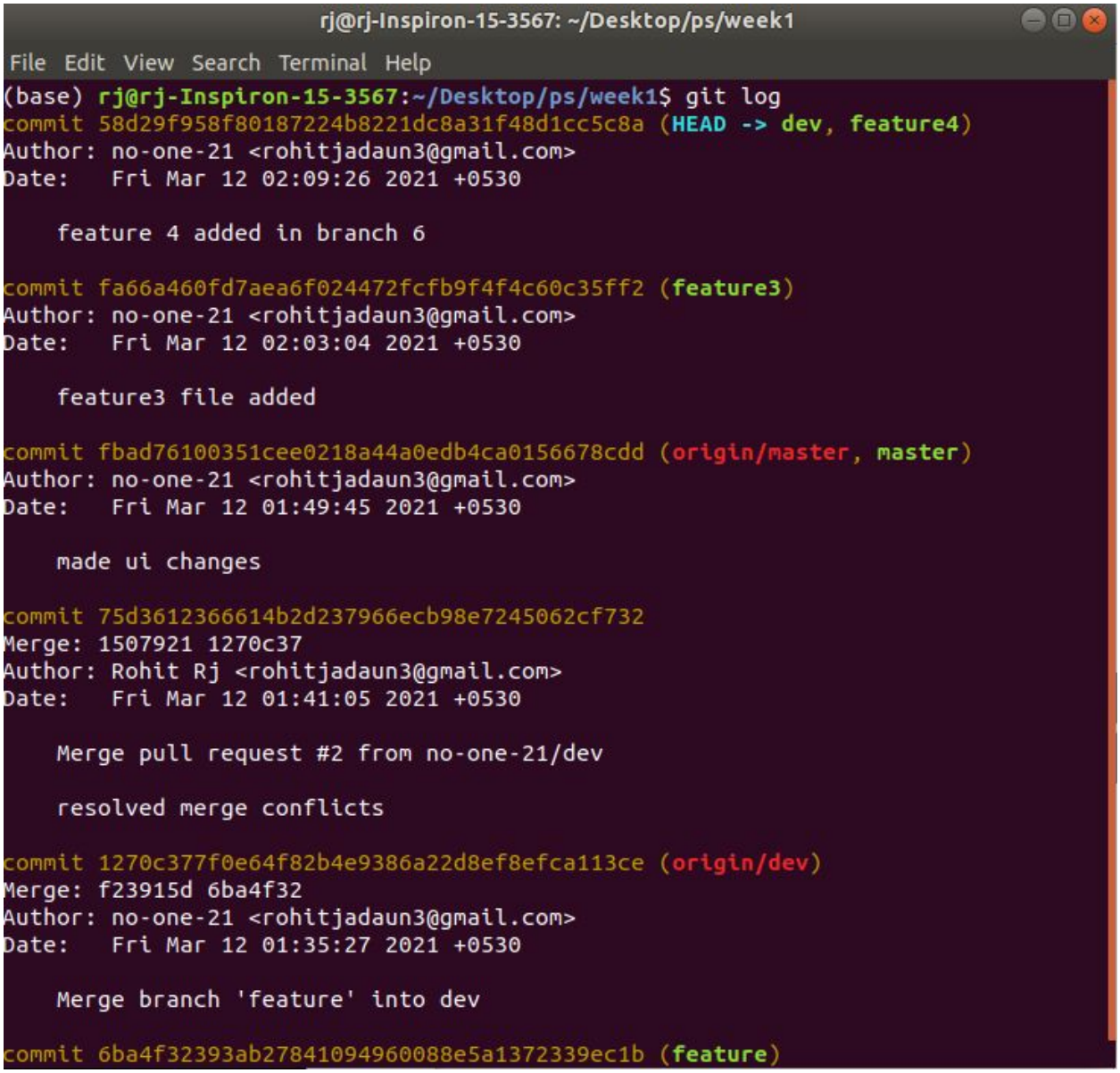
3. Git rebase

Adding new.txt file by master

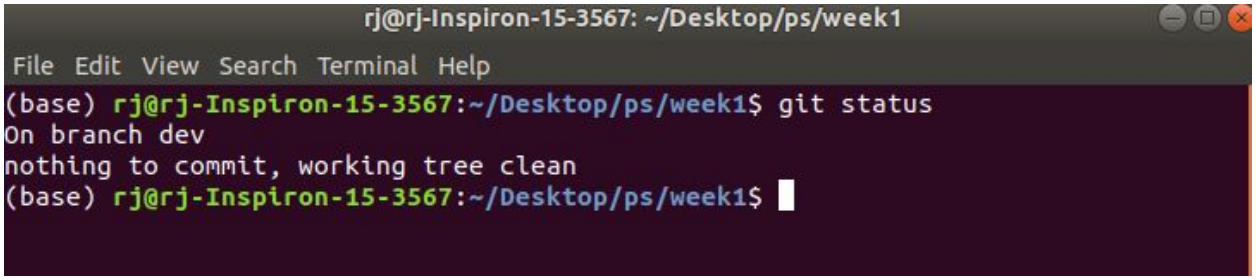


4. Git log ,reflog, status

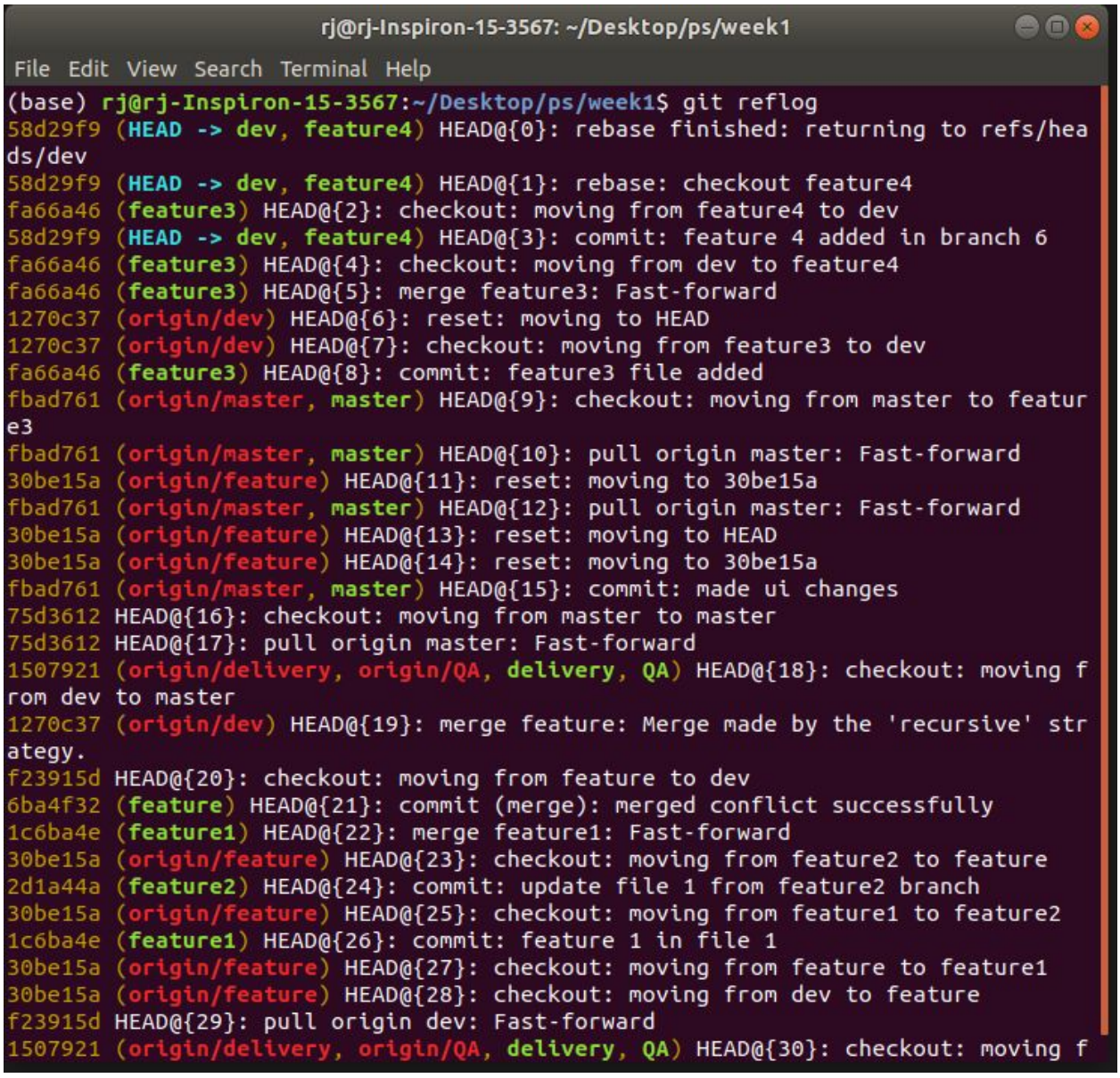
Git log



Git status


rj@rj-Inspiron-15-3567: ~/Desktop/ps/week1
File Edit View Search Terminal Help
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git status
On branch dev
nothing to commit, working tree clean
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$

Git reflog


rj@rj-Inspiron-15-3567: ~/Desktop/ps/week1
File Edit View Search Terminal Help
(base) rj@rj-Inspiron-15-3567:~/Desktop/ps/week1$ git reflog
58d29f9 (HEAD -> dev, feature4) HEAD@{0}: rebase finished: returning to refs/heads/dev
58d29f9 (HEAD -> dev, feature4) HEAD@{1}: rebase: checkout feature4
fa66a46 (feature3) HEAD@{2}: checkout: moving from feature4 to dev
58d29f9 (HEAD -> dev, feature4) HEAD@{3}: commit: feature 4 added in branch 6
fa66a46 (feature3) HEAD@{4}: checkout: moving from dev to feature4
fa66a46 (feature3) HEAD@{5}: merge feature3: Fast-forward
1270c37 (origin/dev) HEAD@{6}: reset: moving to HEAD
1270c37 (origin/dev) HEAD@{7}: checkout: moving from feature3 to dev
fa66a46 (feature3) HEAD@{8}: commit: feature3 file added
fbad761 (origin/master, master) HEAD@{9}: checkout: moving from master to feature3
fbad761 (origin/master, master) HEAD@{10}: pull origin master: Fast-forward
30be15a (origin/feature) HEAD@{11}: reset: moving to 30be15a
fbad761 (origin/master, master) HEAD@{12}: pull origin master: Fast-forward
30be15a (origin/feature) HEAD@{13}: reset: moving to HEAD
30be15a (origin/feature) HEAD@{14}: reset: moving to 30be15a
fbad761 (origin/master, master) HEAD@{15}: commit: made ui changes
75d3612 HEAD@{16}: checkout: moving from master to master
75d3612 HEAD@{17}: pull origin master: Fast-forward
1507921 (origin/delivery, origin/QA, delivery, QA) HEAD@{18}: checkout: moving from dev to master
1270c37 (origin/dev) HEAD@{19}: merge feature: Merge made by the 'recursive' strategy.
f23915d HEAD@{20}: checkout: moving from feature to dev
6ba4f32 (feature) HEAD@{21}: commit (merge): merged conflict successfully
1c6ba4e (feature1) HEAD@{22}: merge feature1: Fast-forward
30be15a (origin/feature) HEAD@{23}: checkout: moving from feature2 to feature
2d1a44a (feature2) HEAD@{24}: commit: update file 1 from feature2 branch
30be15a (origin/feature) HEAD@{25}: checkout: moving from feature1 to feature2
1c6ba4e (feature1) HEAD@{26}: commit: feature 1 in file 1
30be15a (origin/feature) HEAD@{27}: checkout: moving from feature to feature1
30be15a (origin/feature) HEAD@{28}: checkout: moving from dev to feature
f23915d HEAD@{29}: pull origin dev: Fast-forward
1507921 (origin/delivery, origin/QA, delivery, QA) HEAD@{30}: checkout: moving f