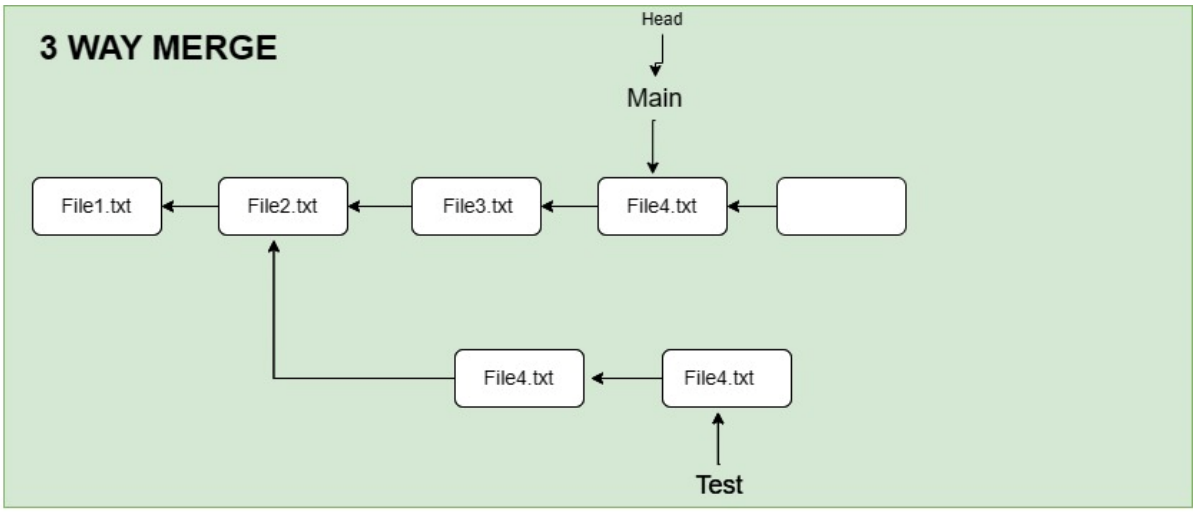
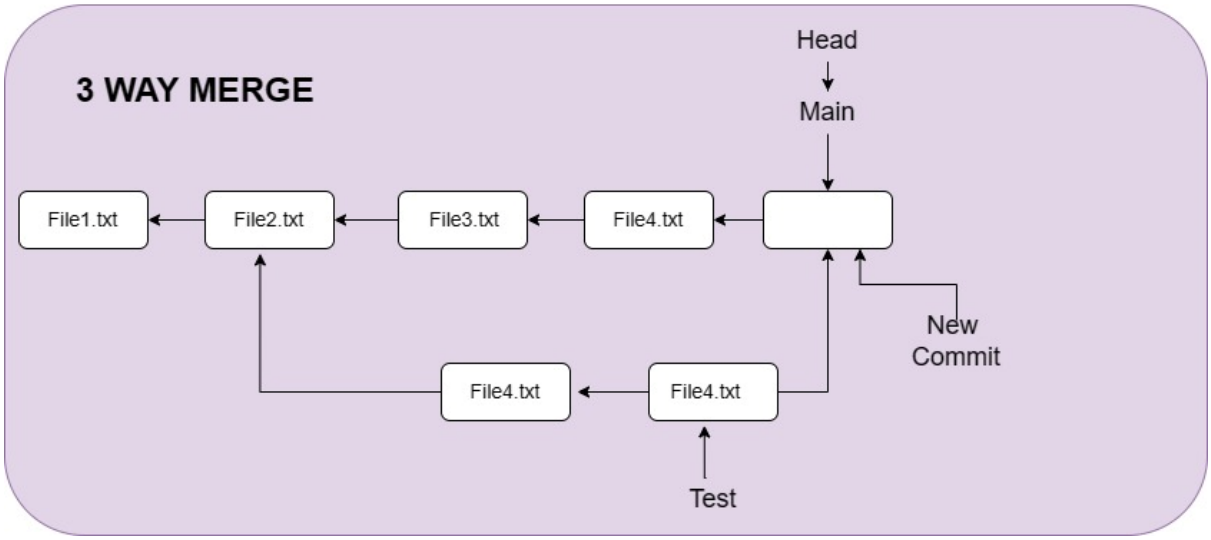
Assignment- 2 Git Merge

a) Design Schema





b) Explanation

Three-way merge **happens** when the tip of the current branch ("test" in our case) is a direct ancestor of the target branch ("main" in our case). Here instead of merging the two branches git simply moves the current branch tip up to the target branch tip. Let us look at an example of a 3-way merge. In this example, the **test branch** is two commits ahead of the *Master* branch. Before we merge it with Master, let us say we have added an additional commit to the Master.

Due to the commit performed on the Master branch, both our branches ***Master* and test** are now diverged. This means we have some changes in the ***Master* branch** that is not present in the **test branch.** If we perform a merge in this case, Git cannot move the master pointer towards the test branch. If git simply moves the Master pointer to the test pointer, then the latest commit performed on the *Master* branch will be lost.

So how do we perform a merge if the branches are diverged? When we want to merge the branches that are diverged, Git creates a **new commit** (Merge Commit) and combines the changes of these two branches

c) User Case

1) First create repo having name 3waymerge and initialize it.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice
$ mkdir 3waymerge

bhush@Bhushan MINGW64 /c/DevOps-Practice
$ cd 3waymerge/

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge
$ git init
Initialized empty Git repository in C:/DevOps-Practice/3waymerge/.git/

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ ls

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ ls -la
total 4
drwxr-xr-x 1 bhush 197609 0 Mar 14 22:58 ./
drwxr-xr-x 1 bhush 197609 0 Mar 14 22:57 ../
drwxr-xr-x 1 bhush 197609 0 Mar 14 22:58 .git/
```

2) Now I create files File1.txt and File2.txt. Check the status it is untracked. Now add the File1.txt in staging area followed by File2.txt and commit it.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ vi File1.txt

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        File1.txt

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

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git add File1.txt
warning: in the working copy of 'File1.txt', LF will be replaced by CRLF the next time Git touc
hes it

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git commit -m "File1.txt is committed!"
[master (root-commit) 6be47f8] File1.txt is committed!
 1 file changed, 1 insertion(+)
 create mode 100644 File1.txt
```

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ vi File2.txt

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        File2.txt

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

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git add File2.txt
warning: in the working copy of 'File2.txt', LF will be replaced by CRLF the next time Git touc
hes it

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git commit -m "File2.txt is created!"
[master e9cf45a] File2.txt is created!
 1 file changed, 1 insertion(+)
 create mode 100644 File2.txt
```

3) Now Create branch having name test and check the all the branches. * is in-front of the main (master) it means that we are on the master branch.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git branch test

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git branch
* master
  test
```

4) Now switch the branch test and check the branch * is in-front of test it means that we are on test branch.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git checkout test
Switched to branch 'test'

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git branch
  master
* test
```

5) Now create files File5.txt and File6.txt. Check the status it is untracked. Now add the File5.txt in staging area followed by File6.txt and commit it.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ vi File5.txt

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git status
On branch test
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        File5.txt

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

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git add File5.txt
warning: in the working copy of 'File5.txt', LF will be replaced by CRLF the next time Git touc
hes it

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git commit -m "File5.txt is committed!"
[test ab90a42] File5.txt is committed!
 1 file changed, 1 insertion(+)
 create mode 100644 File5.txt
```

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ vi File6.txt

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git status
On branch test
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        File6.txt

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

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git add File6.txt
warning: in the working copy of 'File6.txt', LF will be replaced by CRLF the next time Git touc
hes it

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git commit -m "File6.txt is committed!"
[test c4cde34] File6.txt is committed!
 1 file changed, 1 insertion(+)
 create mode 100644 File6.txt
```

6) Now if you see the one-line graph then it clearly mention that 2 commits from the master (main) branch and 2 from the test branch.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git log --graph --pretty=oneline
* c4cde34965b7782377db0b7c7f3b2830f4d8d0e3 (HEAD -> test) File6.txt is committed!
* ab90a42f6f17d91ed5fbd87ade96cd8a68b536f6 File5.txt is committed!
* e9cf45ab4a3196e7de09d4507d7406269613fb8 (master) File2.txt is created!
* 6be47f843581f88f0013a30d22e54a95679617e0 File1.txt is committed!
```

7) Now switch the branch master(Main) and check the branch * is in-front of master it means that we are on master branch.

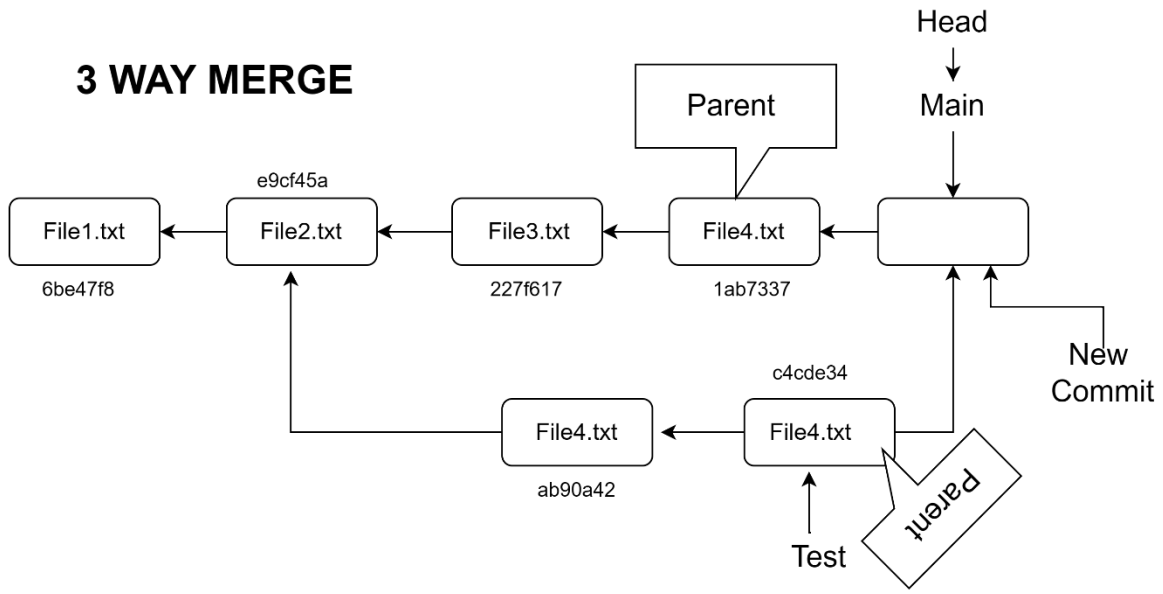
```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (test)
$ git checkout master
Switched to branch 'master'

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git branch
* master
  test
```

8) Now create files File3.txt and File3.txt. Check the status it is untracked. Now add the File3.txt in staging area followed by File4.txt and commit it.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ vi File3.txt

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        File3.txt

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

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git add File3.txt
warning: in the working copy of 'File3.txt', LF will be replaced by CRLF the next time Git touc
hes it

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git commit -m "File3.txt is committed!"
[master 227f617] File3.txt is committed!
 1 file changed, 1 insertion(+)
 create mode 100644 File3.txt
```

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ vi File4.txt

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        File4.txt

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

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git add File4.txt
warning: in the working copy of 'File4.txt', LF will be replaced by CRLF the next time Git touc
hes it

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git commit -m "File4.txt is committed!"
[master 1ab7337] File4.txt is committed!
 1 file changed, 1 insertion(+)
 create mode 100644 File4.txt
```

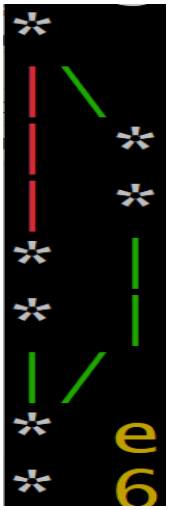
9) Now **new commit** is created with the help of **git merge** command which is made from 2 parent commit one from the master branch and another from the test branch.




```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git merge test
Merge made by the 'ort' strategy.
 File5.txt | 1 +
 File6.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 File5.txt
 create mode 100644 File6.txt
```

10) If you check the graph then it will show the same way as show in above diagram.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git log --graph --pretty=oneline
*   670066c2897a2fda92ad1eeb07cc2936577caf39 (HEAD -> master) Merge branch 'test'
|\
| * c4cde34965b7782377db0b7c7f3b2830f4d8d0e3 (test) File6.txt is committed!
| * ab90a42f6f17d91ed5fbd87ade96cd8a68b536f6 File5.txt is committed!
* | 1ab7337c0dbe7c86a581db4b624a42147bed1d1d File4.txt is committed!
* | 227f617d59411c5d5b5224aa52287909bcef4d4f File3.txt is committed!
|/
* e9cf45ab4a3196e7de09d47507d7406269613fb8 File2.txt is created!
* 6be47f843581f88f0013a30d22e54a95679617e0 File1.txt is committed!
```



11) Check the content type information we use cat-file command which is use to Provide content or type and size information for repository objects

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git cat-file  670066c2 -p
tree 55c98e274d56467eea720be995666729e944f33d
parent 1ab7337c0dbe7c86a581db4b624a42147bed1d1d
parent c4cde34965b7782377db0b7c7f3b2830f4d8d0e3
author rutujatopre <rutujamm27@gmail.com> 1678818283 +0530
committer rutujatopre <rutujamm27@gmail.com> 1678818283 +0530

Merge branch 'test'
```

12) If we want to delete the branch test then use -d option.

```
bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git branch test -d
Deleted branch test (was c4cde34).

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git log --graph --pretty=oneline
*   670066c2897a2fda92ad1eeb07cc2936577caf39 (HEAD -> master) Merge branch 'test'
|\
| * c4cde34965b7782377db0b7c7f3b2830f4d8d0e3 File6.txt is committed!
| * ab90a42f6f17d91ed5fbd87ade96cd8a68b536f6 File5.txt is committed!
* | 1ab7337c0dbe7c86a581db4b624a42147bed1d1d File4.txt is committed!
* | 227f617d59411c5d5b5224aa52287909bcef4d4f File3.txt is committed!
|/
* e9cf45ab4a3196e7de09d47507d7406269613fb8 File2.txt is created!
* 6be47f843581f88f0013a30d22e54a95679617e0 File1.txt is committed!

bhush@Bhushan MINGW64 /c/DevOps-Practice/3waymerge (master)
$ git branch
* master
```