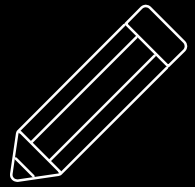
ECE20023, Spring 2021



오픈소스 소프트웨어 실습

12

Git

VCS (Version Control System)

이력 수정

- 마지막 commit 메세지 수정

```
walab-HGU:~/hello:> git commit --amend
```

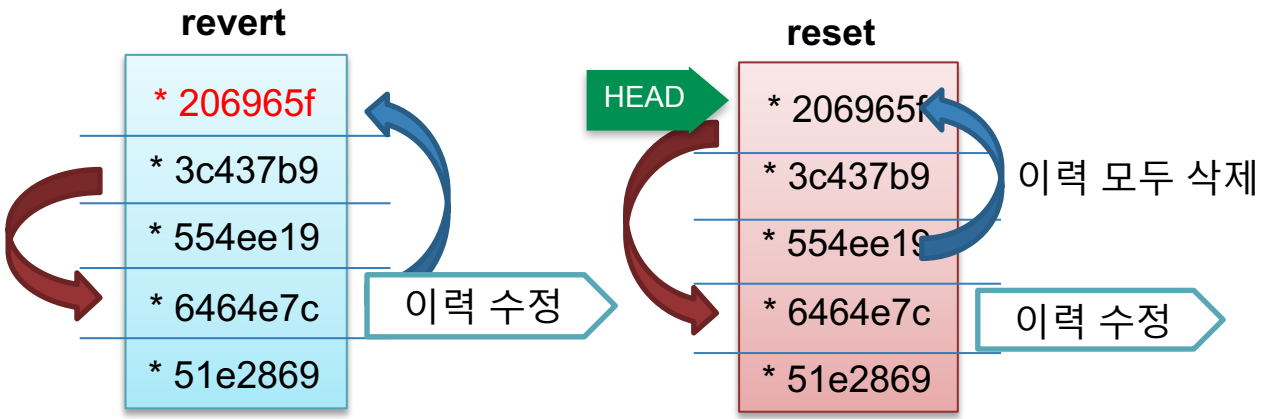
- Modified 상태를 unmodified 상태로 변경 (WD)

```
walab-HGU:~/hello:> git checkout <file1>
```

- git reset, git revert

revert vs reset

- 특정 커밋으로 되돌아 갈 수 있음. 취소 효과



reset

- Commit 취소
- 특정 커밋으로 돌아감으로 취소 작업
- 특정 커밋 이후의 버전들을 히스토리에서 삭제

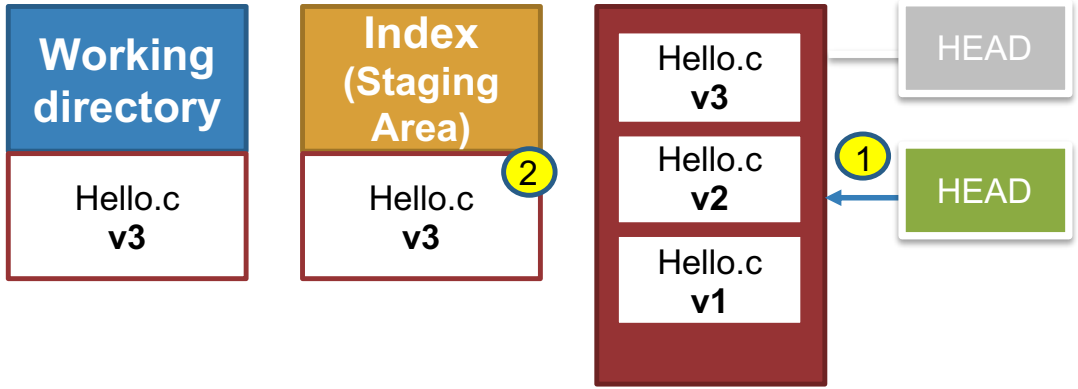
```
$ git reset [option] commit_checksum
```

- Option
  - --soft
  - --mixed
  - --hard

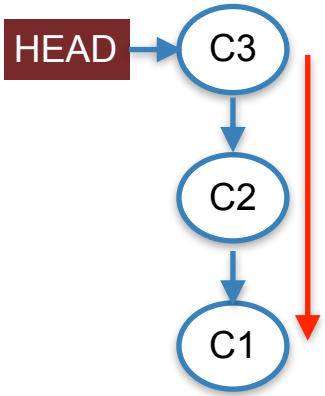
git reset --soft

- HEAD를 특정 커밋으로 이동
- WD파일 보존, 해당 파일은 staged로 이동
- Commit 하면 원래 상태로 복원 가능

```
$ git reset --soft HEAD~
```



git reset --soft

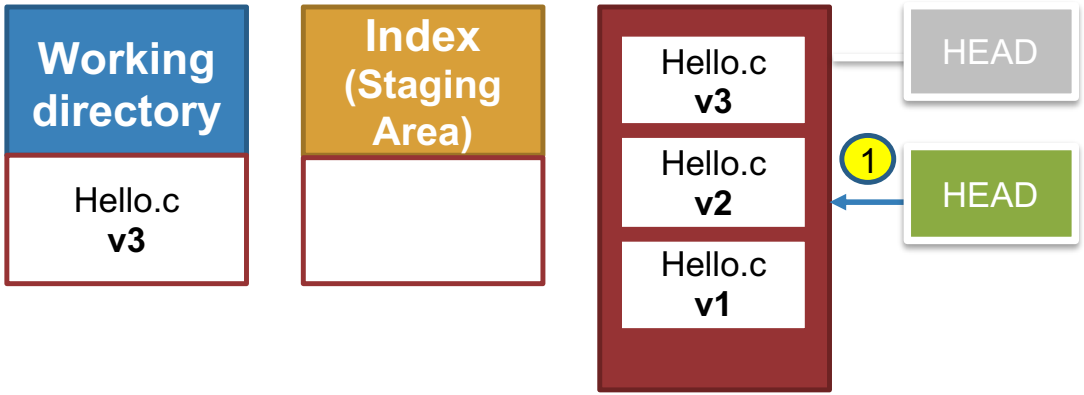



```
walab-HGU:~/lab7> touch test.txt
walab-HGU:~/lab7> git add test.txt
walab-HGU:~/lab7> git commit -m "C1"
walab-HGU:~/lab7> vim test.txt
walab-HGU:~/lab7> git commit -am "C2"
walab-HGU:~/lab7> vim test.txt
walab-HGU:~/lab7> git commit -am "C3"
walab-HGU:~/lab7> git log --oneline
dcf2782 (HEAD -> master) C3
edc8683 C2
a977e1c C1
walab-HGU:~/lab7> git reset --soft a977e1c
walab-HGU:~/lab7> git log --oneline
a977e1c (HEAD -> master) C1
walab-HGU:~/lab7> git status
```

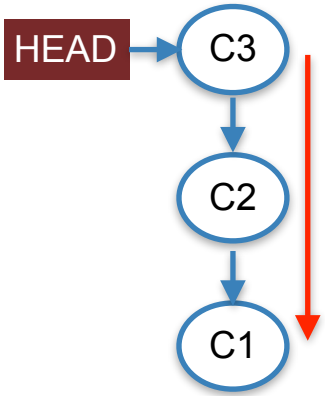
git reset --mixed

- Default 옵션
- WD파일 보존, 해당 파일 unstated, HEAD 이동

```
$ git reset --mixed HEAD~
$ git reset HEAD~
```



git reset --mixed

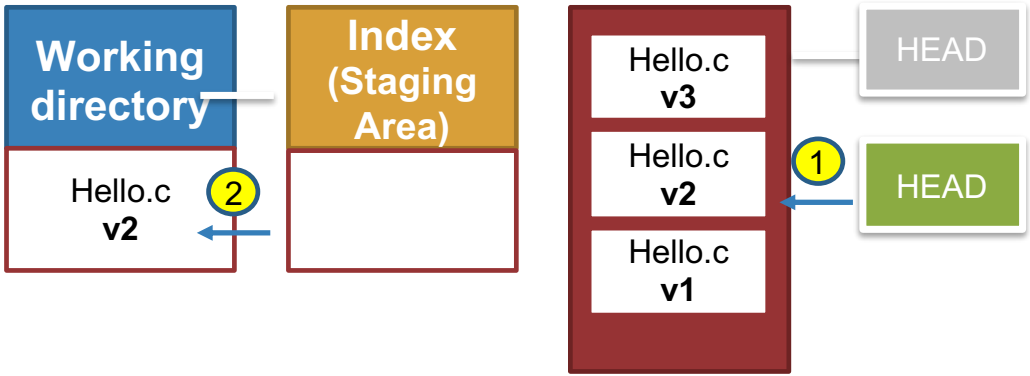



```
walab-HGU:~/lab7> git commit -m "C2"
walab-HGU:~/lab7> vim test.txt
walab-HGU:~/lab7> git commit -am "C3"
walab-HGU:~/lab7> git log --oneline
408214c (HEAD -> master) C3
6e793c0 C2
a977e1c C1
walab-HGU:~/lab7> git reset --mixed a977e1c
walab-HGU:~/lab7> git status
walab-HGU:~/lab7> git log --oneline
a977e1c (HEAD -> master) C1
```

git reset --hard

- WD파일 변경사항 삭제, 해당 파일 unstaged, HEAD 이동

```
$ git reset --hard HEAD~
```

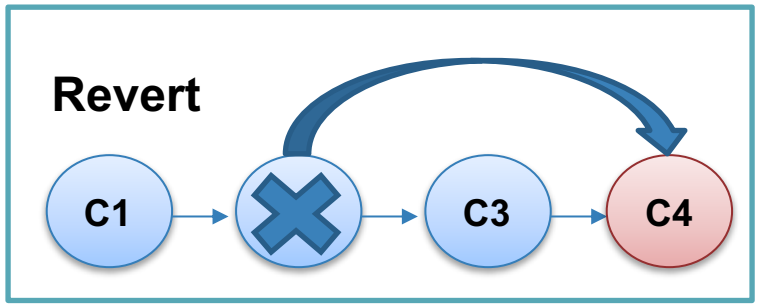


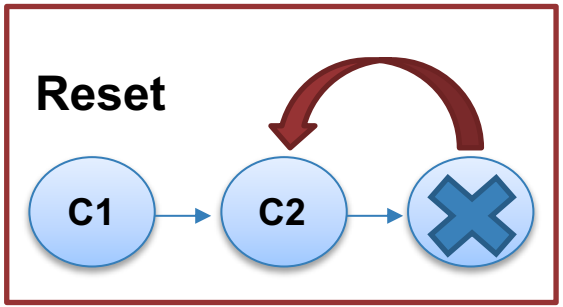
git revert

- commit된 스냅샷을 취소하는 명령

```
$ git revert HEAD~
```

- 커밋 이력에서 취소하기 원하는 커밋에 의해 변경된 내용을 취소하기 위한 방법을 찾고 그 결과를 새로운 커밋으로 추가함
- 현재 커밋 이력을 삭제하지 않음





git rm

- Untracked files

```
$ rm sample.txt
```

- Tracked files
  - git 저장소 + 로컬디렉터리 모두 삭제

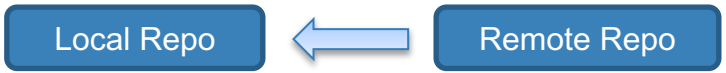
```
$ git rm sample.txt
```

  - git 저장소 삭제, 로컬디렉터리 삭제하지 않음

```
$ git rm --cached sample.txt
```

git clone

- 원격저장소에서 로컬 저장소로 복제



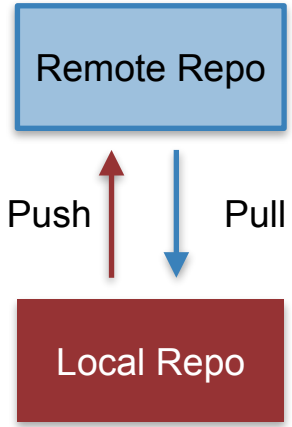

```
$ git clone <remote repository URL>
$ git clone <remote repository URL> <new folder>
$ git clone -b <branchname> <remote repository URL>
```

```
walab-HGU:~:> git clone -b gh-pages https://github.com/ahfarmer/calculator.git
walab-HGU:~/calculator:> git branch -a
walab-HGU:~:> git clone https://github.com/ahfarmer/calculator.git
walab-HGU:~/calculator:> git branch -a
```

Remote repo

- 로컬저장소에 원격 저장소를 연결하여 소스를 push 하거나 pull 할 수 있음
- 로컬저장소에 원격저장소 연결/삭제/ 정보보기

```
$ git remote
$ git remote -v
$ git remote -h
$ git remote add origin <remote repository URL>
$ git remote remove origin
$ git remote add calculator <remote repository URL>
```

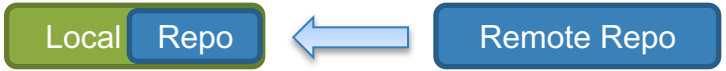


Remote repo

- Push


```
$ git push origin master
```

- Pull

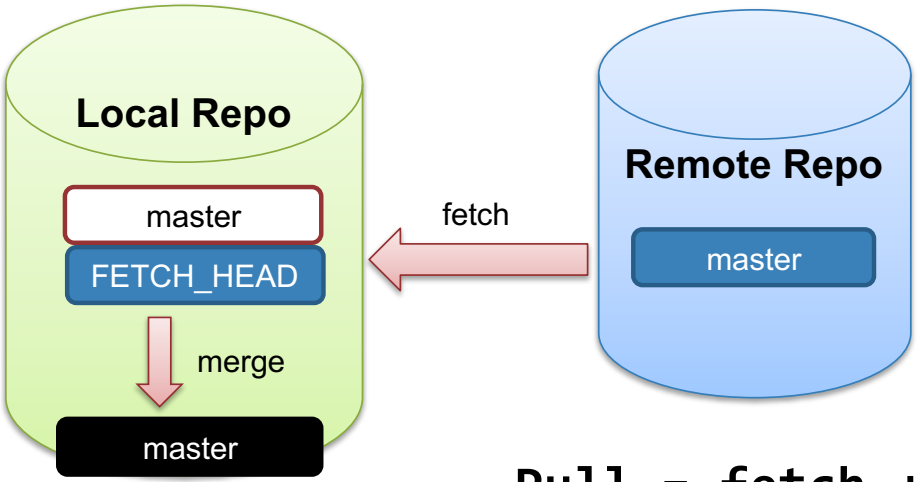



```
$ git pull origin master
```

- fetch


```
$ git fetch
```

Pull vs fetch

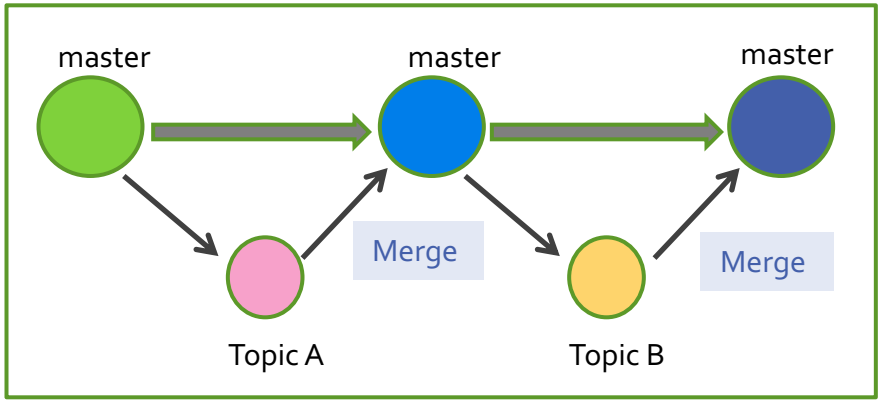



```
Pull = fetch + merge
```

Branch

- 기본 브랜치 : master branch
- 새로운 작업이 발생할 때 브랜치를 생성하여 작업
- 브랜치에서 작업이 완료되면 변경내용 및 이력을 master branch로 병합(merge)

```
$ git branch <new branch>
$ git branch -h
$ git branch -a
$ git branch
$ git branch -d <branchname>
$ git branch -D <branchname>
```

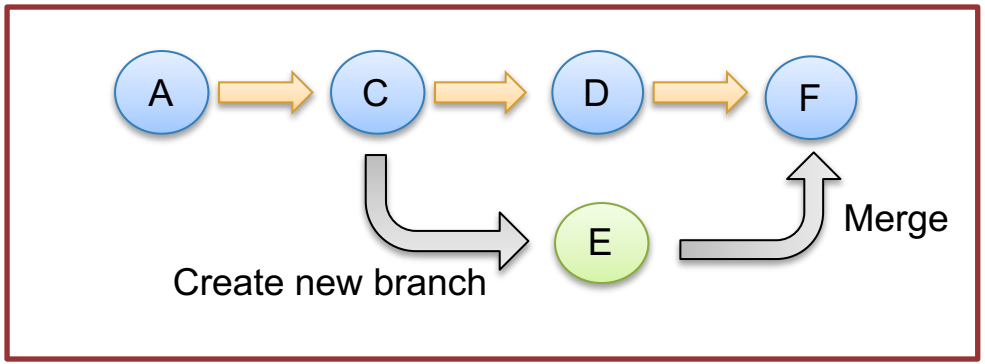


Checkout

- 다른 브랜치 전환할 때 사용

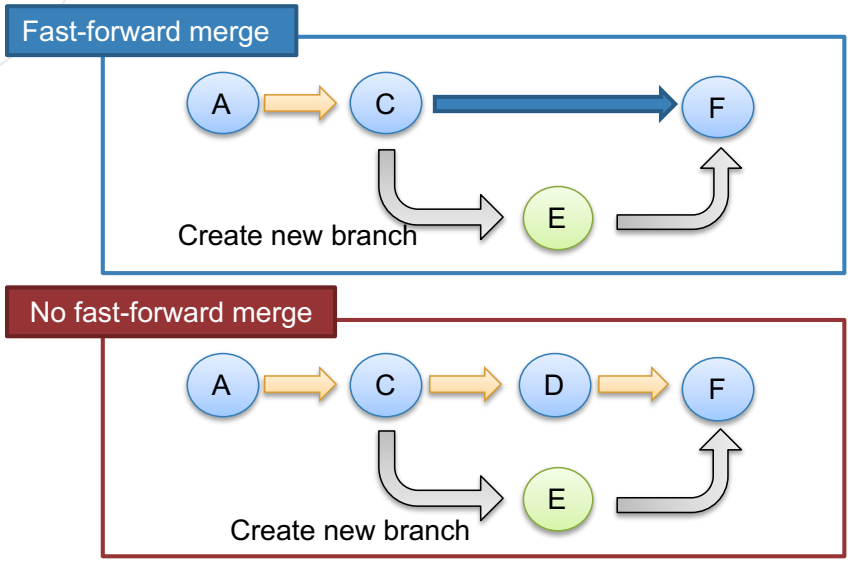
```
$ git checkout <branchname>
$ git checkout -b <new branch>
```

```
walab-HGU:~:> git branch
  master
  test
* test2
```



git merge

- 브랜치와 브래치(master)에 병합하는 작업




```
$ mkdir git1
$ cd git1
$ git init
$ touch hello.c
$ git add hello.c
$ git commit -m “A"
$ vim hello.c
$ git commit -am "C"
$ git checkout -b new
$ vim hello.c
$ git commit -am “E"
$ git checkout master
$ git merge new
```

과제설명

* 제출내용 : 각 문제의 답을 한 파일로 작성하여 **이름_product.pdf** 저장하여 제출할 것

1. 다음 명령어나 개념에 대해 설명하시오.(dynalist document에 작성후 내용을 복사 &붙여넣기 해서 제출해도 됨)
   - git diff
   - git diff --cached
   - git diff commit1 commit2
   - git commit --amend
   - git reset --soft HEAD~
   - git reset HEAD~2
   - git reset --mixed HEAD~
   - git reset --hard HEAD~
   - git rm sample.txt
   - git rm --cached sample.txt
   - git clone “Remote repo URL”
   - git clone “Remote repo URL” abc
   - git clone -b Lab1 “Remote repo URL” al
   - git remote
   - git remote -v
   - git remote add origin “remote repo URL
   - git remote remove origin

2. 다음 Step을 실습해 보세요.

git push

1. Github login
2. Create new repository (**URL 제출**)
3. Remote repo URL 복사
4. Local Repo에서 원격 repo 에 연결(**화면 제출**)
5. git push origin master(**화면 제출**)
6. 원격 repo 확인(**화면 제출**)

git pull

1. 원격 Repo 변경(**웹 commit 리스트 화면 제출**)
2. Local Repo 이력 변경**(git log 화면 제출)**
3. git push origin master(**화면 제출**)
4. 오류 확인 (**화면 제출**)
5. git pull origin master (**화면 제출**)
6. git push origin master (**화면 제출**)

Walab 서버에 접속해서 코딩할 것!

MINI Project 4. 파일 저장 및 검색

제출내용 :

- 본인 github 계정에 만든 mini project repository 주소
- 현재 진행정도 checklist : CRUD/MENU/다중데이터/파일 저장/파일 불러오기/검색1/검색2/검색3
  예) CRUD : 완성
  다중 데이터 : 완성
  ...
- 모든 소스파일 텍스트 복&붙 제출(product.h, product.c, manager.h.manager.c, main.c,Makefile)
- 모든 메뉴 실행결과 텍스트 복&붙 제출

* 과제제출시 반드시 하나의 파일로 작성하여 학번_이름_mini4.pdf 로 제출

CRUD를 구현한 본인의 mini project의 소스를 이용하여 다음 조건에 따라 코딩하세요.

1. mini project 폴더를 git 저장소로 초기화
2. mini project에 파일저장, 파일 불러오기, 검색 기능을 추가하면서 변경 이력(커밋 이력 10회 이상)을 기록하면서 git 명령어를 다양하게 사용한다. (본인이 설계한 검색기능 3가지 포함)
3. github에 로그인 > 새 저장소 생성(remote repository)
4. mini project을 원격 저장소에 연결하여(git remote add 명령 참조) push한다.

중간고사

- **Quiz 공지**
  - 일시 : 4/20 (8주차 화요일 수업시간), HD LMS 퀴즈에서 시험
    - 1분반 : 10시 ~11:15 (2교시)
    - 2분반 : 11:30 ~ 12:45 (3교시)

- **실기시험(동영상)**
  - 기한 : ~ 4/25(일) 23:50 까지
  - 실습과제를 동영상으로 제작 제출
  - 자세한 내용은 히즈넷 확인