Agenda :

→ Basics of GIT ✓

→ Branching

→ Merge

→ Rebase

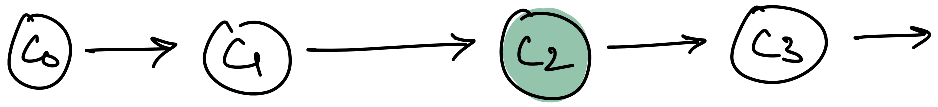
→ Merge (vs) Rebase

→ HEAD pointer.

⇒ GIT Commit.

↓

Distributed VCS.

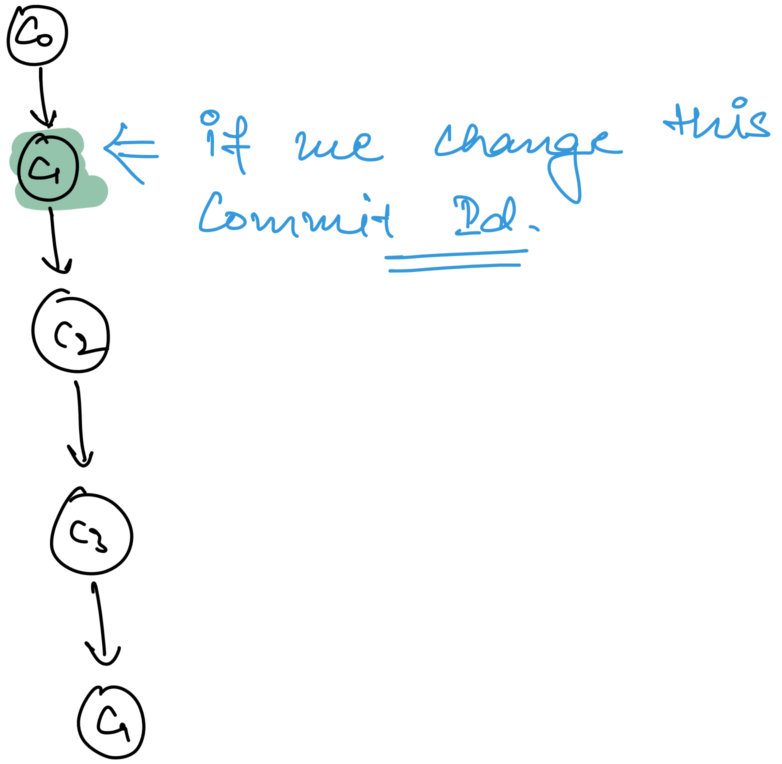


Every commit stores the changes | delta wrt previous commit.

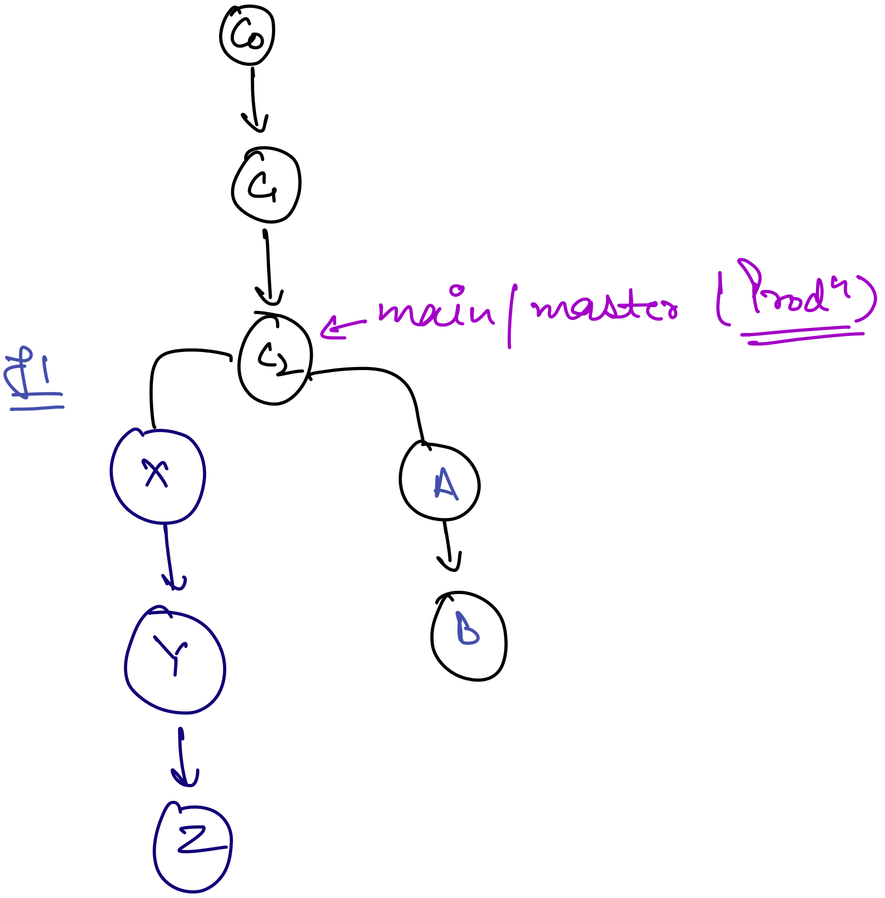
Every commit is Immutable.

↳ message
Id
timestamp
author

Commit Id = hash(authordetails, timestamp, ...
List<Previous commit Ids>)



Branching.

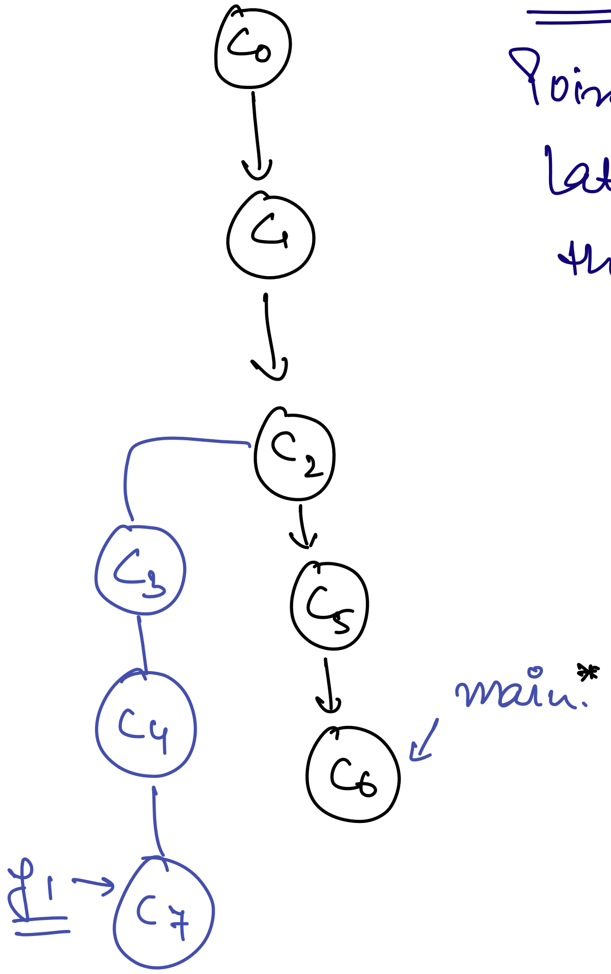


⇒ To work on a new feature, we create a separate branch to work parallely.

Merge.

Branch name:

Pointer pointing to the latest commit of that branch.

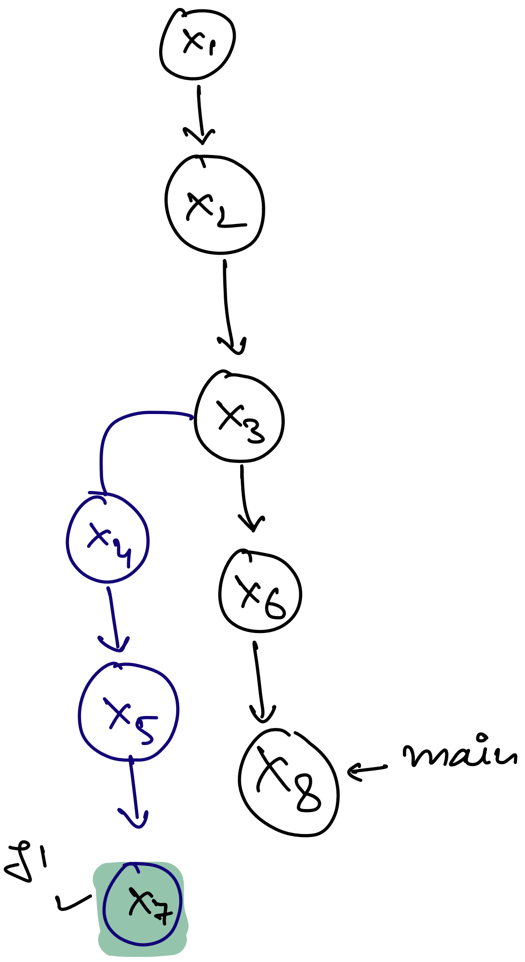


⇒ Once the feature has been implemented & tested completely, we'll add our branch changes to the Main branch

⇒ Merge

GIT Merge.

Command used to bring changes from one branch to another branch.



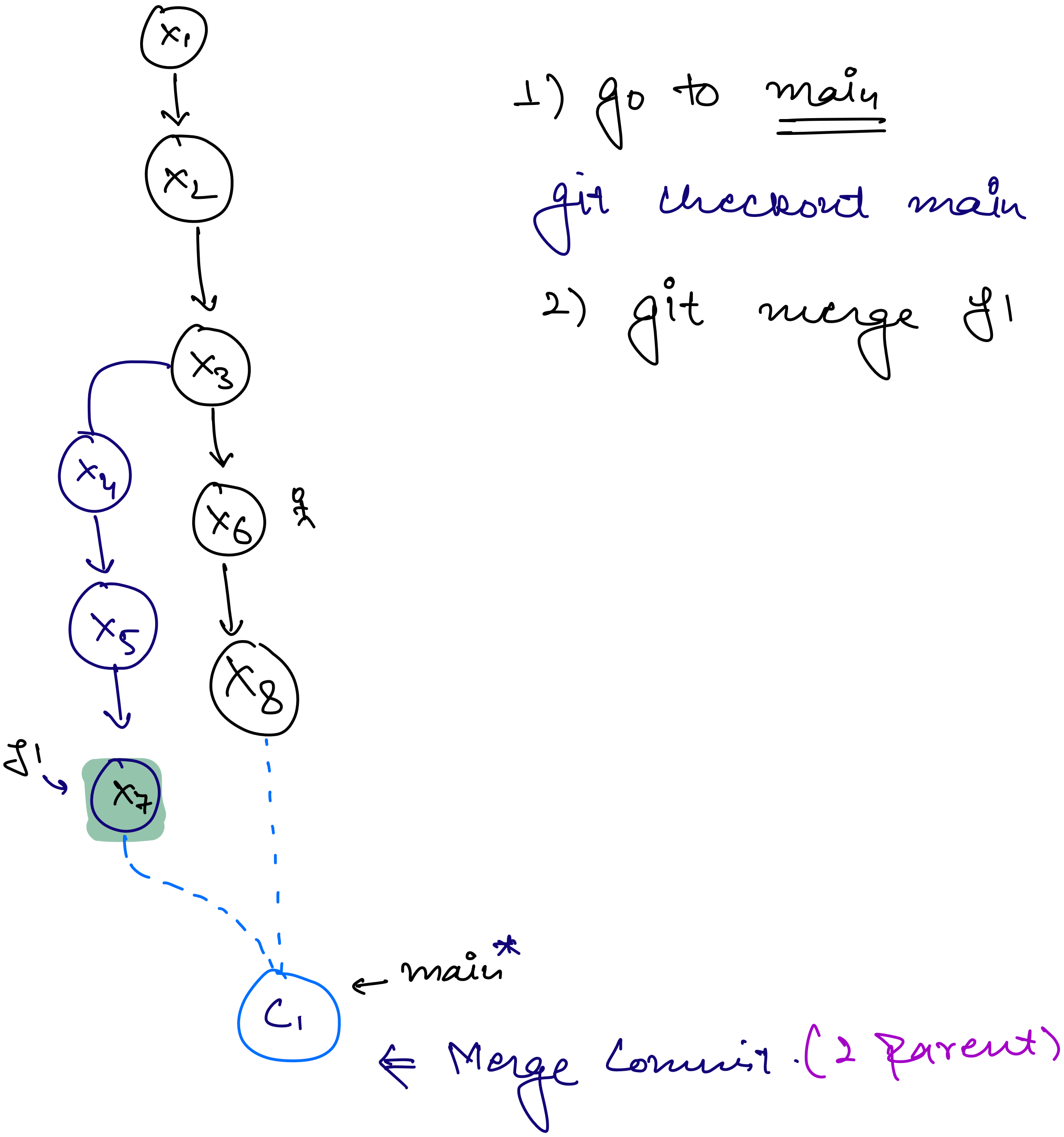
Bringing our changes (x_4, x_5, x_7) to the main. branch.

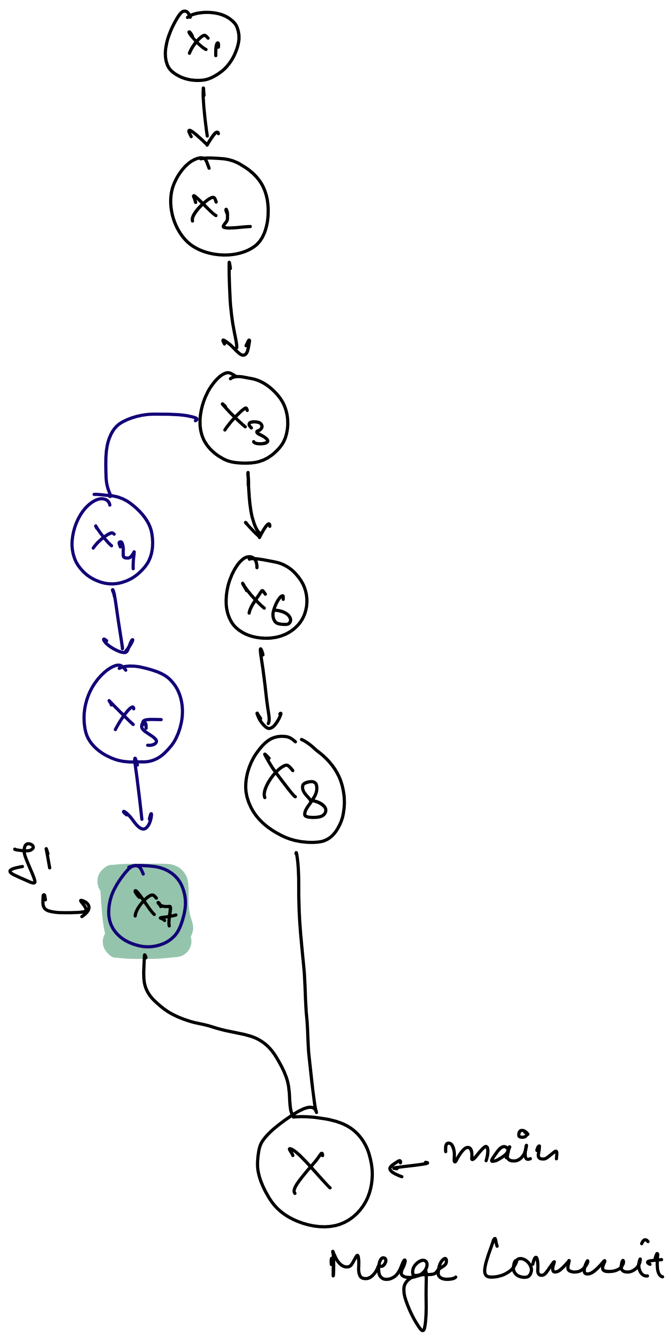
⇒ Merge f1 branch into Main.

1) go to main

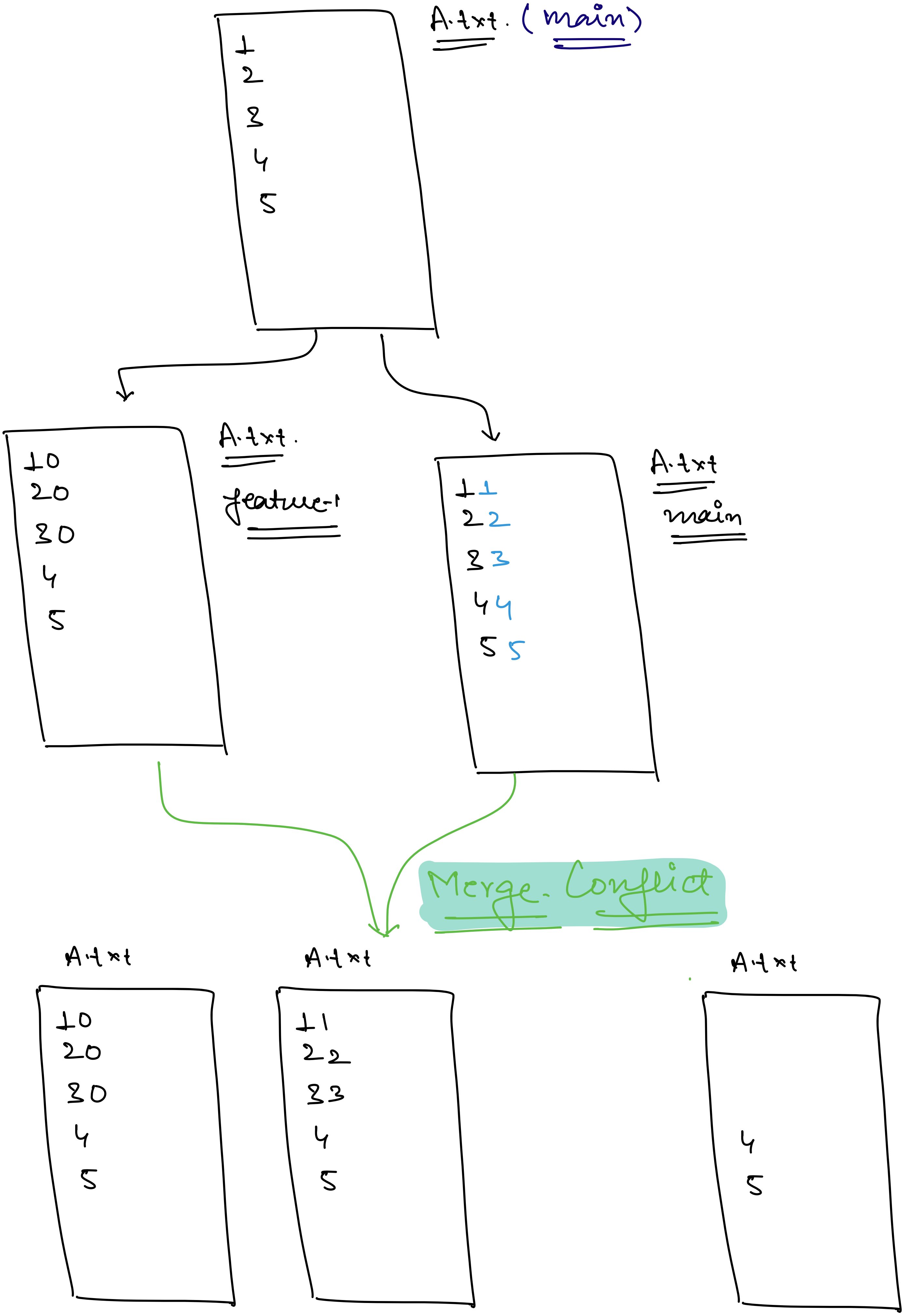
git checkout main

2) git merge f1





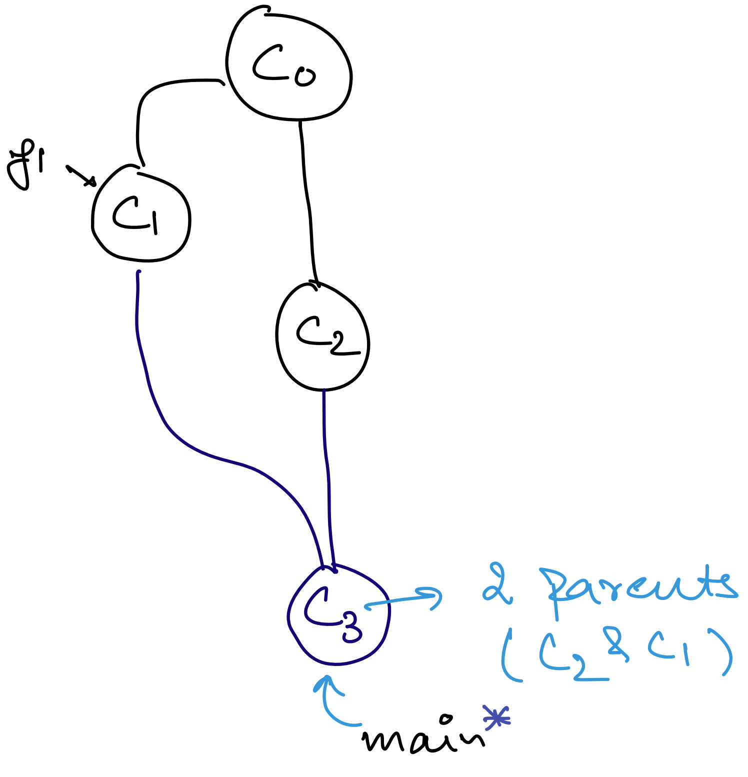
⇒ Merge


A.txt. (main)
1
2
3
4
5
A.txt.
feature-1
10
20
30
4
5
A.txt
main
11
22
33
44
55
Merge. Conflict
A.txt
10
20
30
4
5
A.txt
11
22
33
4
5
A.txt
4
5

A.txt

```
10
20
30
11
22
33
|
|
```

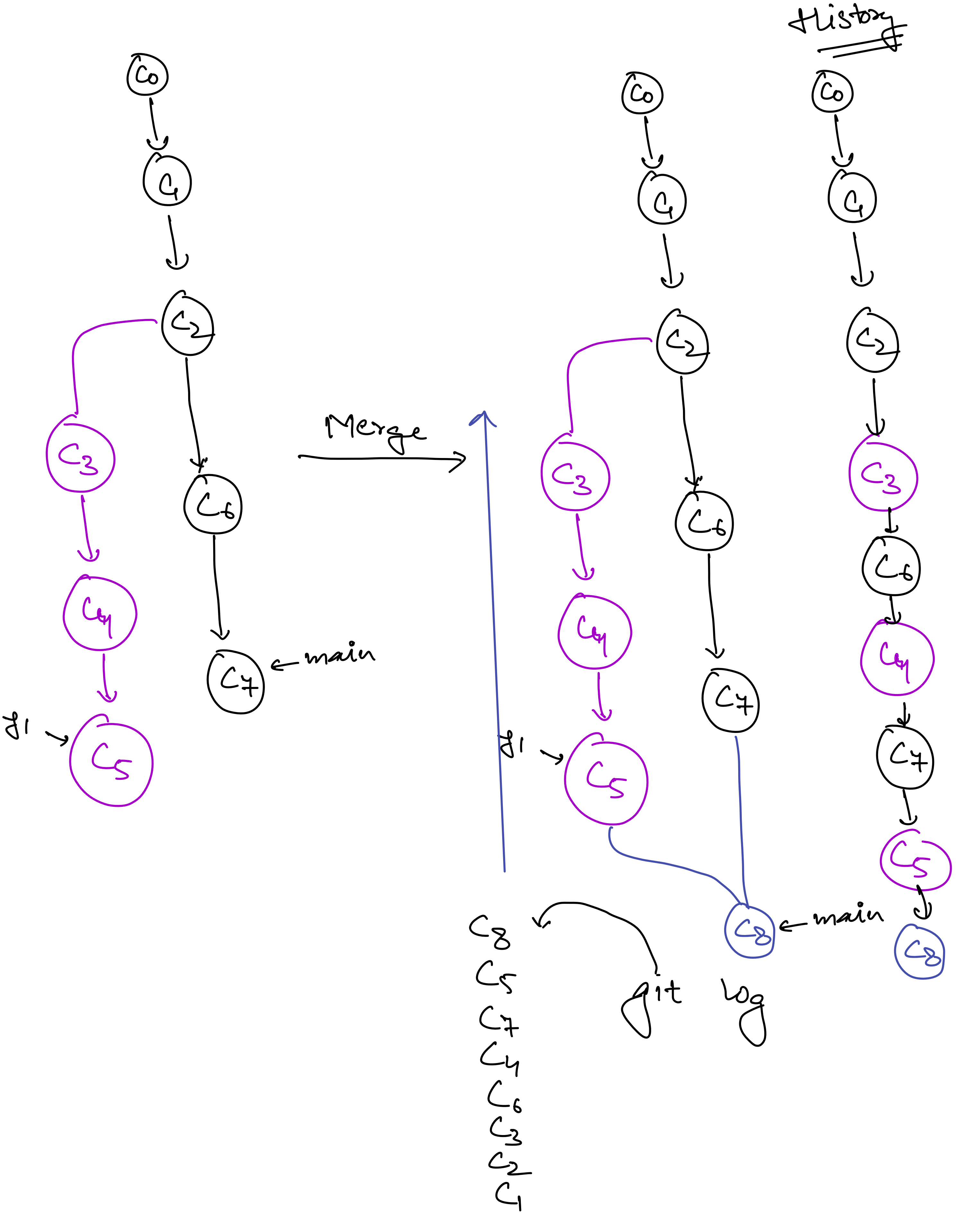
⇒ We'll have to resolve conflicts manually.



⇒ git log

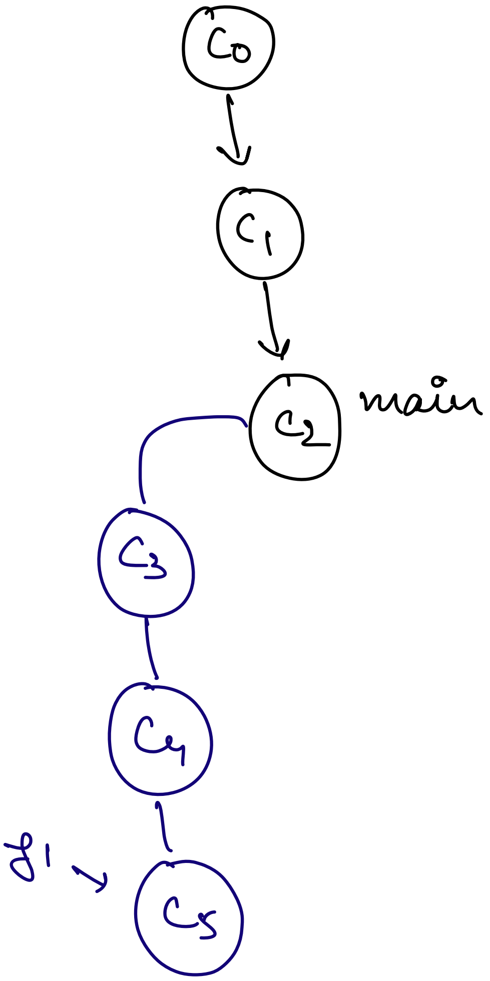
History of commits
- C3
- C2
- C1
- C0

git checkout -b f2 ⇒ Create branch f2 & Checkout to f2.



⇒ In the merge commit, History is getting overlapped from different branches.

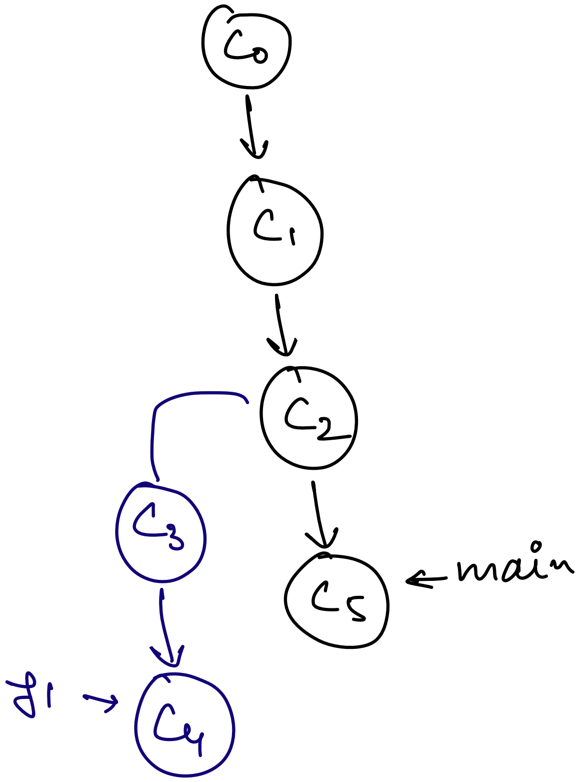
⇒ Because of overlapping, debugging can be difficult.



Merge f1 into main.

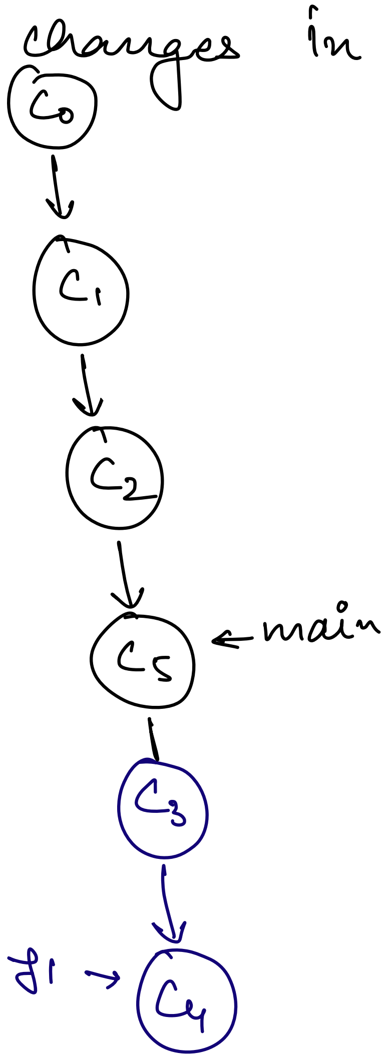
⇒ fast forward merge.

⇒ No merge commit will be created.



⇒ Rebase.

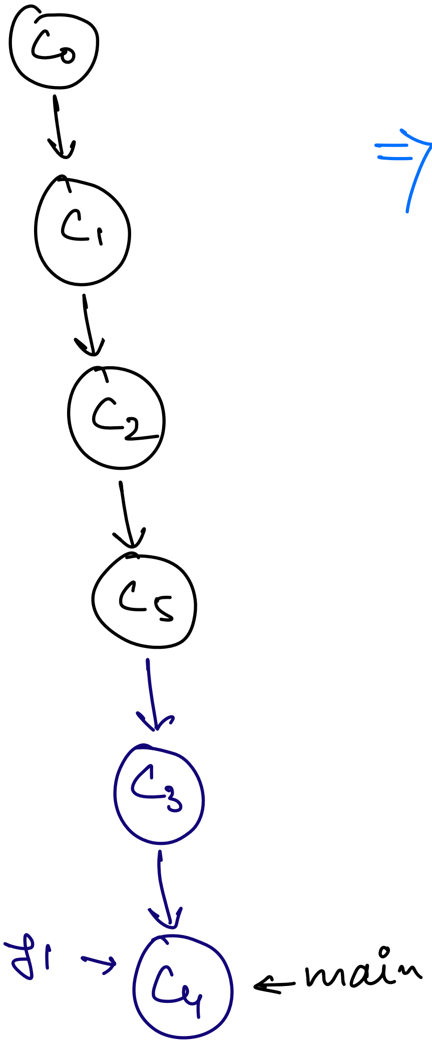
GIT Rebase.: Allows us to remap all the changes that we have done in a new branch below the latest changes in the other branch.



1) git checkout f1.
2) git rebase main

⇒ No change in the main branch.

#



1 ⇒ Rebase.

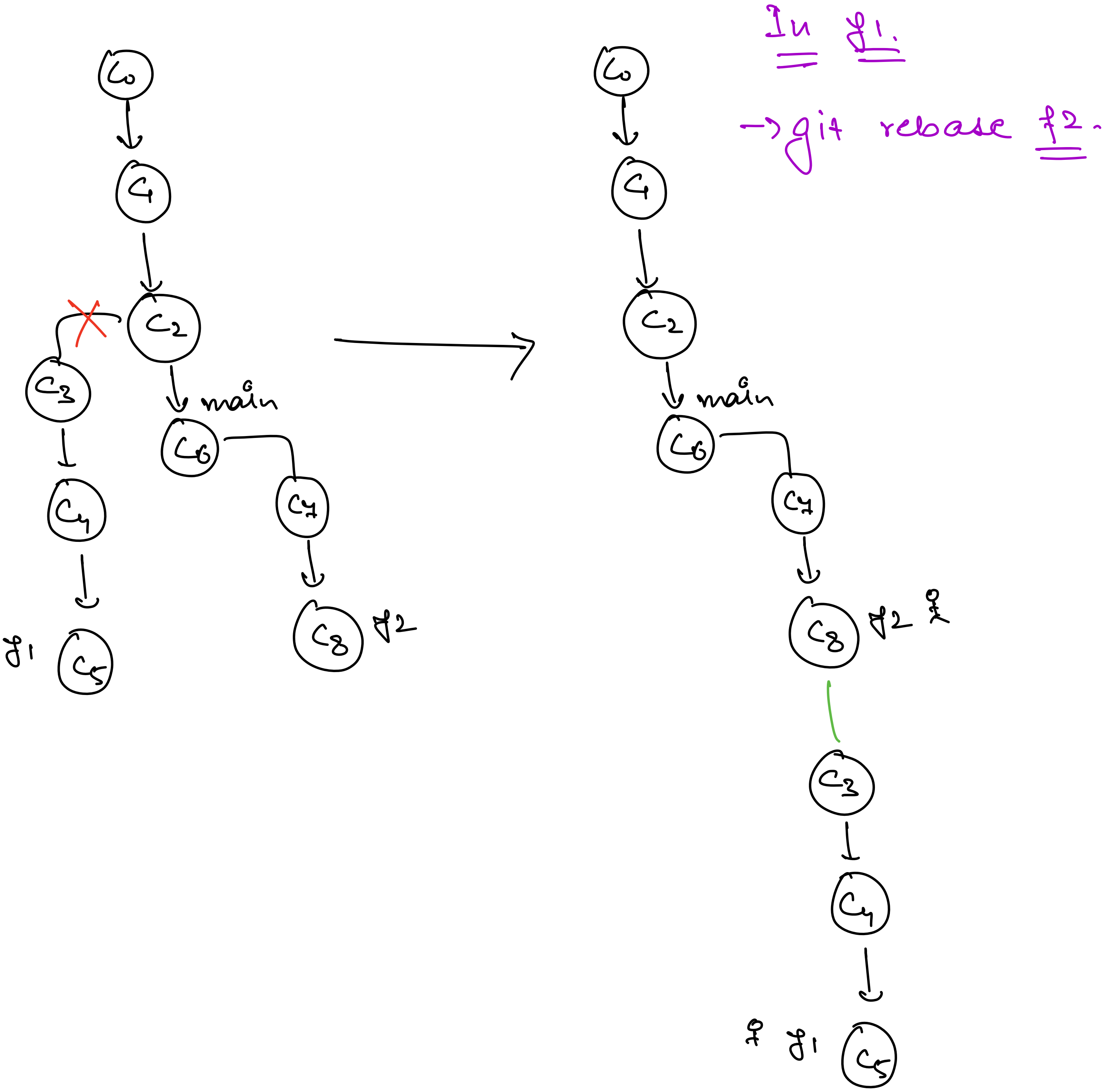
+

2 Merge.

git log

Rebase + Merge ⇒ Cleaner History.

Merge vs Rebase + Merge ✓



In f1:

→ git rebase f2.

HEAD.

↳ Special pointer in GIT that always points to the latest commit.

↓

Commit on which the new commit will be added.

Detached HEAD.

When HEAD pointer points to commit instead of pointing to a branch