

# Commit 취소



About..

컴퓨터소프트웨어공학과  
김 원 일



## Commit 취소



- Commit

- 로컬 데이터베이스에 Staged 파일을 기록하고, 관리 상태로 만드는 것
- 이력(history) 관리를 위해 해시 값과 스냅샷 등이 데이터베이스에 기록
- Commit은 파일 관련 정보가 같이 기록되므로 주의하여 수행해야 함

- 기록 내용 삭제

- commit을 통해 관리 상태에 들어간 파일들은 이력 관리가 시작됨
- 잘못된 commit이 발생한 경우에는 Commit을 취소할 수 있음
- 취소될 때, Commit에서 입력 또는 저장된 정보가 삭제됨
- 삭제 방법에 따라 복구가 되지 않는 부분도 존재하기 때문에 주의 필요
- 가장 좋은 방법은 Commit을 너무 자주 수행하지 않는 것
- 어느 정도는 Commit을 자주 하는 것이 좋으나 너무 남발하면 취소가 어려움



# 실습 환경 구성 - 1



## • 실습 저장소 생성

- 처음 만들어지는 파일에 대한 commit 취소 확인을 먼저 진행
- 실습을 위한 새로운 디렉토리를 생성하고, git init으로 초기화
- 파일을 새로 생성하고, commit 까지 진행

```
MINGW64:/f/202507001/reset

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001
$ mkdir reset

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001
$ cd reset/

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset
$ git init
Initialized empty Git repository in F:/202507001/reset/.git/

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ ls -al > list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git add list.txt
warning: in the working copy of 'list.txt', LF will be replaced by CRLF the next
time Git touches it

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git commit
```



## 실습 환경 구성 - 2



### • 실습 환경 구성

- git log로 commit이 존재하는 상황까지 확인할 것
- 저장소 생성 후 첫 번째 Commit 이므로 1개의 log 정보만 출력
- Commit만 수행한 상태로 환경을 구성하여 Commit 취소 실습 진행

```
MINGW64:/f/202507001/reset
[master (root-commit) a05ab9f] 1. list.txt [ + ] add list.txt file
1 file changed, 5 insertions(+)
create mode 100644 list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git log
commit a05ab9f67a37327a1b8cd4891344990638b1dbd1 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 11:52:50 2025 +0900

    1. list.txt
    [ + ] add list.txt file

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ |
```



# Commit 되돌리기 - 1



- commit 취소하기

- 취소는 "git reset" 명령어를 통해서 commit 취소 수행이 가능
- reset 뒤에 commit을 어느 정도까지 취소할 것인지를 option으로 입력
- option은 별도 입력 없이도 가능하지만 취소 상태를 결정하므로 대부분 입력
- 뒤에 어디를 취소할 것인지를 알리는 형태로 취소를 진행
- 항상 현재 Commit을 가리키는 HEAD로 Commit 취소 위치를 설정
- Ex) git reset --option HEAD
- HEAD 뒤에 ^ 또는 숫자를 이용하여 취소 위치를 전달
- ^를 이용할 경우 ^ 하나당 1개의 이전 Commit을 의미
- Ex) git reset --option HEAD^ or HEAD^^ ...
- 숫자를 이용하는 경우 "~숫자" 형식으로 이전 Commit을 가리킴
- Ex) git reset --option HEAD~1 or HEAD~2 ...



## Commit 되돌리기 - 2



### • 취소 실습

- 현재 상태에서 reset 명령어를 이용하여 수행된 결과를 확인
- 저장소의 별도의 변화는 없는 것으로 확인되고 Commit도 그대로 유지

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git reset HEAD

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
nothing to commit, working tree clean

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git log
commit a05ab9f67a37327a1b8cd4891344990638b1dbd1 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 11:52:50 2025 +0900

    1. list.txt
    [ + ] add list.txt file

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ ls
list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ |
```



## Commit 되돌리기 – 3



### • 저장소 변경

- Commit한 list.txt 파일을 수정하고, Staged 상태까지 진행
- 새로운 new.txt 파일을 생성하고 Untracked 상태로
- 새로운 test.txt 파일을 생성하고, Staged 상태까지 진행

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ vi list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git add list.txt
warning: in the working copy of 'list.txt', LF will be replaced by CRLF the next time
Git touches it

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ touch new.txt ; touch test.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git add test.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   list.txt
        new file:   test.txt

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



## Commit 되돌리기 - 4



- 저장소 reset

- "git reset HEAD" 실행 후 Staged 상태의 파일이 모두 해제
- reset을 옵션 없이 사용하면 현재 저장소 상태를 변경
- Staged 상태를 모두 해제하여 Commit 직후 상태로 되돌림
- 즉, reset은 Commit 취소 기능과 상태 되돌림을 동시에 수행할 수 있음

```
MINGW64:/f/202507001/reset
unange1@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git reset HEAD
Unstaged changes after reset:
M      list.txt

unange1@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list.txt

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

no changes added to commit (use "git add" and/or "git commit -a")

unange1@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$
```





## Commit 되돌리기 – 5



### • reset 결과

- “git reset HEAD^”은 이전 Commit 취소와 되돌리기 수행 명령
- 현재 Commit이 하나만 존재하는 상태이기 때문에 이전 Commit이 없는 상태
  - HEAD가 1개만 존재하는 Commit을 가리키고 있는 상태
- 따라서 Commit이 1개일 때는 HEAD에 이전 표시는 잘못된 지정으로 표시

```
MINGW64:/f/202507001/reset
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list.txt

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

no changes added to commit (use "git add" and/or "git commit -a")

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git reset HEAD^
fatal: ambiguous argument 'HEAD^': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ |
```



## Commit 되돌리기 - 6



### • 실습 환경 설정 - 1

- 추적되지 않는 파일은 삭제하고, 수정된 파일은 restore 처리
- Commit에서 변경된 내역을 모두 취소하고 원래 상태로 되돌림

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ rm -rf new.txt test.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git restore list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
nothing to commit, working tree clean

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git log
commit a05ab9f67a37327a1b8cd4891344990638b1dbd1 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 11:52:50 2025 +0900

    1. list.txt
    [ + ] add list.txt file

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$
```



# Commit 되돌리기 - 7



## • 실습 환경 설정 - 2

- 3개의 Commit을 추가하여 reset에 따라 변경되는 내역 확인을 위한 설정
- HEAD를 이용한 reset의 결과 확인을 위한 Commit 메시지 입력

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git log
commit ce234306330d11a1938e87364d574e6f2dc9a1cd (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 21:03:24 2025 +0900

    4. list.txt
    [ * ] 4th edit

commit 7b10169fff67b6edd6b484bfe6a465672f6c75c0
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 21:02:36 2025 +0900

    3. list.txt
    [ * ] modify 3rd

commit 7bd9eead44f4749e4471d07e30792aebbddd1f602
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 21:01:49 2025 +0900

    2. list.txt
    [ * ] modify

commit a05ab9f67a37327a1b8cd4891344990638b1dbd1
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 11:52:50 2025 +0900

    1. list.txt
    [ + ] add list.txt file
```



## Commit 되돌리기 – 8



- Commit 취소

- “git reset HEAD^” 명령어를 실행하면 마지막 실행 Commit이 취소됨
- 4번 Commit이 해제되어 git log에 3번 Commit이 마지막으로 나타남

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git reset HEAD^
Unstaged changes after reset:
M      list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git log
commit 7b10169fff67b6edd6b484bfe6a465672f6c75c0 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 21:02:36 2025 +0900

    3. list.txt
    [ * ] modify 3rd

commit 7bd9eead44f4749e4471d07e30792aebbddd1f602
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 21:01:49 2025 +0900

    2. list.txt
    [ * ] modify

commit a05ab9f67a37327a1b8cd4891344990638b1dbd1
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 11:52:50 2025 +0900

    1. list.txt
    [ + ] add list.txt file

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ |
```



## Commit 되돌리기 - 9



### • 취소 상태 확인

- 3번째 Commit 이후 4번째 수정이 이루어진 상태까지 취소된 것을 확인
- 수정 사항을 확인해보면 4번째 Commit 전에 수정한 상태임을 확인할 수 있음

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list.txt

no changes added to commit (use "git add" and/or "git commit -a")

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git add list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git diff --cached list.txt
diff --git a/list.txt b/list.txt
index e9877ea..7f5392b 100644
--- a/list.txt
+++ b/list.txt
@@ -2,6 +2,7 @@ total 0

    1. modify
    2. modify second
+3. third modify

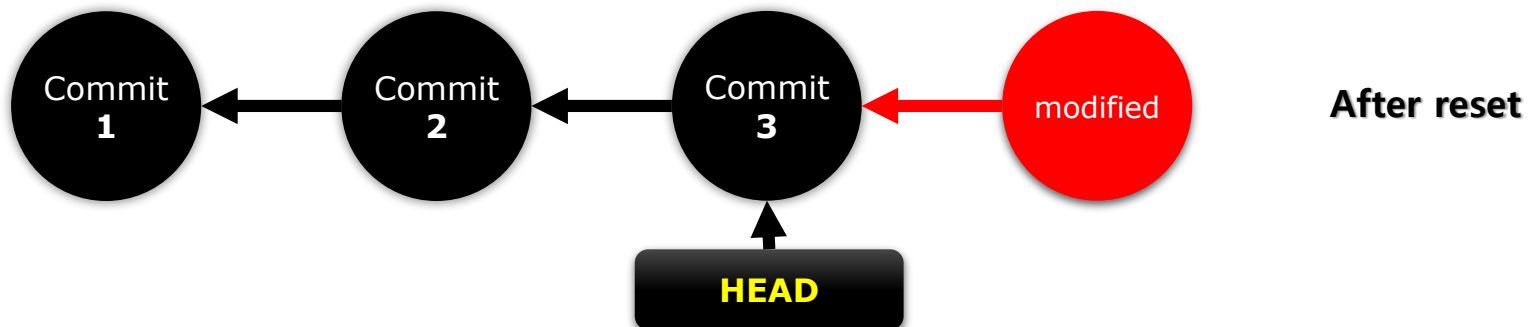
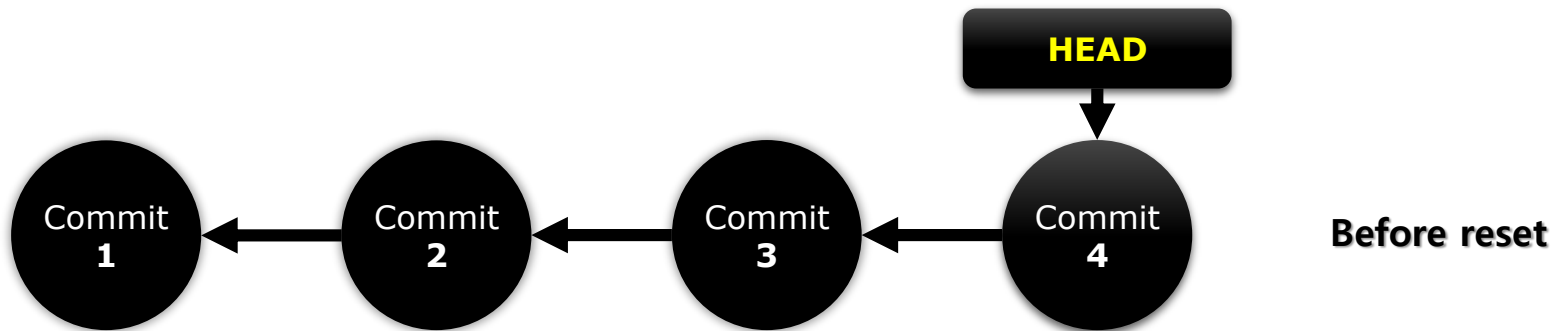
drwxr-xr-x 1 unangel 197121 0 May 18 11:52 ./
drwxr-xr-x 1 unangel 197121 0 May 18 11:52 ../

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ |
```



## • 취소 상태

- "git reset HEAD^" 명령 실행으로 HEAD가 가리키는 Commit은 일단 해제
- Commit 4에서 Commit 되었던 코드는 파일에 수정 상태로 적용
- 즉시 Commit하면 Commit 4와 동일하게 Commit 재실행 가능
- Commit 메시지는 삭제되기 때문에 유일하게 완전히 삭제되는 정보





# Commit 되돌리기 - 11



- 다시 Commit

- 4번째 Commit을 다시 적용하고, 다시 Commit 하였음을 메시지로 입력

```
MINGW64:/f/202507001/reset

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list.txt

no changes added to commit (use "git add" and/or "git commit -a")

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git add list.txt ; git commit
```

```
MINGW64:/f/202507001/reset

commit 71eb76f145f812aaf367d43b44bb08e9b197f313 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 23:47:44 2025 +0900

    4. list.txt
    [ * ] re-commit

commit 7b10169fff67b6edd6b484bfe6a465672f6c75c0
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 21:02:36 2025 +0900

    3. list.txt
:|
```



# Commit 되돌리기 - 12



## • 2개 Commit 취소

- HEAD 뒤에 ^를 2개 붙여 HEAD로 부터 2개 Commit을 취소
- git log에서도 2번이 나타난 것을 확인할 수 있음

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git reset HEAD^^
Unstaged changes after reset:
M    list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list.txt

no changes added to commit (use "git add" and/or "git commit -a")

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ |
```

```
MINGW64:/f/202507001/reset
commit 7bd9eead44f4749e4471d07e30792aebbddd1f602 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 21:01:49 2025 +0900

    2. list.txt
    [ * ] modify

commit a05ab9f67a37327a1b8cd4891344990638b1dbd1
Author: wikim <unangel@yuhan.ac.kr>
Date:   Sun May 18 11:52:50 2025 +0900

:|
```





# Commit 되돌리기 - 13



## • 2개가 취소된 상태

- 1개가 취소된 것과 같이 Commit이 해제된 모든 내용 파일에 그대로 남음
  - 다수 Commit 해제 시, 해제된 모든 Commit 내용이 수정된 것으로 파일 내용이 기록됨

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git add list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git diff --cached list.txt
diff --git a/list.txt b/list.txt
index a5f67cd..7f5392b 100644
--- a/list.txt
+++ b/list.txt
@@ -1,7 +1,8 @@
| total 0
|
| 1. modify
-
+2. modify second
+3. third modify

drwxr-xr-x 1 unangel 197121 0 May 18 11:52 ./
drwxr-xr-x 1 unangel 197121 0 May 18 11:52 ../
```

```
MINGW64:/f/202507001/reset
1 total 0
2
3 1. modify
4 2. modify second
5 3. third modify
6
7 drwxr-xr-x 1 unangel 197121 0 May 18 11:52 ./
8 drwxr-xr-x 1 unangel 197121 0 May 18 11:52 ../
9 drwxr-xr-x 1 unangel 197121 0 May 18 11:52 .git/
10 -rw-r--r-- 1 unangel 197121 0 May 18 11:52 list.txt
list.txt [dos] (21:03 18/05/2025) 6,0-1 All
```

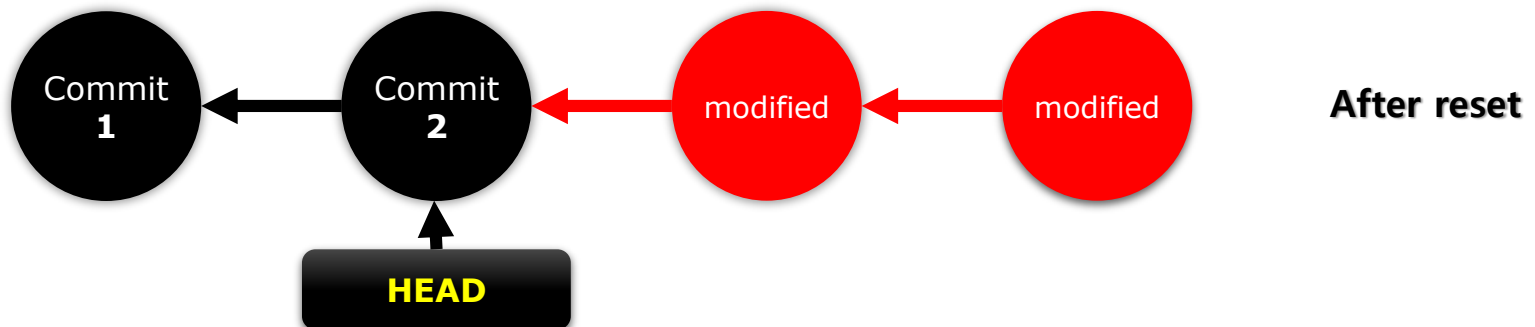
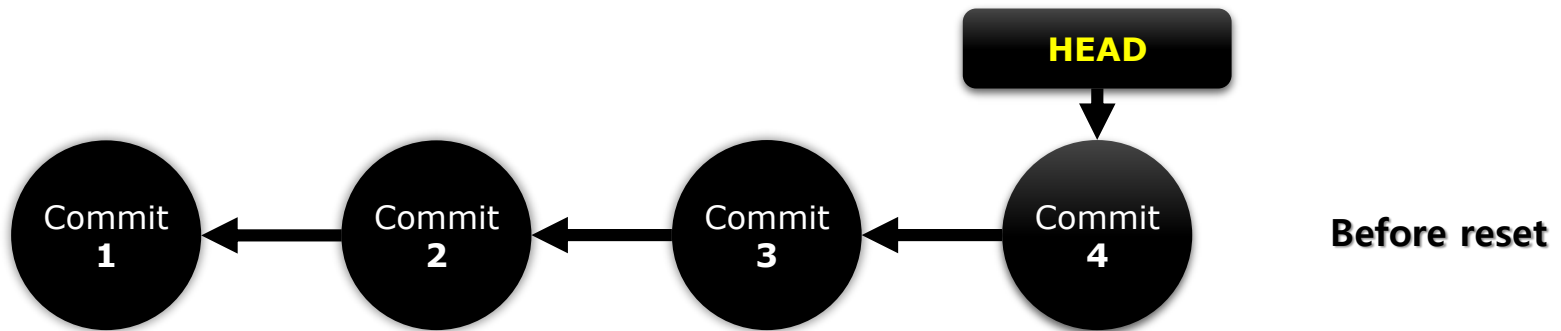


## Commit 되돌리기 - 14



### • 다수 Commit 취소 상태

- 취소된 Commit의 변경 상태가 취소 요청 위치에 모두 적용된 상태
- N개의 취소를 요청하면 현재 위치 - N 위치에 취소된 내용이 수정 상태로 적용
- 마찬가지로 Commit 메시지는 삭제되기 때문에 주의
- squash 병합과 달리 Commit 메시지가 유지되지 않음



# git reset



About..

컴퓨터소프트웨어공학과  
김 원 일

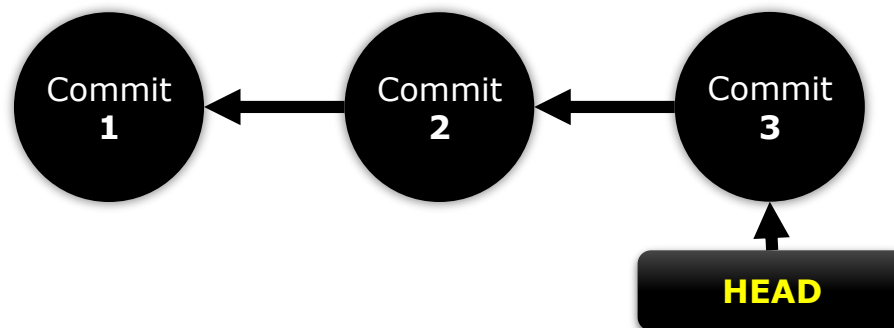


# git reset option - 1



## • “git reset” 옵션

- git에는 Commit을 취소하기 위한 옵션을 3가지를 제공
- 옵션은 해제 시, 파일의 상태를 결정하기 때문에 필요한 옵션을 선택하여 사용
- “-- soft” : 현재 파일 상태를 그대로 유지하면서 취소
- “-- mixed” : add 전 상태로 모든 파일을 되돌리면서 취소
  - git reset을 옵션 없이 사용하면 “--mixed” 옵션을 사용한 것과 동일한 상태
- “-- hard” : 모든 정보를 취소하므로 주의해서 사용
- 다수 Commit 취소에서 Commit을 수행하여 아래와 같은 상태에서 실습 진행
- 옵션 확인을 위한 실습이므로 다수 Commit 취소를 하지는 않음
- 파일 상태 확인이므로 Staged, Modified 상태의 파일 하나씩 추가하여 확인





## git reset option - 2



### • 실습 환경

- Commit 수행 후, Staged, Modified 2개의 파일을 추가한 상태로 실습
- reset 수행 시에 발생하는 저장소의 변화를 확인하기 위한 실습

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git commit -m "다 수 의 파 일 취 소 를 하 나 의 Commit으 로 합 침 "
[master 7095ba0] 다 수 의 파 일 취 소 를 하 나 의 Commit으 로 합 침
1 file changed, 2 insertions(+), 1 deletion(-)

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ touch staged.txt ; touch modified.txt ; git add staged.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   staged.txt

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

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$
```



## git reset option - 3



### • "--soft" 옵션 - 1

- 현재 파일들의 상태가 변화되지 않는 상태에서 Commit만 해제
  - 저장소에 존재하는 파일들은 상태를 유지하면서 Commit 해제만 수행
- Staged, Modified 상태의 파일들은 별도의 변화가 발생하지 않음
  - list.txt는 Staged 상태로 변경되어 Commit을 즉시 수행할 수 있는 상태로 취소
- 단순히 HEAD가 가리키는 현재 Commit만 해제하는 것을 확인

```
MINGW64:/f/202507001/reset

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git reset --soft HEAD^

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   list.txt
    new file:   staged.txt

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

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ |
```



## git reset option - 4



- “--soft” 옵션 - 2

- 이전 상태와 동일하지만, Commit된 파일인 list.txt를 수정한 상태
- 해당 상태에서 soft 옵션으로 Commit을 해제하고 상태를 확인
- 이전과 동일하게 하나의 Commit을 취소하여 결과를 확인

```
MINGW64:/f/202507001/reset

unange1@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   staged.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   list.txt

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

unange1@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git reset --soft HEAD^
```



## git reset option - 5



- “--soft” 옵션 - 3

- 취소 수행 시, 이전 상태와 동일하게 Commit은 해제
- list.txt가 Commit이 해제된 상태와 취소 전에 수정한 상태 2가지가 존재
- 파일 3개(원본, Staged, Modified)가 모두 생성되므로 주의해야 함
- **취소 수행할 때는 이러한 사태가 발생하지 않도록 반드시 확인 후 수행**

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   list.txt
    new file:   staged.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   list.txt

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

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$
```





## git reset option - 6



- “--mixed” 옵션

- soft 되돌리기에서 list.txt만 Commit하여 동일한 실습 환경 구성
- “--mixed”로 Commit을 되돌리고 파일들의 상태를 확인
- “git reset”의 옵션 없는 실행과 동일하게 Commit 해제 후 수정까지 진행
- Staged 상태의 파일들도 모두 Modified 상태로 변경해버리는 것을 주의

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git reset --mixed HEAD^
Unstaged changes after reset:
M      list.txt

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   list.txt

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

no changes added to commit (use "git add" and/or "git commit -a")

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ |
```



# git reset option - 7



- “--hard” 옵션 - 1

- “--hard” 옵션 적용으로 reset 수행 시 Commit과 메시지 제거
- 취소한 Commit에 대한 정보 자체가 존재하지 않음
  - Staged 상태였던 파일에 대한 정보가 나타나지 않는 것도 확인 가능

```
MINGW64:/f/202507001/reset
unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   staged.txt

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

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git reset --hard HEAD^
HEAD is now at 7bd9eea 2. list.txt [ * ] modify

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    modified.txt

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

unangel@DESKTOP-I80QG85 MINGW64 /f/202507001/reset (master)
$
```



## git reset option - 8



- “--hard” 옵션 - 2

- “git log”에서 3번째 Commit에 대한 정보가 삭제된 것을 확인
- Commit과 메시지가 완전히 삭제되기 때문에 주의해야 함
- 파일 목록에서 staged.txt 파일 또한 존재하지 않는 것을 확인
- 지정한 이전 Commit 상태로 파일들을 완전히 되돌림
  - 즉, 취소 Commit에서 Staged 파일이 존재하지 않는 상태였으므로 완전히 제거

```
MINGW64:/f/202507001/reset

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ git log
commit 7bd9eead44f4749e4471d07e30792aebbdd1f602 (HEAD -> master)
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 21:01:49 2025 +0900

    2. list.txt
    [ * ] modify

commit a05ab9f67a37327a1b8cd4891344990638b1dbd1
Author: wikim <unangel@yuhan.ac.kr>
Date: Sun May 18 11:52:50 2025 +0900

    1. list.txt
    [ + ] add list.txt file

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$ ls
list.txt  modified.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/reset (master)
$
```



# git reset 정리



## • git reset 명령어

- "--hard, --soft, --mixed" 3개의 옵션으로 Commit을 되돌릴 수 있음
- Commit 취소로 되돌리는데, 옵션에 따라 다르게 적용 가능하므로 주의
- 특히 "--hard" 옵션을 이용하는 경우
  - 요구한 만큼의 Commit을 취소하고, 모든 파일을 지정한 Commit 상태로 복원
  - 수정하거나 작성한 모든 파일이 제거되거나 지정 Commit으로 되돌려짐
  - 이전 상태로 완전 복원되므로, 작업한 파일 모두가 완전 삭제
- Commit을 취소하기 때문에 원격 저장소와 Commit이 다를 수 있음
- 원격에 5개, 로컬에 취소를 통해 4개 Commit으로 변경된 경우
  - 로컬의 4번째 Commit은 로컬에서 별도로 이루어지는 Commit
  - 즉, 원격 저장소와 로컬 저장소의 4번째 Commit부터 전혀 다른 정보를 가지게 됨
  - 이렇게 정보가 달라지는 경우에는 원격 저장소로 업로드 시 충돌이 발생
  - 강제로 업로드 할 수 있으나 다른 사용자와도 충돌이 발생
- git reset은 로컬에서 개인적으로 또는 테스트 용으로만 사용
- 원격 저장소에 존재하는 Commit을 변경하는 경우에는 절대 사용해서는 안됨
- 원격 저장소까지 영향을 주는 Commit 취소는 revert를 사용해야 함

*All things are difficult, before they are easy.*

