GIT RESET

Git add .----moves to staging area

Commit---commits the changes

Soft----commit id is deleted and changes remains same moves from comes back from commit area

Mixed---commit id is deleted and chenges done remains same comes back from staging area

Hard--changes are deleted and comes three stages back   
  
  
In git reset we have three types

Git reset --soft commitid

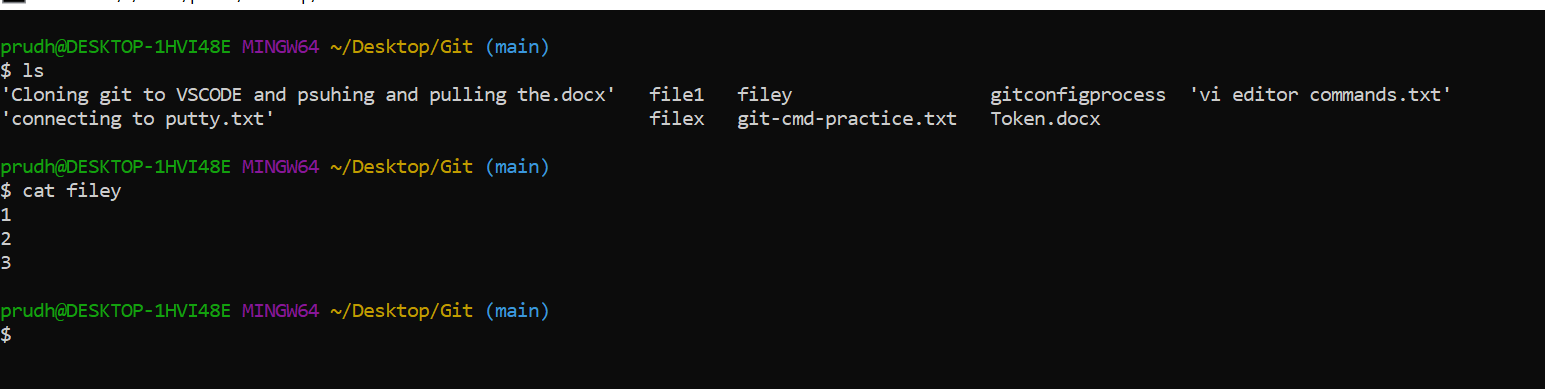
Git reset--mixed commitid

Git reset--hard commitid

**Soft**

Comes back from staging area-----------one step back

Take below example

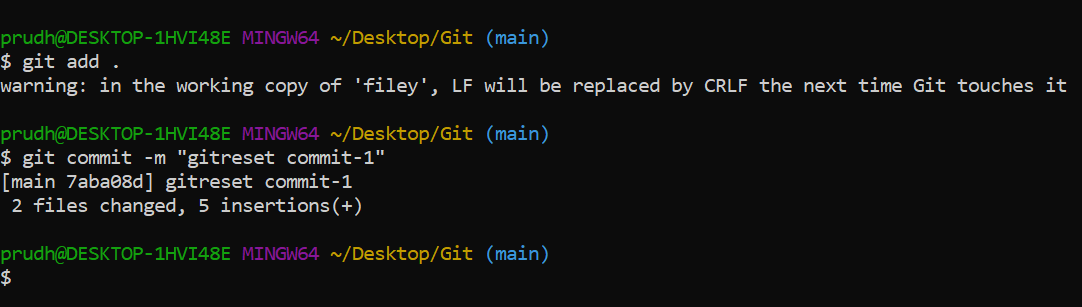


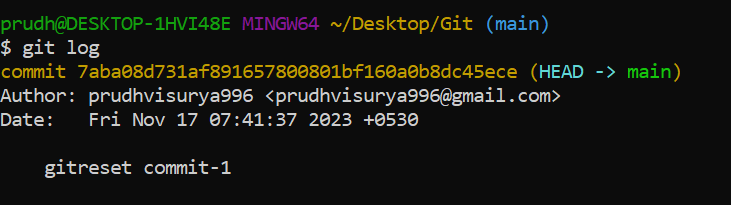
Now edited the file as below



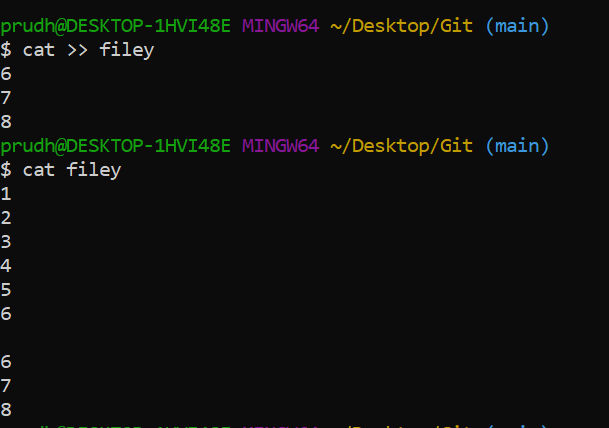


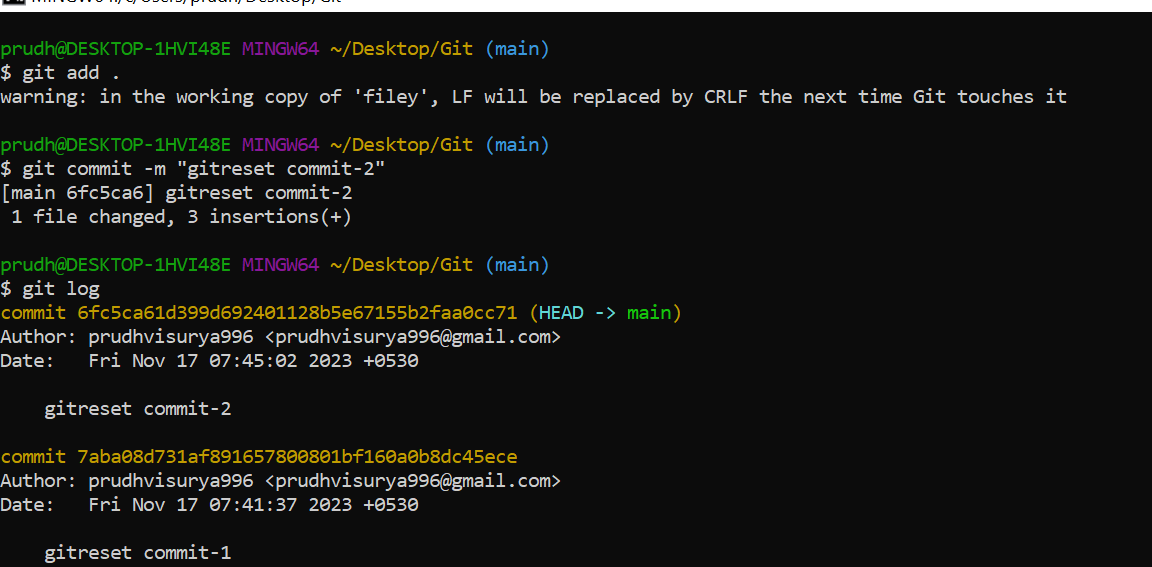
Now creating first commit

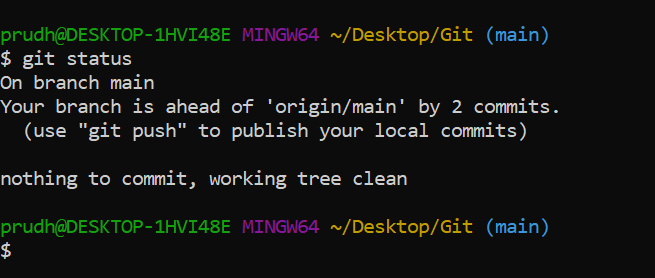




Do some changes in file again and go for 2nd commit



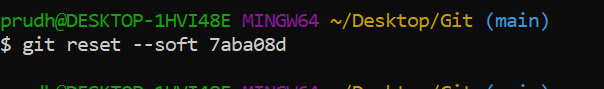




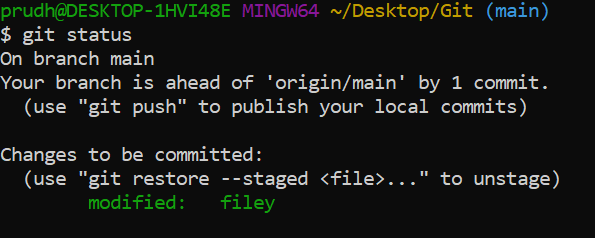
Now if we want to revert changes done in 2nd commit then we have to reset 1st commit

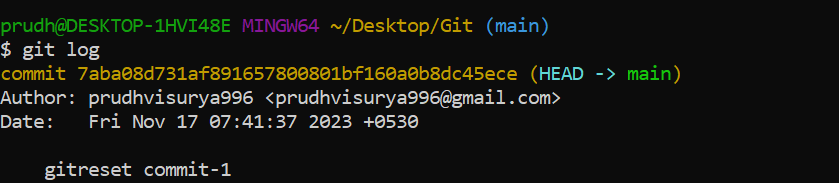
As shown below

Note down the first 7 digits in 1st commit

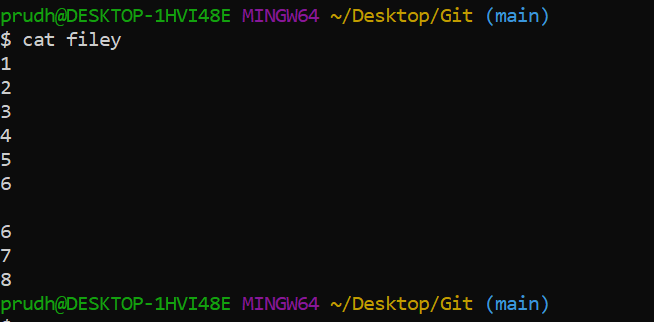


Now you can 2nd commit eliminated from git log and file comes back from staging area





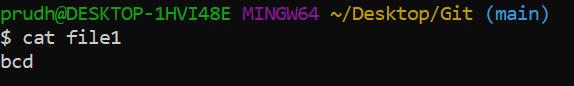
But file will be same as chanages done before doing second commit



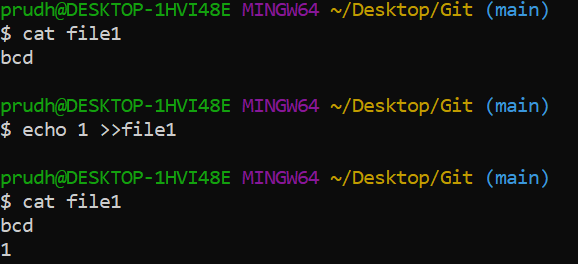
Only commit id is deleted from log and file come backs from commit area

Changes in file remain the same

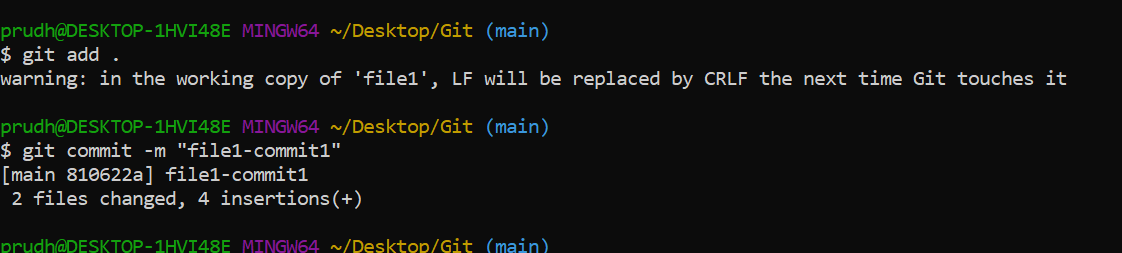
Now lets take example of mixed reset---2 stages back  
  
take new file

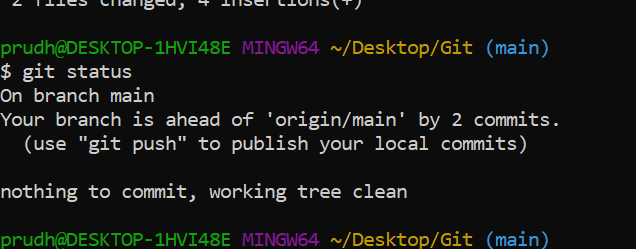


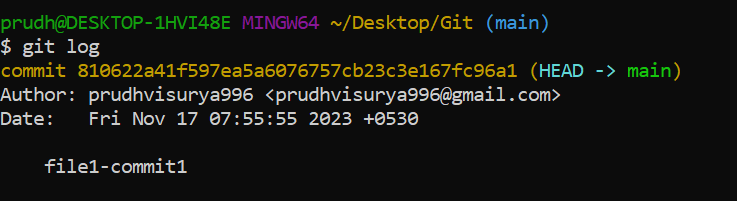
Edits done in file1



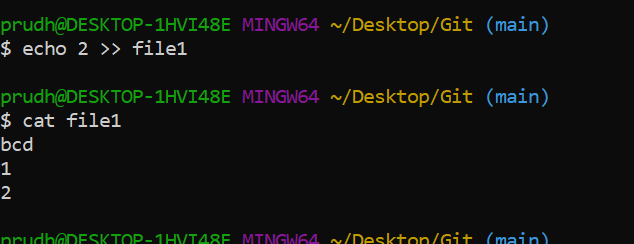
Do git add . and commit







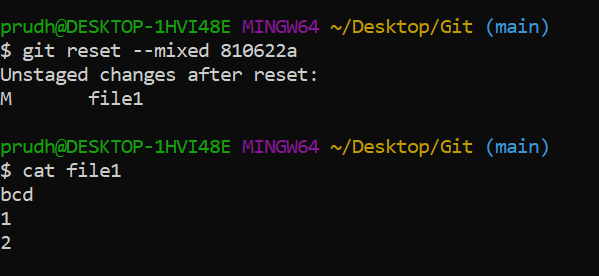
Now go forsecond commit





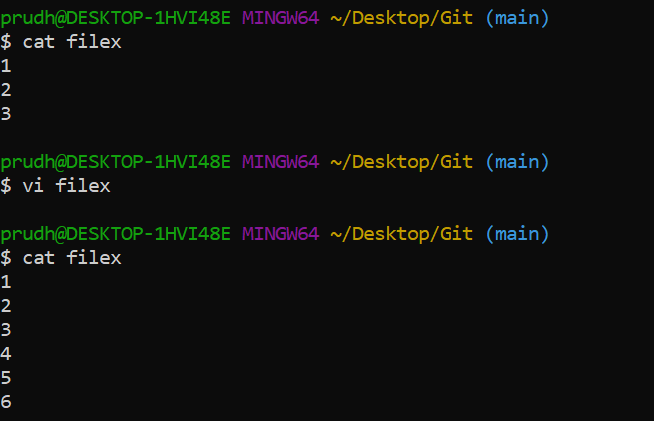
Now if we want the 2nd commit to be reverted and unstage

Changes in file remain the same



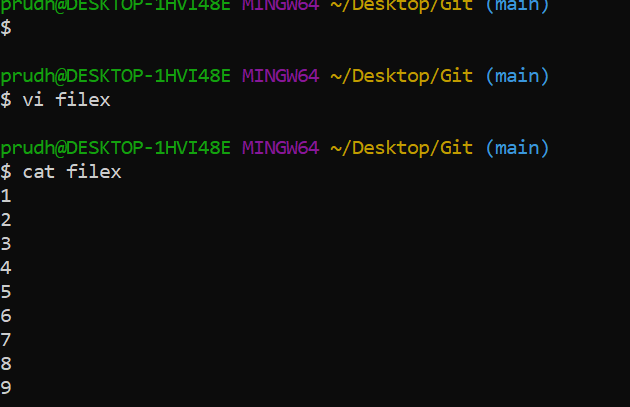
If we want the changes also to be deleted then we use hard 3 stages back

Take new file

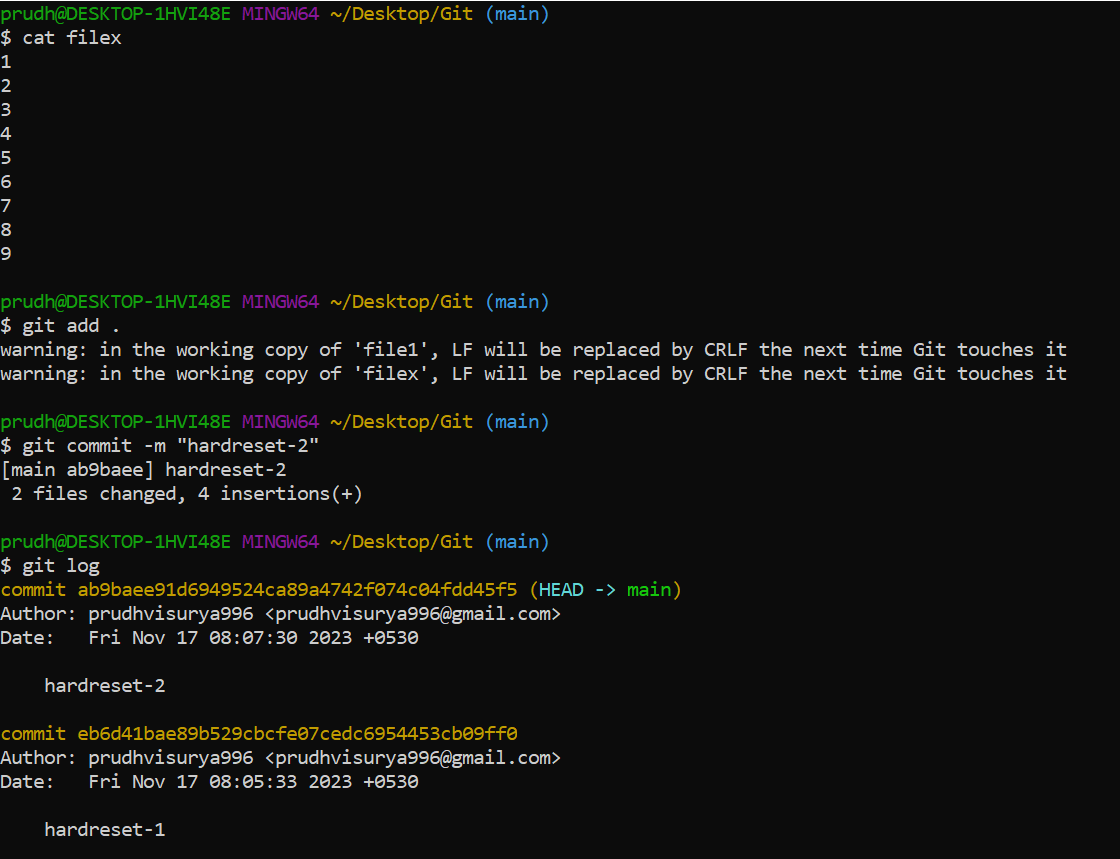


Now do git add and commit





Now do second commit mean while after commit check log and status





You can see second commit is deleted and changes done in second commit are also deleted