

Open source Software

오픈소스 소프트웨어

23. [실습] 다양한 브랜치 병합



학습 개요

1. 브랜치 병합 merge
2. fast-forward, 3-way merge
3. squash merge



학습 목표

1. 브랜치의 병합(merge)을 이해하고 수행할 수 있다.
2. 빨리 감기(fast-forward) 병합을 수행할 수 있다.
3. 3-way 상태를 이해하고 3-way 병합을 수행할 수 있다.
4. 기본이 fast-forward 병합이 수행되는 일련 상태에서 3-way 병합을 수행할 수 있다.

LESSON 01

fast-forward 병합



1 fast-forward 병합

병합을 실습할 기본 상태 1

저장소 gline 생성

일렬 상태의 두 브랜치 생성



```
$ git init gline
Initialized empty Git repository in C:/[smart
git]/gline/.git/
```

```
$ cd gline
```

```
$ echo 111 > f
```

```
$ git add f
```

```
$ git commit -m 1
[main (root-commit) 3d00e27] 1
1 file changed, 1 insertion(+)
create mode 100644 f
```

```
$ git switch -c topic
Switched to a new branch 'topic'
```

```
$ echo aaa > g
```

```
$ git add g
```

```
$ git commit -m A
```

```
$ echo bbb >> g
```

```
$ git commit -am B
[topic 8dd39e7] B
1 file changed, 1 insertion(+)
```

```
$ git log --all --graph --oneline
* 8dd39e7 (HEAD -> topic) B
* eb0bfd8 A
* 3d00e27 (main) 1
```

```
$ git config --global alias.lgag1 'log --all --
graph --oneline'
```

```
$ git lgag1
* 8dd39e7 (HEAD -> topic) B
* eb0bfd8 A
* 3d00e27 (main) 1
```

1 fast-forward 병합

저장소의 복사 활용

처음에 만든 저장소 gline을 복사

- 새 이름 gffm 등으로 만들어 사용 가능
 - \$ cp -r gline gffm
 - 일렬 상태에서 바로 다음 기능을 수행 가능



```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gline (topic)
$ cd ..
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ cp -r gline gffm
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ cd gffm
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (topic)
$ git lgag1
* 8dd39e7 (HEAD -> topic) B
* eb0bfd8 A
* 3d00e27 (main) 1
```

1 fast-forward 병합

fast-forward 병합

이전에 만든 폴더 복사해서 사용

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gline (topic)
$ cd ..
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ cp -r gline gffm
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ cd gffm
```

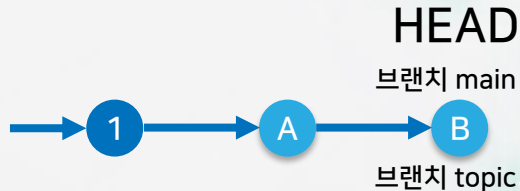
```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (topic)
$ git lgag1
* 8dd39e7 (HEAD -> topic) B
* eb0bfd8 A
* 3d00e27 (main) 1
```



```
$ git switch main
Switched to branch 'main'
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git merge topic
Updating 3d00e27..8dd39e7
Fast-forward
 g | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 g
```

```
$ git lgag1
* 8dd39e7 (HEAD -> main, topic) B
* eb0bfd8 A
* 3d00e27 1
```



1 fast-forward 병합

fast-forward 병합 이전으로 돌아가기

나중에 학습할 reset으로 다음 두 명령 모두 가능

- \$ git reset --hard HEAD~~
- \$ git reset --hard 3d00e27

브랜치 main에서 작업

```
$ git lgag1
* 8dd39e7 (HEAD -> main, topic) B
* eb0bfd8 A
* 3d00e27 1
```

```
$ git reset --hard 3d00e27
HEAD is now at 3d00e27 1
```

```
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```

1 fast-forward 병합

옵션 --ff--only

일렬 상태에서는 --ff-only로 빨리 감기 가능

이전 병합을 취소하고 다시 병합

\$ git merge topic --ff-only

```
$ git lgag1
* 8dd39e7 (HEAD -> main, topic) B
* eb0bfd8 A
* 3d00e27 1
```

```
$ git reset --hard 3d00e27
HEAD is now at 3d00e27 1
```

```
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```

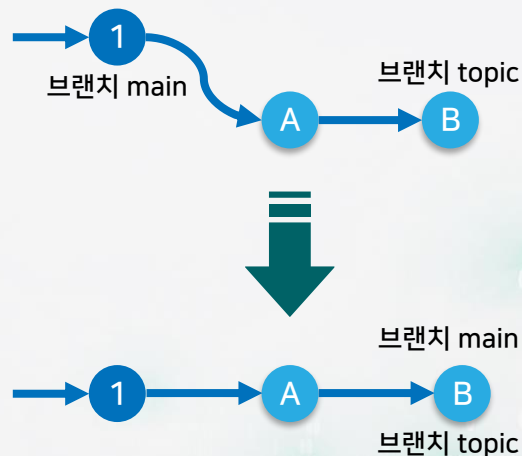
```
$ git reset --hard 3d00e27
HEAD is now at 3d00e27 1
```

```
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```

이전 커밋 상태로 이동하는 명령

```
$ git merge topic --ff-only
Updating 3d00e27..8dd39e7
Fast-forward
 g | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 g
```

```
$ git lgag1
* 8dd39e7 (HEAD -> main, topic) B
* eb0bfd8 A
* 3d00e27 1
```



LESSON 02

3-way와 squash 병합



2 3-way와 squash 병합

3-way 병합

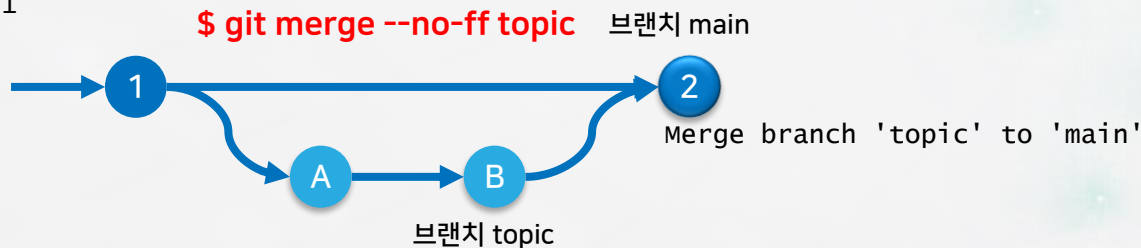
일렬 상태에서 3-way 병합

```
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```

```
$ git merge topic --no-ff
Merge made by the 'ort' strategy.
g | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 g
```

```
$ git lgag1
* 1963612 (HEAD -> main) Merge branch 'topic' to 'main'
\
* 8dd39e7 (topic) B
* eb0bfd8 A
/
* 3d00e27 1
```

```
◆ MERGE_MSG ◆
C: > [smart git] > gffm > .git > ◆ MERGE_MSG
1 Merge branch 'topic' to 'main'
2 # Please enter a commit message to explain why this merge is necessary,
3 # especially if it merges an updated upstream into a topic branch.
4 #
5 # Lines starting with '#' will be ignored, and an empty message aborts
6 # the commit.
7 |
```



2 3-way와 squash 병합

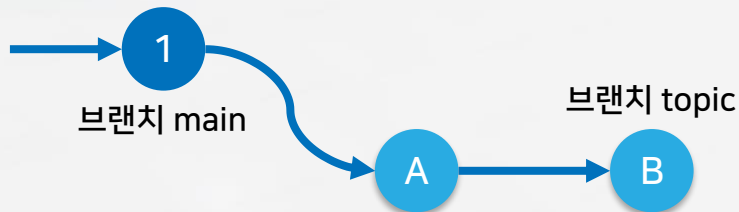
⚙ 다시 일렬 상태로 복구

✓ 나중에 학습할 reset 명령으로

- \$ git reset --hard

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git reset --hard 3d00e27
HEAD is now at 3d00e27 1
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```



2 3-way와 squash 병합

직접 메시지 입력하는 3-way 병합

옵션 -m 'msg'

```
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```

```
$ git merge topic --no-ff -m 'merge --no--ff'
```

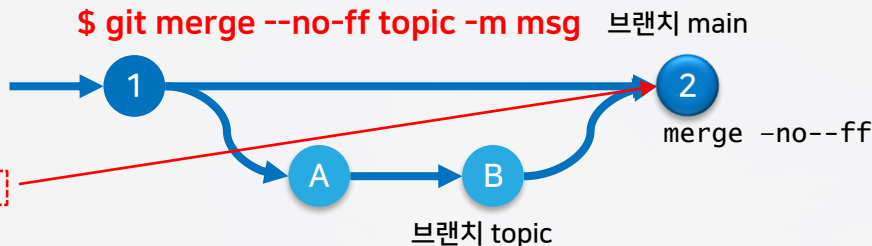
```
Merge made by the 'ort' strategy.
```

```
g | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 g
```

```
$ git lgag1
* 6873bda (HEAD -> main) merge --no--ff
| \
| * 8dd39e7 (topic) B
| * eb0bfd8 A
| /
* 3d00e27 1
```

```
$ git reset --hard 3d00e27
HEAD is now at 3d00e27 1
```

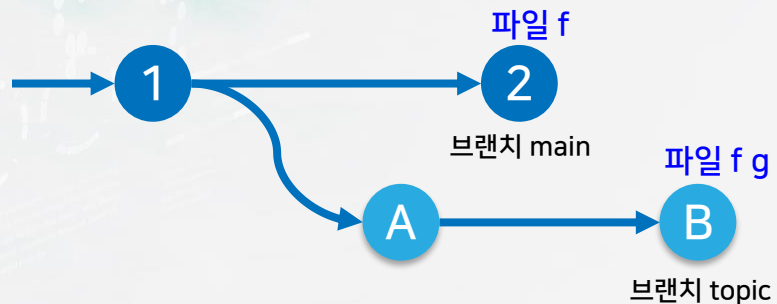
```
$ git lgag1
* 8dd39e7 (topic) B
* eb0bfd8 A
* 3d00e27 (HEAD -> main) 1
```



2 3-way와 squash 병합

3-way 상태 만들기

브랜치 main에서 커밋 추가



```
$ echo 222 >> f
```

```
$ git commit -am 2
[main 20c8673] 2
1 file changed, 1 insertion(+)
```

```
$ git lgag1
* 20c8673 (HEAD -> main) 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

```
$ ls
f
```

2 3-way와 squash 병합

3-way 상태에서 3-way 병합

기본이 3-way 병합

```
$ git lgag1
* 20c8673 (HEAD -> main) 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

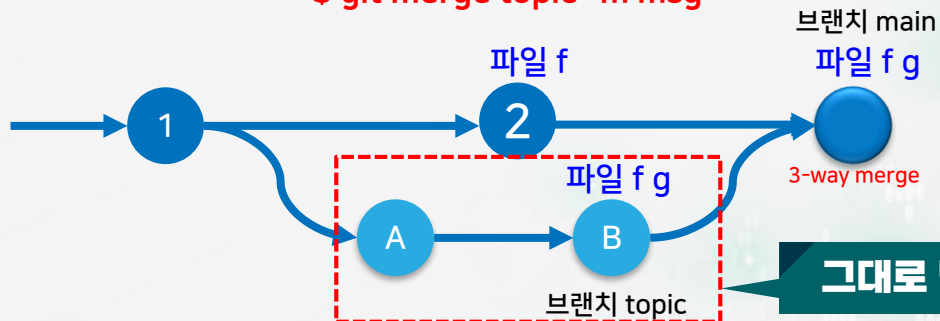
```
$ git merge topic -m '3-awy merge'
Merge made by the 'ort' strategy.
g | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 g
```

```
$ git lgag1
* 88d8834 (HEAD -> main) 3-awy merge
| \
| * 8dd39e7 (topic) B
| * eb0bfd8 A
* | 20c8673 2
|/
* 3d00e27 1
```

```
$ git reset --hard 20c8673
HEAD is now at 20c8673 2
```

```
$ git lgag1
* 20c8673 (HEAD -> main) 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

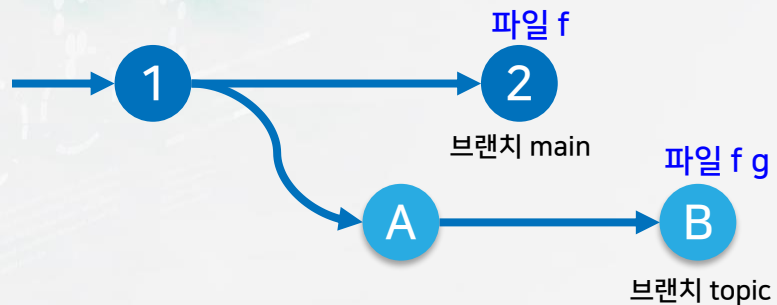
Merge branch 'topic' to 'main'
\$ git merge topic -m msg



2 3-way와 squash 병합

⚙ 옵션 --squash 병합 준비

☑ 합병 전 상태



```
$ git lgag1
* 20c8673 (HEAD -> main) 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

2 3-way와 squash 병합

Squash 병합

새로운 커밋에 상대 브랜치의 내용을 모두 합해 새로운 커밋 하나로 병합

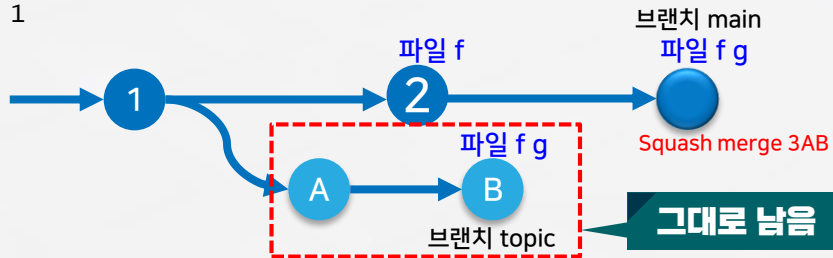
- 직접 메시지와 함께 커밋을 해야 함

```
$ git merge --squash topic
Automatic merge went well; stopped before committing as requested
Squash commit -- not updating HEAD
```

```
$ git commit -m 'Squash merge 3AB'
[main 6a74e74] Squash merge 3AB
1 file changed, 2 insertions(+)
create mode 100644 g
```

```
$ git lgag1
* 6a74e74 (HEAD -> main) Squash merge 3AB
* 20c8673 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

```
$ ls
f g
```



```
$ git switch topic
Switched to branch 'topic'
```

```
$ git lgag1
* 6a74e74 (main) Squash merge 3AB
* 20c8673 2
| * 8dd39e7 (HEAD -> topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

```
$ git log --oneline
8dd39e7 (HEAD -> topic) B
eb0bfd8 A
3d00e27 1
```

```
$ ls
f g
```


2 3-way와 squash 병합

Squash 병합 이전으로 돌아가기

나중에 학습할 reset

- \$ git reset --hard HEAD~

브랜치 main에서 작업

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (topic)
$ git lgag1
* 6a74e74 (main) Squash merge 3AB
* 20c8673 2
| * 8dd39e7 (HEAD -> topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (topic)
$ git switch -
Switched to branch 'main'
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git lgag1
* 6a74e74 (HEAD -> main) Squash merge 3AB
* 20c8673 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git reset --hard HEAD~
HEAD is now at 20c8673 2
```

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git lgag1
* 20c8673 (HEAD -> main) 2
| * 8dd39e7 (topic) B
| * eb0bfd8 A
|/
* 3d00e27 1
```

2 3-way와 squash 병합

병합 전에 파일 수정이 된 경우

상태가 3영역이 동일하지 않아도 파일 충돌만 없으면 병합 가능

```
$ cat f
111
222

$ echo 333 >> f

$ git status
On branch main
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:   f

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

$ git merge topic -m '3-way merge'
Merge made by the 'ort' strategy.
  g | 2 ++
  1 file changed, 2 insertions(+)
  create mode 100644 g
```

```
$ git status
On branch main
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:   f

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

$ git lgag1
*   d8994b2 (HEAD -> main) 3-way merge
| \
|  * 8dd39e7 (topic) B
|  * eb0bfd8 A
* | 20c8673 2
|/
* 3d00e27 1
```

Summary

» 기준 브랜치에서 hotfix 브랜치 병합

- ◆ \$ git merge hotfix
 - fast-forward, 3-way merge

» 무조건 3-way 병합 수행

- ◆ \$ git merge --no-ff hotfix

» fast-forward인 경우에만 병합 진행

- ◆ \$ git merge --ff-only hotfix

» 현재 브랜치에서 커밋 하나만 생성해서 병합

- ◆ \$ git merge --squash hotfix