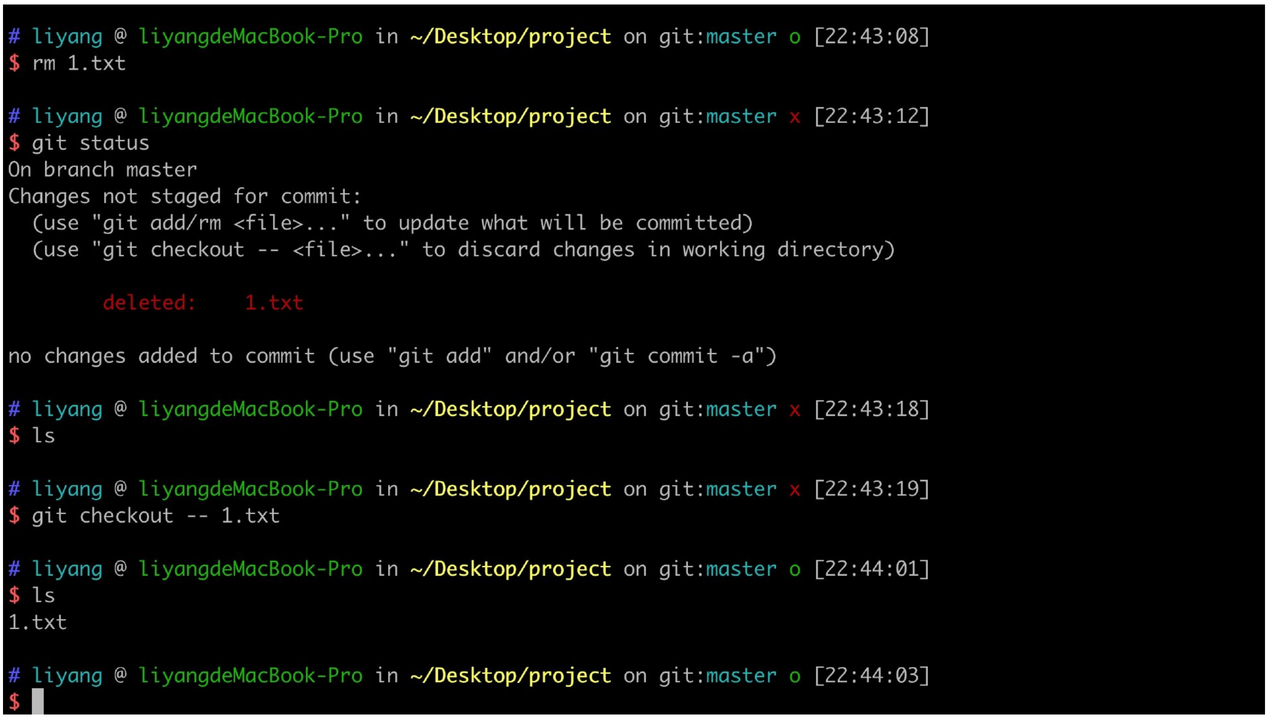
恢复被删除文件

我们可以通过checkout命令，来恢复我们删除的文件

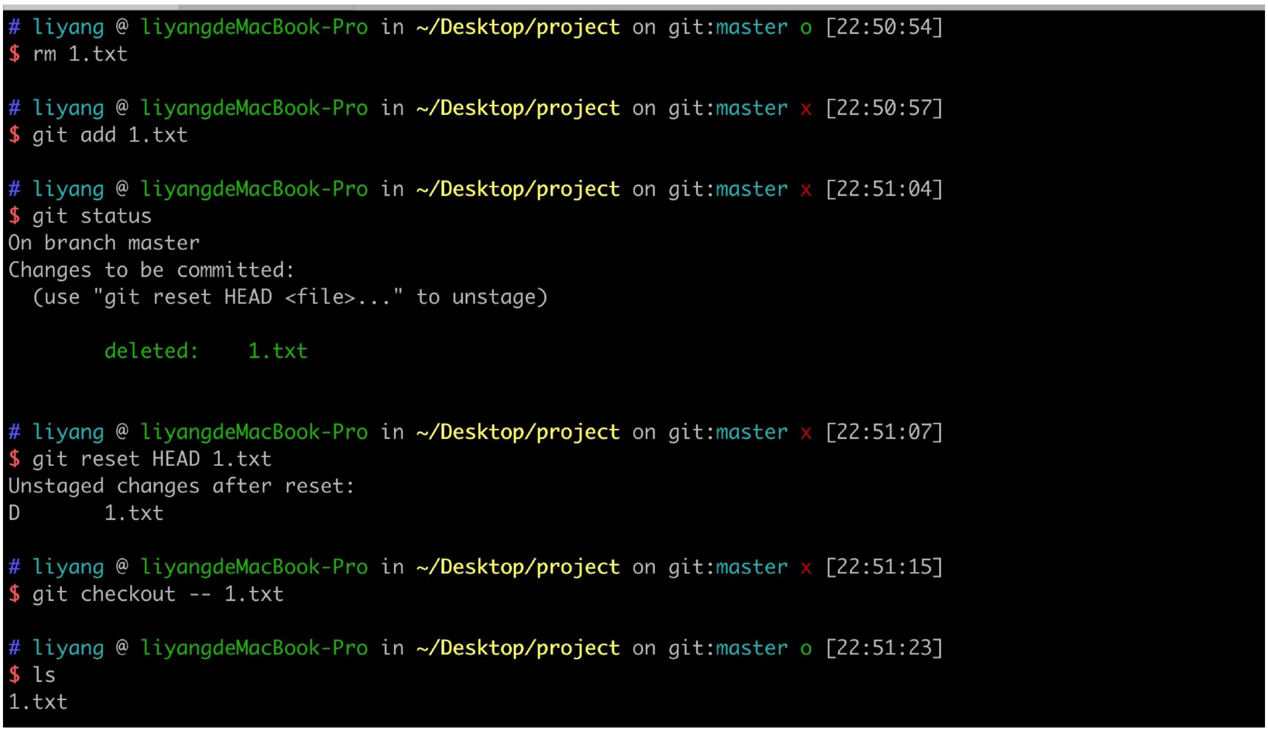
比如下图所示，没有执行添加暂存区，恢复被删除的文件



```
# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master o [22:43:08]
$ rm 1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:43:12]
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    1.txt

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

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:43:18]
$ ls

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:43:19]
$ git checkout -- 1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master o [22:44:01]
$ ls
1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master o [22:44:03]
$ 
```

如果已经添加到了暂存区，先执行撤销操作，在执行恢复操作

```
git status 先看一下add 中的文件


git reset HEAD 如果后面什么都不跟的话 就是上一次add 里面的全部撤销了
git reset HEAD 1.txt 就是对某个文件进行撤销了
```



```
# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master o [22:50:54]
$ rm 1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:50:57]
$ git add 1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:51:04]
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    1.txt


# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:51:07]
$ git reset HEAD 1.txt
Unstaged changes after reset:
D       1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master x [22:51:15]
$ git checkout -- 1.txt

# liyang @ liyangdeMacBook-Pro in ~/Desktop/project on git:master o [22:51:23]
$ ls
1.txt
```