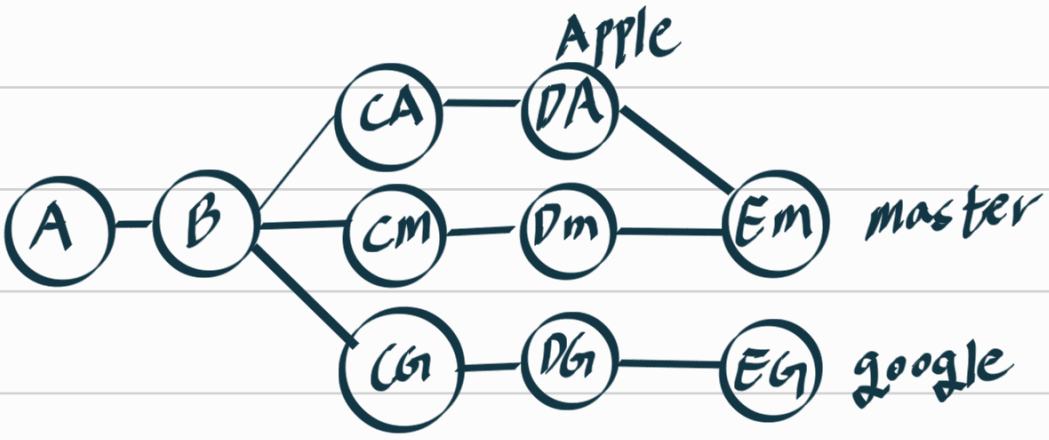
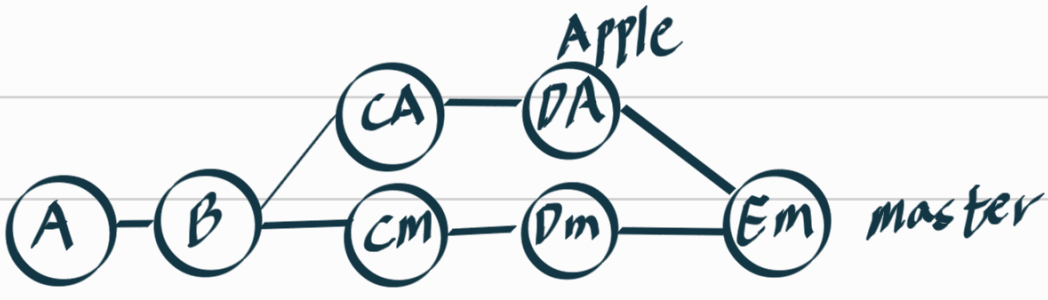
Branch



기존 버전(부터)이 같은 여러 버전들을 효율적으로 관리하기 위한 시스템

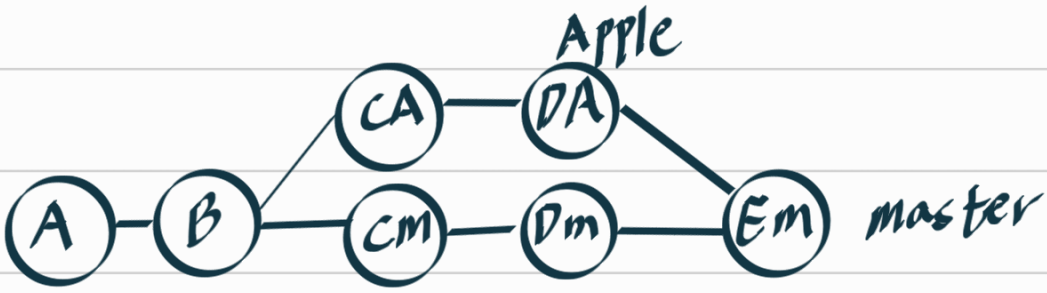
$ git branch 브랜치 명 : 새로운 브랜치 생성
$ git checkout 브랜치 명 : 해당 브랜치로 HEAD를 옮김

브랜치 병합



$ git checkout master
$ git merge Apple : master 브랜치에 Apple을 병합
↳ DA 버전에 존재하는 Iphone.txt 와
DM 버전에 존재한 Galaxy.txt 가
EM 버전에 둘다 존재한다.

같은 문서의 다른 위치를 수정한다면?

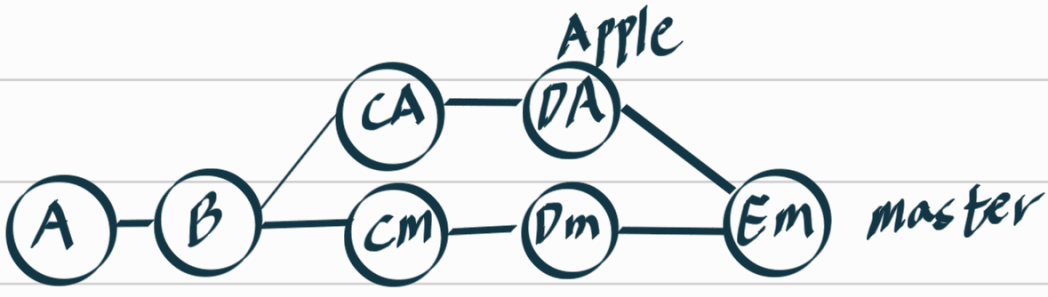


DA와 DM에 공통으로 존재하는 work.txt의
서로 다른 부분을 수정

↓ Merge

수정한 것이 모두 적용된 채 EM에 존재

같은 문서의 같은 위치를 수정한다면?



DA는 work.txt의 첫줄을 수정한 파일을 갖고있고
DM도 work.txt의 첫줄을 수정한 파일을 갖고있다

↓ merge

Conflict (충돌 발생)

↓ vim work.txt로 충돌 수정 후 커밋

해결

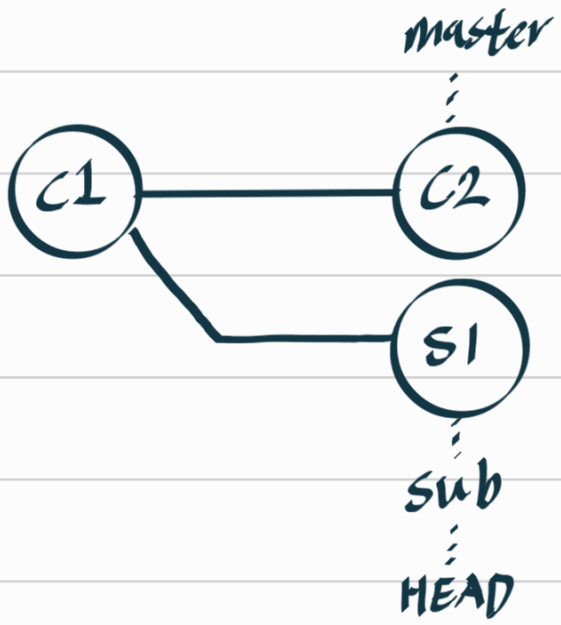
브랜치 삭제

$ git branch -d Apple

↳ 완전히 저장소에서 없어진 것이 아님*

Apple branch를 재생성하면 이전 내용이 복구됨

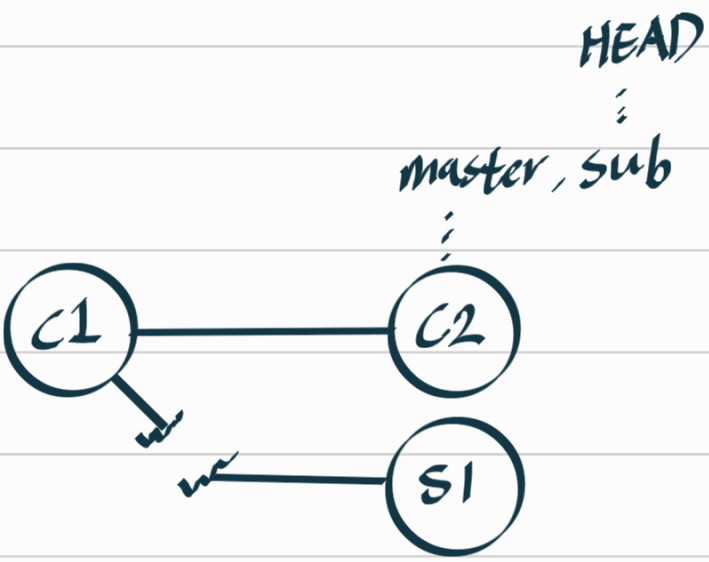
브랜치와 reset



(sub가 HEAD인 상태에서)

$ git reset C2 커밋해시 사용시

↓



sub브랜치의 최신버전이 C22 번경 / S1은 삭제

Git Stash

파일수정 후 커밋전에 급히 다른 파일을 커밋해야 할경우
$ git stash를 사용하면 파일을 감출수 있다.

$ git stash pop으로 다시 복원할수 있다.