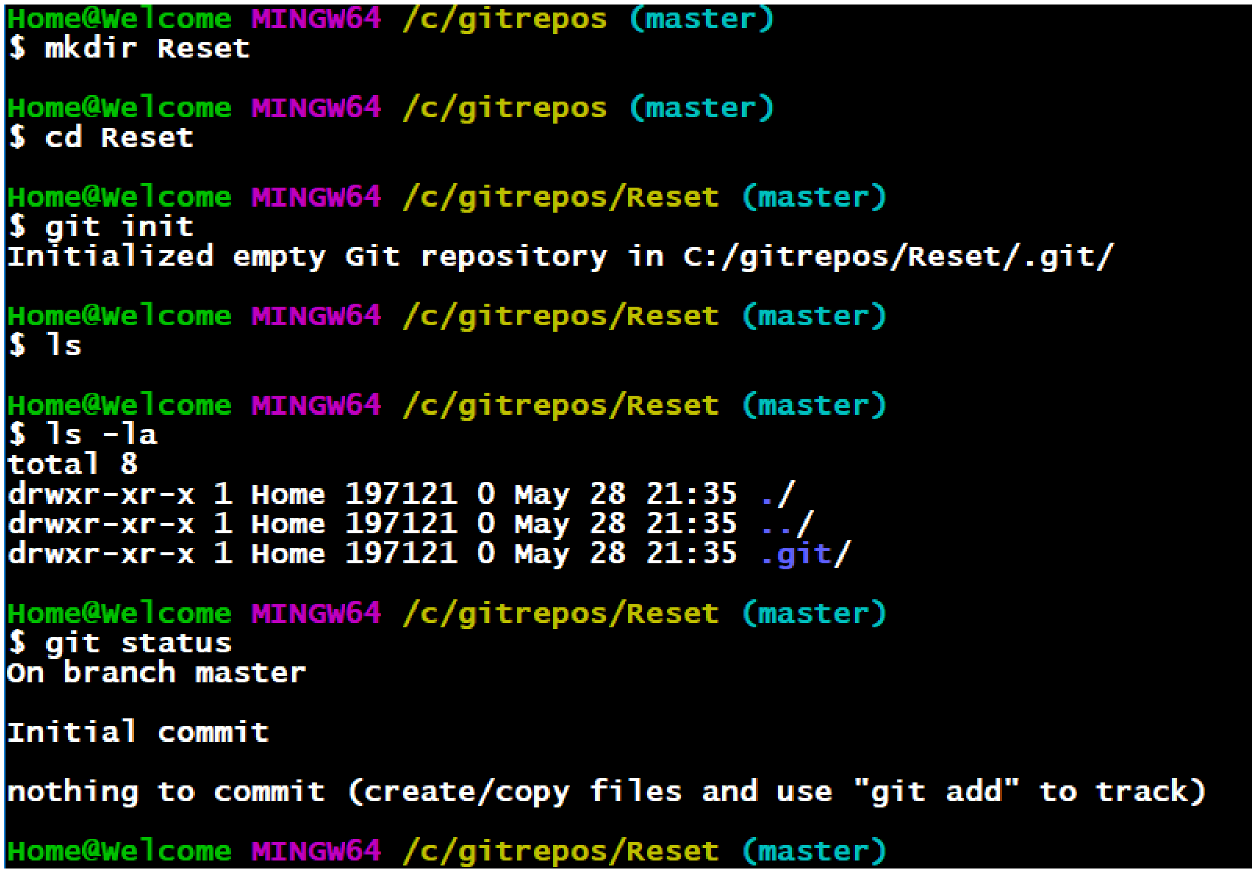
Here am going to create a new repository at working area and creating code files for our project and then coming back to old versions of a file using Reset

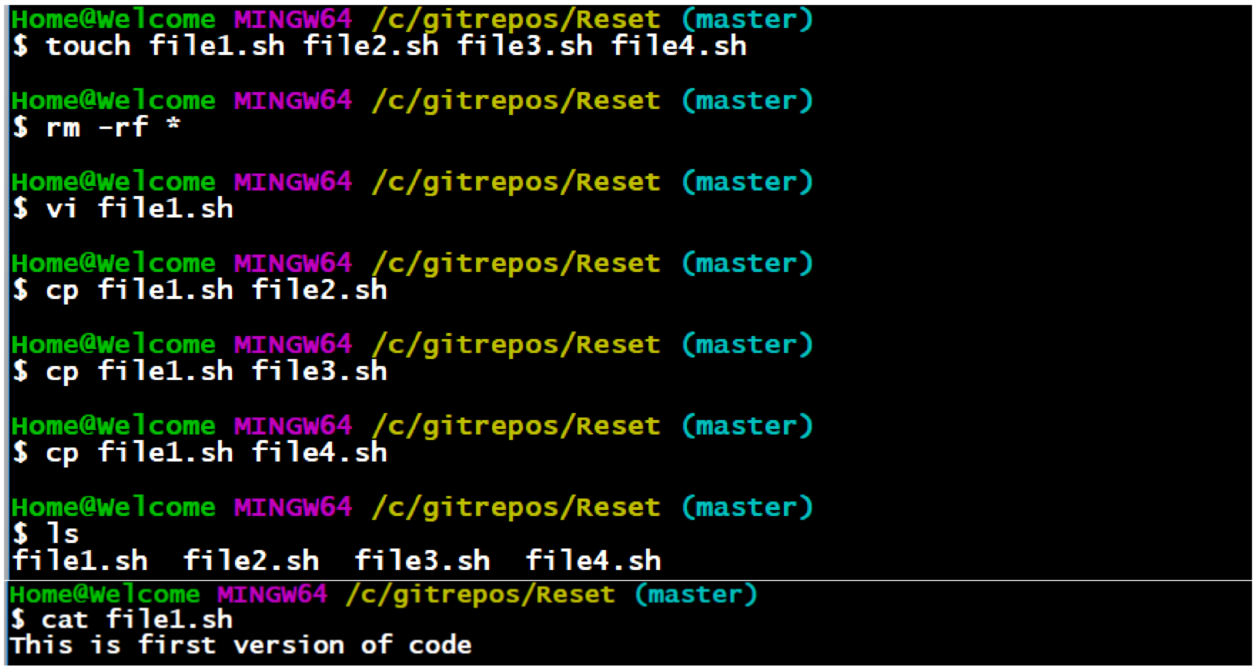
Open git bash

Go to newly created folder



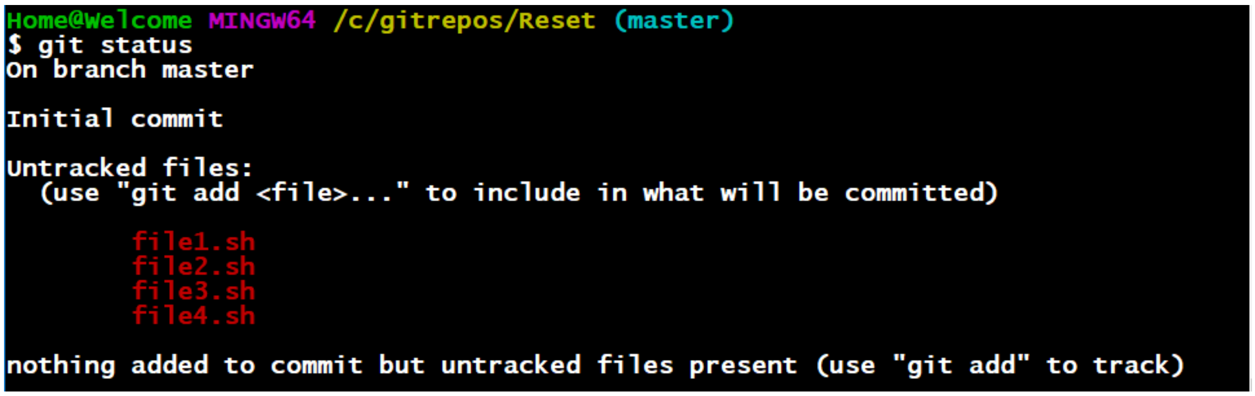
```
Home@Welcome MINGW64 /c/gitrepos (master)
$ mkdir Reset

Home@Welcome MINGW64 /c/gitrepos (master)
$ cd Reset

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git init
Initialized empty Git repository in C:/gitrepos/Reset/.git/

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ ls

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ ls -la
total 8
drwxr-xr-x 1 Home 197121 0 May 28 21:35 ./
drwxr-xr-x 1 Home 197121 0 May 28 21:35 ../
drwxr-xr-x 1 Home 197121 0 May 28 21:35 .git/

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
```

and create files to our project like below.



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ touch file1.sh file2.sh file3.sh file4.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ rm -rf *

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ vi file1.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cp file1.sh file2.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cp file1.sh file3.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cp file1.sh file4.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ ls
file1.sh  file2.sh  file3.sh  file4.sh
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file1.sh
This is first version of code
```

We check my repository status here like below



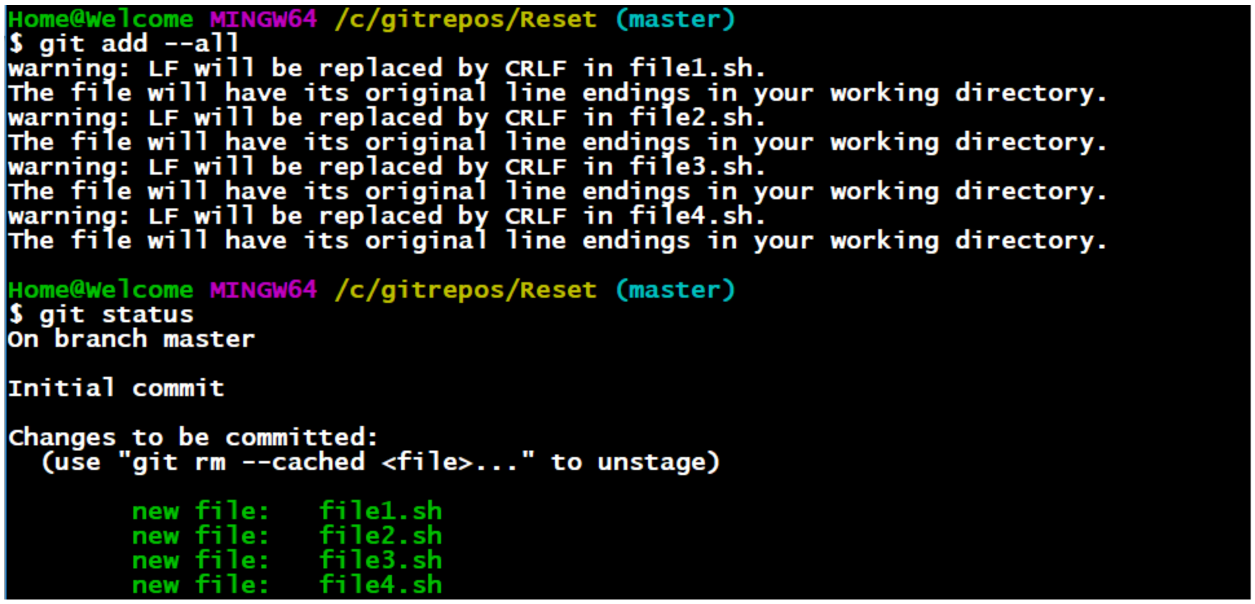
```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master

Initial commit

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

        file1.sh
        file2.sh
        file3.sh
        file4.sh

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

Now we can add all our files to staging area using below commands

git add <filename>

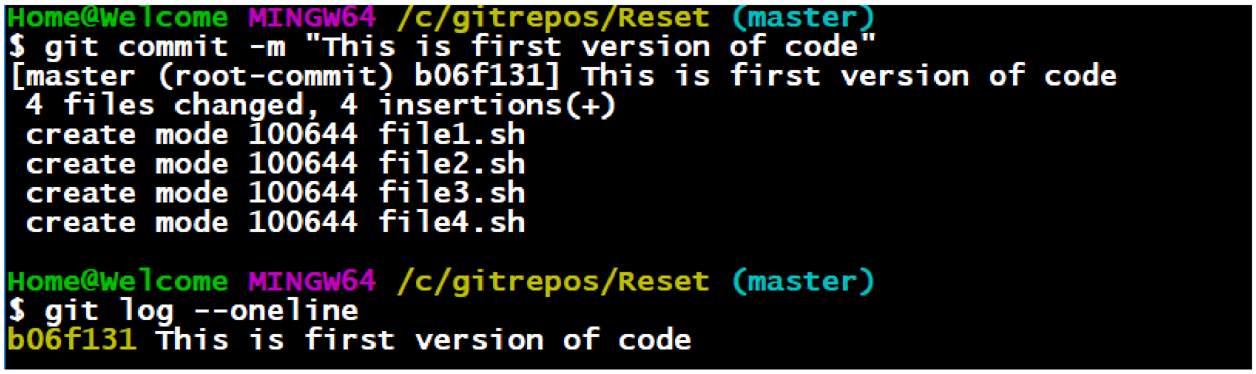
git add .

git add –all



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git add --all
warning: LF will be replaced by CRLF in file1.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file2.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file3.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file4.sh.
The file will have its original line endings in your working directory.

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master

Initial commit

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

        new file:   file1.sh
        new file:   file2.sh
        new file:   file3.sh
        new file:   file4.sh
```

We can commit our staged changes to our local repository.

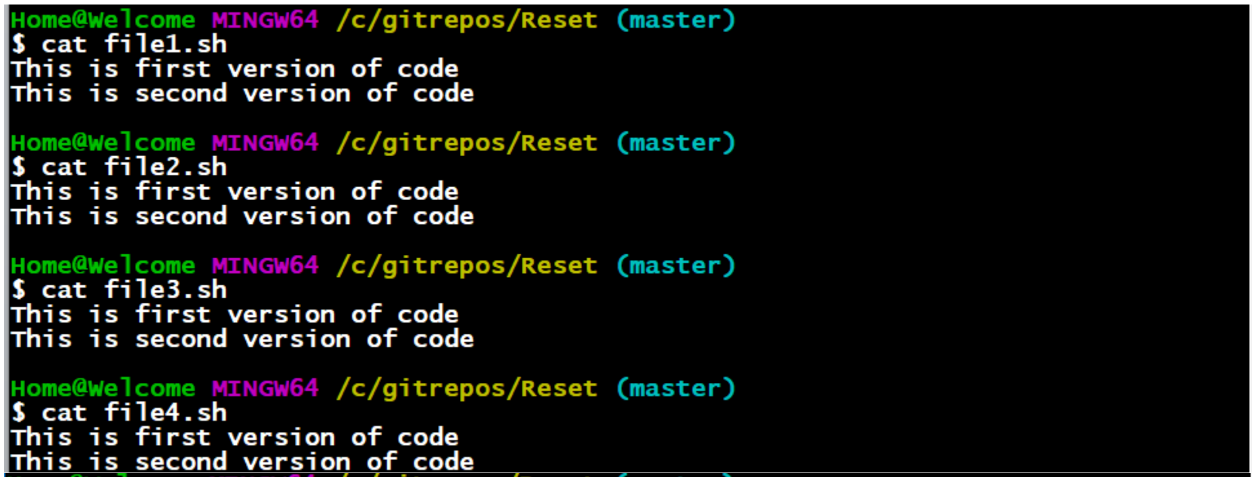


```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git commit -m "This is first version of code"
[master (root-commit) b06f131] This is first version of code
 4 files changed, 4 insertions(+)
 create mode 100644 file1.sh
 create mode 100644 file2.sh
 create mode 100644 file3.sh
 create mode 100644 file4.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git log --oneline
b06f131 This is first version of code
```

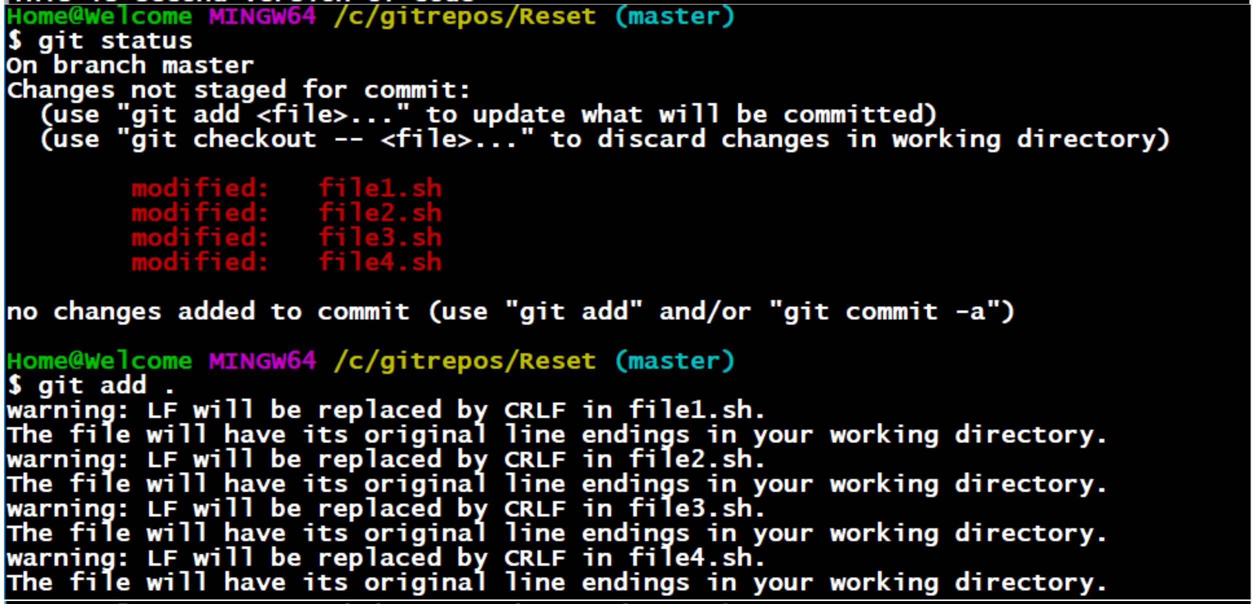
We can ready to push our first commit changes to Server repository which is available on github.

But Now we can do more changes to our repository to teset Reset command here.

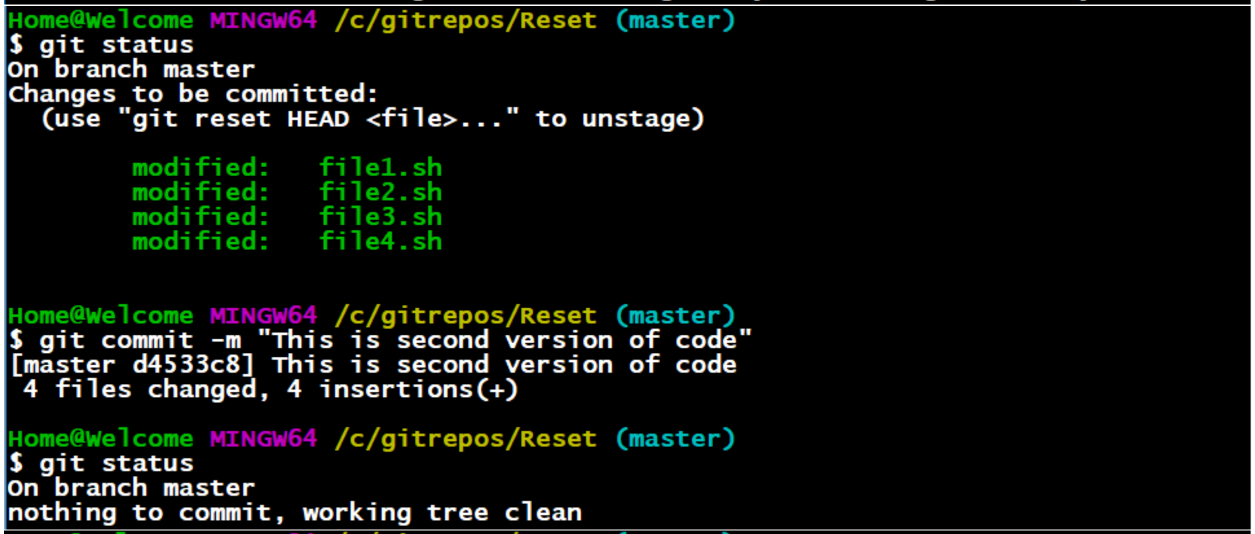
Again adding one more line to all our files like below



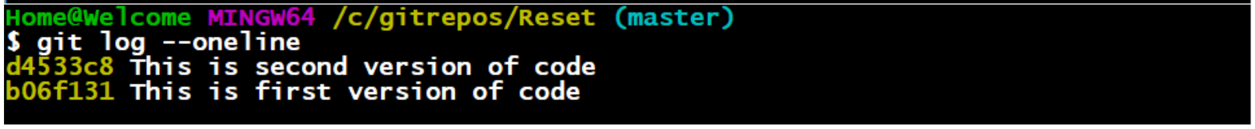
```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file1.sh
This is first version of code
This is second version of code

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file2.sh
This is first version of code
This is second version of code

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file3.sh
This is first version of code
This is second version of code

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file4.sh
This is first version of code
This is second version of code
```



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ 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:   file1.sh
        modified:   file2.sh
        modified:   file3.sh
        modified:   file4.sh

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

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git add .
warning: LF will be replaced by CRLF in file1.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file2.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file3.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file4.sh.
The file will have its original line endings in your working directory.
```



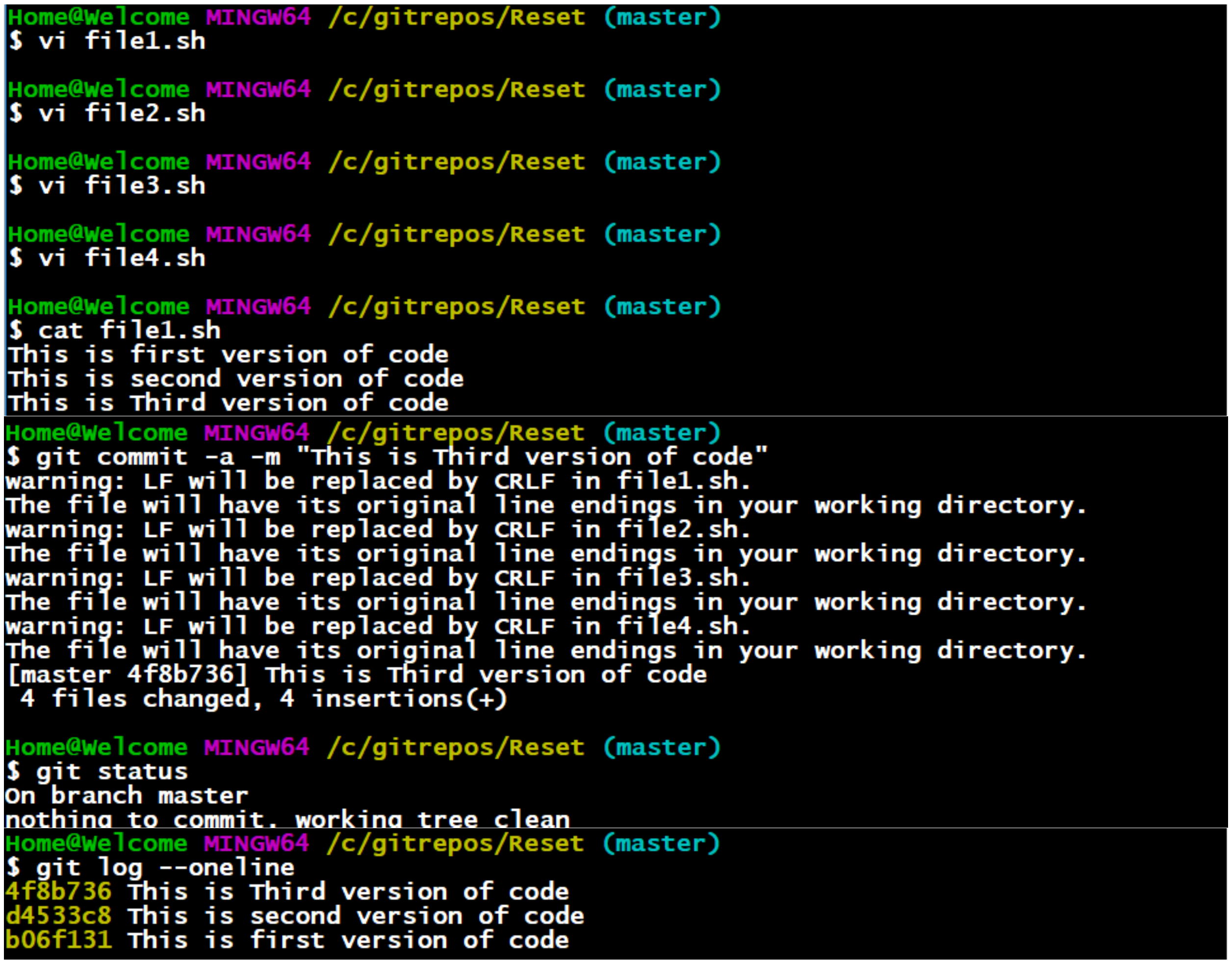
```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   file1.sh
        modified:   file2.sh
        modified:   file3.sh
        modified:   file4.sh


Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git commit -m "This is second version of code"
[master d4533c8] This is second version of code
 4 files changed, 4 insertions(+)

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master
nothing to commit, working tree clean
```



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git log --oneline
d4533c8 This is second version of code
b06f131 This is first version of code
```

Note : HEAD --- > Means the latest commit so always HEAD points latest commit id in GIT

Again we can do one more addition to our code files like above



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ vi file1.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ vi file2.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ vi file3.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ vi file4.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file1.sh
This is first version of code
This is second version of code
This is Third version of code
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git commit -a -m "This is Third version of code"
warning: LF will be replaced by CRLF in file1.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file2.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file3.sh.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in file4.sh.
The file will have its original line endings in your working directory.
[master 4f8b736] This is Third version of code
 4 files changed, 4 insertions(+)

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master
nothing to commit, working tree clean
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git log --oneline
4f8b736 This is Third version of code
d4533c8 This is second version of code
b06f131 This is first version of code
```

Now we want to go backward means we don't want to continue with committed changes to GIT hub so here for this task we are going to use soft and hard reset option like below,

Now we have 3 commits and currently HEAD is pointing to lateset commit id i.e 3rd commit in our repo.

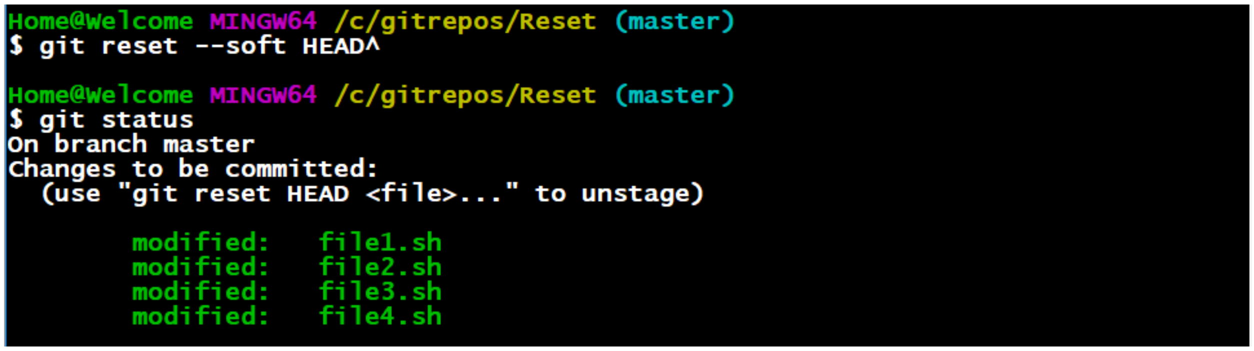
Before doing reset command check the status about the repository and we can check what is our last commit.So if you do reset for one commit we are going back to second commit means second commit gping to be points second commit.

Now the status of repo is working directory is clean and nothing to commit and now do reset like below for one commit go back.

git reset - -soft HEAD^

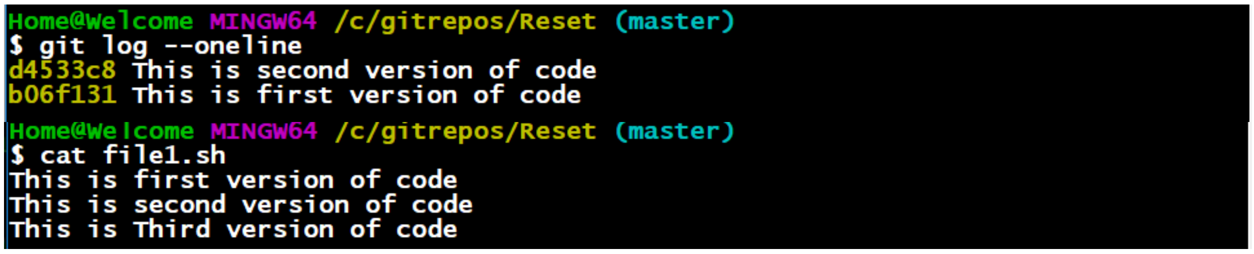
----- ^ ---- cap means we are going back to one previous commit ID

----- ^^ ----- double cap means we are going back to two prvious commit IDs.



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git reset --soft HEAD^

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   file1.sh
        modified:   file2.sh
        modified:   file3.sh
        modified:   file4.sh
```

Before doing reset working tree is clean when we check the status about our repo but after doing soft reset now the status about repo is showing there are files are available at staging area and ready to commit files. Means soft means it will just remove commit history and moves to only staging area. Ie called soft reset.

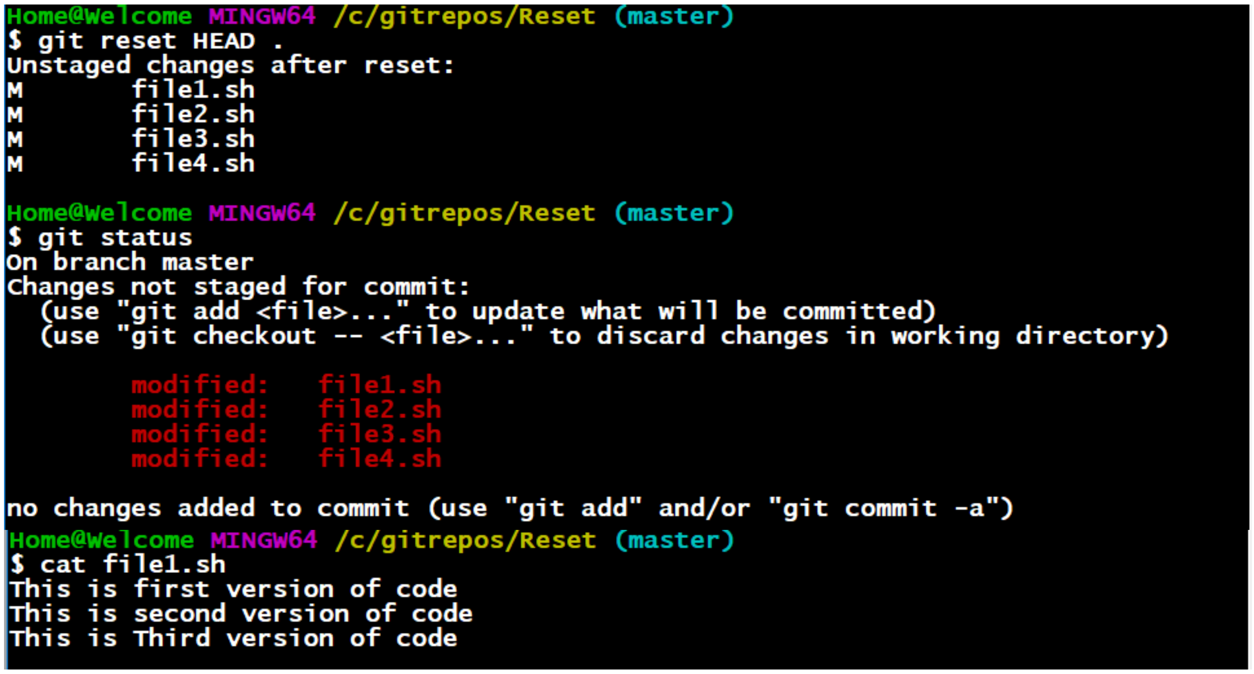


```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git log --oneline
d4533c8 This is second version of code
b06f131 This is first version of code
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file1.sh
This is first version of code
This is second version of code
This is Third version of code
```

Now am going to remove all files from staging area to un staging area using below command

git reset HEAD .   or   git reset HEAD <filename1>

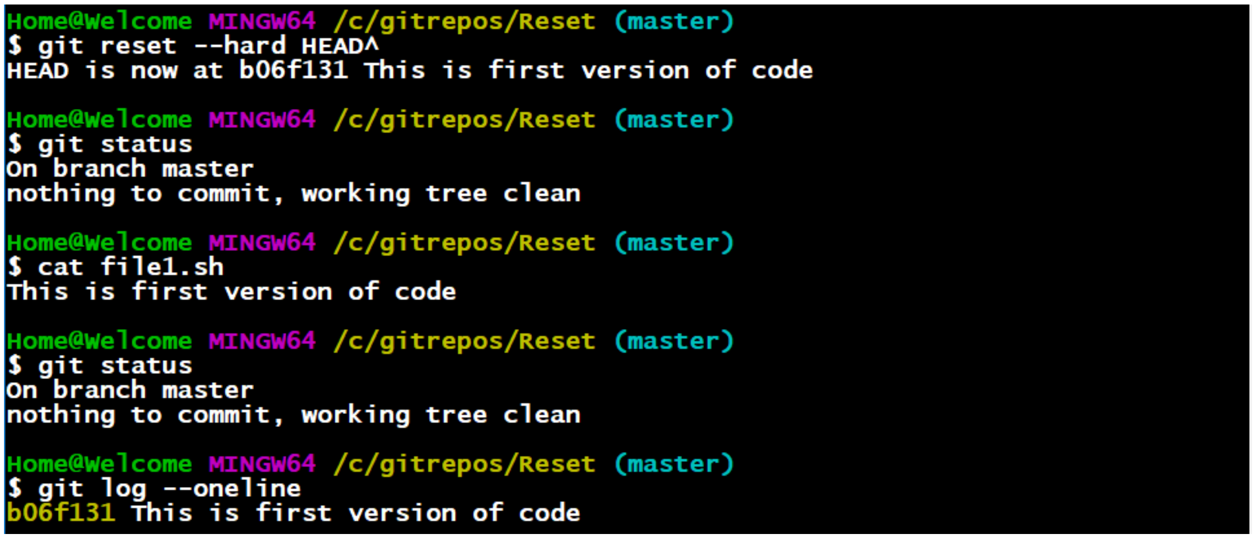
. ---- > means all files are available in staging area



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git reset HEAD .
Unstaged changes after reset:
M       file1.sh
M       file2.sh
M       file3.sh
M       file4.sh

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ 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:   file1.sh
        modified:   file2.sh
        modified:   file3.sh
        modified:   file4.sh

no changes added to commit (use "git add" and/or "git commit -a")
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file1.sh
This is first version of code
This is second version of code
This is Third version of code
```

In this way we have done reverse of our code using soft reset in 2 steps

Now doing hard reset below steps



```
Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git reset --hard HEAD^
HEAD is now at b06f131 This is first version of code

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master
nothing to commit, working tree clean

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ cat file1.sh
This is first version of code

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git status
On branch master
nothing to commit, working tree clean

Home@Welcome MINGW64 /c/gitrepos/Reset (master)
$ git log --oneline
b06f131 This is first version of code
```

Hard reset wil not bring files to unstage area directly it will remove changes from all files till the the last commit id files status. So it is very dangerous command be careful while using hard reset.