

16.14Apr.GitAdv(Rebase, Revert, Reset, Linux Setups)

14 April 2023 07:19 PM

```
$ git log
```

Untracked Files

1. Repo1 --> File1.txt(b2)

```
$ git checkout master
```

```
$ git branch -b b2
```

```
$ git log --oneline
```

OR

```
$ git log --pretty=oneline
```

```
CommitID1 (HEAD, origin/branchName,branchName) "Commit Message"
```

```
CommitID2 "Commit Message"
```

```
CommitID3 "Commit Message"
```

```
CommitID4 "Commit Message"
```

```
CommitID "Commit Message"
```

```
CommitID "Commit Message"
```

```
CommitID "Commit Message"
```

```
CommitID "Commit Message"
```

```
CommitID "Commit Message"
```

```
$ cat .git/HEAD # Unix
```

```
CommitID1
```

```
$ type .git/COMMIT_EDITMSG # Windows
```

Repo1

LOCAL	Remote
B1	A1
B2	B2
B3	B3

Setting Up Upstream/Downstream

Open Source

OS: Ubuntu

App1: Firefox(FD1 --> Changes)

App2: VSCode

App3: Java

FF --> Upstream --> Ubuntu
Ubuntu --> Downstream --> FF

B1 --> Downstream --> BA1

LocalB1 --> Upstream --> RemoteB1

Upstream: if data is flowing from local to origin
Downstream: if data is flowing from origin to local

Delete Changes from Working Directory Area

\$ git checkout -- B3.txt

\$ git checkout -- *.txt

\$ git checkout -- .

Git Revert

R1:

C1 C2 C3 C4(Wrong Commit) C5

➤ On Previous commit we set HEAD and new Commit Message to It.

1. C1---->C2---->C3---->C4(Wrong Commit)
2. C1---->C2---->C3<----C4(Wrong Commit) \$ git revert C4-CommitID
3. C1---->C2---->C3<----C4(Wrong Commit)
|
C5(Merge Commit Message)

\$ git revert C4-CommitID

When you don't want to lose your Wrong/Bad Commits

If you pushed unwanted commit from local then this will help you to get back HEAD to the normal commit.

Dual Boot(Linux / Windows) = 8/10 hrs

VirtualBox

Linux ---> Testing

|-----> SysOps

Git Reset (Undo)

1. Soft
2. Hard
3. Mixed
4. Merge
5. Keep

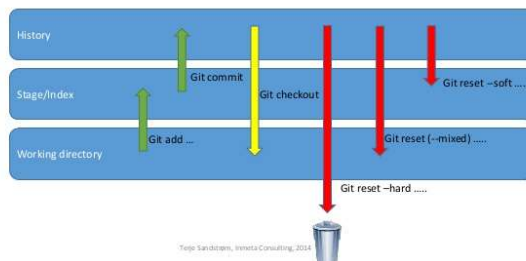
C:

D:

C1---> C2 ---> C3 --> C4 --> C5

\$ git reset --soft

Git tree movements visualized



- Soft: Uncommit changes, changes are left in Stage Area
- Mixed(default): Uncommit changes, changes are left in Working Dir. Area
- Hard: Uncommit changes+ Un stage +Delete Working Dir. changes, Nothing left

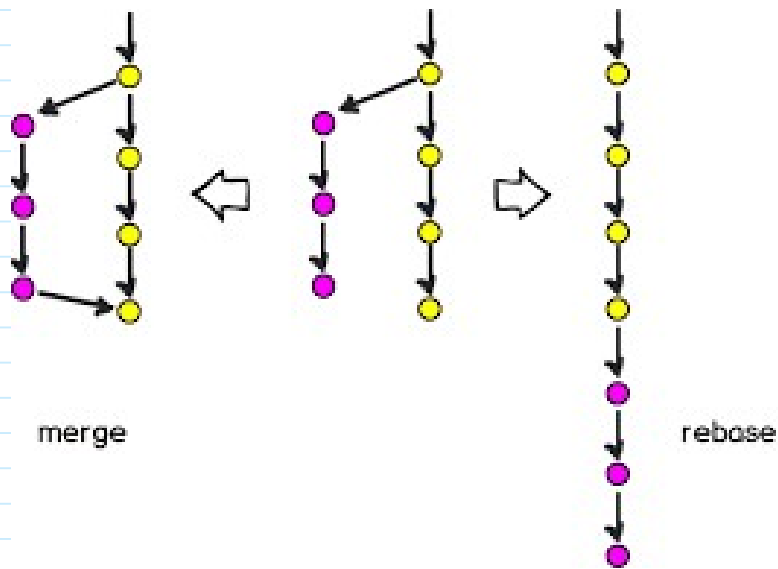
\$ git reset CommitID

Git Rebase

- Process of Moving/Combining a sequence commits to a **new** BASE commit.

\$ git rebase FromBranchName ToBranchName

C1 --> C2 --> C3 --> C4 --> C5 --> C6 --> C7 --> C8 --> C9 --> C10



R: It rewrites history by creating new commits for each commit in source branch

M: It integrates changes while preserving the ancestry of each commit history

M: Commits remain reachable from the branch(pink)

R: Commits once reachable to target branch(yellow) then they are no longer reachable to source branch(pink).

Rebase = Merge + Commits

Merge Strategies:

1. Fast Forward
2. Non- Fast Forward
3. Recursive
4. Octopus
5. Ours
6. Subtree

1. Windows ----> VirtualBox ----> Kali

2. Kali(Zoom App)

3. Kali/Window(Dual Boot)

4. Cloud

