Open Source Software

Staged 파일 상태

About..

컴퓨터소프트웨어공학과

김 원 일

Staged 파일 상태 - 1

- **파일 수정**
  - commit 된 파일을 상태에서 파일 수정
  - 수정한 상태에서 파일을 준비 상태까지 진행

MINGW64:/f/suho

```
unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ ls
list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ ls > list.txt

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


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

Staged 파일 상태 - 2

- 읽기 전용 상태
  - 파일이 준비(Staged) 상태가 되면 읽기 전용 상태가 됨
  - 읽기 전용 상태가 적용되지 않았음을 확인
  - 탐색기의 파일 속성이 읽기 전용이 아님
  - 쉘에서도 "w"가 있어 기록이 가능함

```
MINGW64:/f/suho

unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ ls -al
total 14
drwxr-xr-x 1 unangel 197121  0 Mar 24 21:44 ./
drwxr-xr-x 1 unangel 197121  0 Mar 22 19:32 ../
drwxr-xr-x 1 unangel 197121  0 Mar 24 21:54 .git/
-rw-r--r-- 1 unangel 197121 29 Mar 21 23:59 .gitignore
-rw-r--r-- 1 unangel 197121  9 Mar 24 21:54 list.txt

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

Staged 파일 상태 - 3

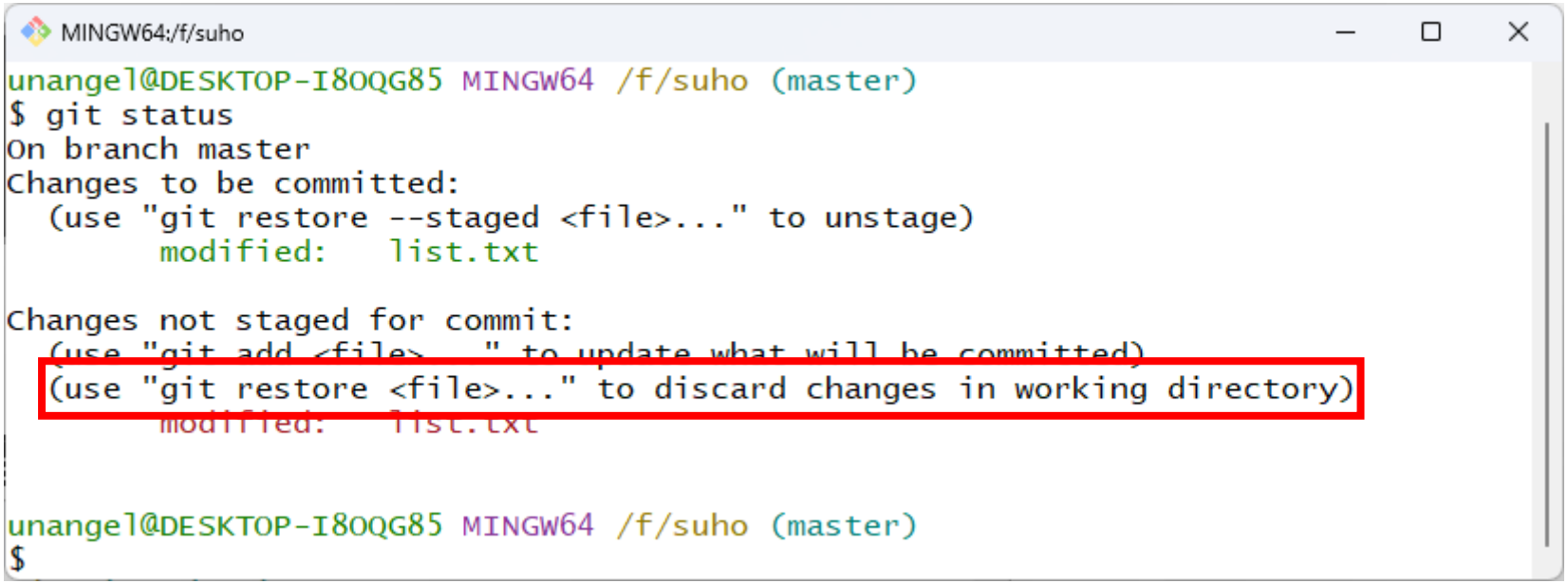
- **파일 수정**
  - Staged 상태의 파일을 수정할 수 있음을 확인

MINGW64:/f/suho

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


unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ ls -a > list.txt

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


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

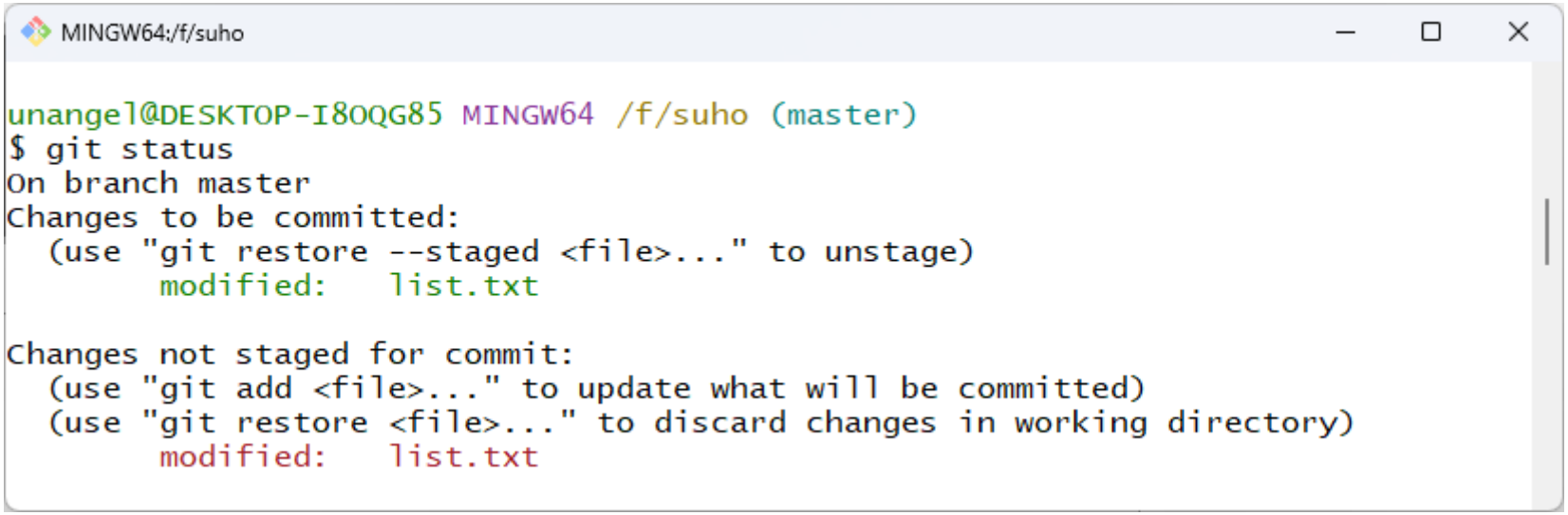
Staged 파일 상태 - 4

- **상태 정보 변화**
  - 아래와 붉은색 박스와 같이 일반적인 출력에 추가 메시지가 출력
  - 일반적인 파일 수정 후 출력에서는 "git add"에 관련된 메시지만 출력
  - 추가 메시지는 staged 파일과 같은 "git restore" 명령 정보 출력
  - 파일이 여러 상태를 가지므로 처리 방식이 변경됨



Staged 파일 상태 - 5

- **다중 파일 상태**
  - **하나의 파일이 3개의 상태를 가지고 있는 것을 확인할 수 있음**
  - **즉, 파일의 읽기 전용 속성은 git에서 관리하는 속성**
  - **일반적인 파일 관리를 위한 속성이 읽기 전용이라는 것이 아님**
  - **현재 파일들의 내용물**
    - commit **파일** : ls -al **명령어의 실행 결과**
    - staged **파일** : ls **명령어의 실행 결과**
    - modified **파일** : ls -a **실행 결과**



Staged 파일 상태 - 6

- **파일들의 실제 내용**
  - 실습을 그대로 진행하였다면 아래와 같은 상태
    - 마지막으로 수정된 상태는 ls -a 명령어 실행 상태

MINGW64:/f/suho

```
unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ ls -al
total 14
drwxr-xr-x 1 unangel 197121  0 Mar 24 21:44 ./
drwxr-xr-x 1 unangel 197121  0 Mar 22 19:32 ../
drwxr-xr-x 1 unangel 197121  0 Mar 24 22:34 .git/
-rw-r--r-- 1 unangel 197121 29 Mar 21 23:59 .gitignore
-rw-r--r-- 1 unangel 197121 33 Mar 24 22:08 list.txt

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

unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ ls -a
./  ../  .git/  .gitignore  list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ cat list.txt
./
../
.git/
.gitignore
list.txt

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

Staged 파일 상태 - 7

- **수정 파일을 취소**
  - "git restore"로 현재 수정된 파일의 변경을 취소
  - 취소된 상태에서 상태 정보를 확인하면 아래와 같음
  - 파일의 내용을 확인하면 "ls -a" 내용 대신 "ls" 결과가 출력
  - 수정했던 "ls -a"의 내용은 완전히 사라지므로 주의

```
MINGW64:/f/suho
unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ git restore list.txt

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


unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ cat list.txt
list.txt

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

Staged 파일 상태 - 8

- Staged 상태로 변경하면 하나의 수정 본으로 합쳐짐

MINGW64:/f/suho

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


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


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

Staged 파일 상태 - 9

- **합쳐진 상태에서 파일 내용을 확인**
  - **마지막 수정 상태인 "ls -a"의 내용으로 파일이 결정됨**
  - Staged **상태로 변경한 상태에서 추가 수정을 계속할 수 있음**
  - **단, 항상 마지막 파일 상태만 유지되기 때문에 주의해야 함**
  - **변경될 때마다** commit**하여 이전 상태를 보관하는 것이 좋음**

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


unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ cat list.txt
./
../
.git/
.gitignore
list.txt

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

Staged 파일 상태 - 10

- Staged **파일 상태 변경**
  - Modified **상태로 파일 상태를 변경하면 하나의 상태만 남음**
  - "git add"**로 합쳐지면 마지막 상태만 남아 파일 내용도 최종본만 남음**

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


unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ git restore --staged list.txt

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

Staged 파일 상태 - 11

- **다중 상태 파일** commit **수행**
  - Staged **상태의 파일은 관리 상태로 포함되고 수정 파일은 그대로 남음**

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


unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ git commit -m "test status"
[master 4ac085e] test status
 1 file changed, 1 insertion(+), 6 deletions(-)

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

Staged 파일 상태 - 12

- Staged 상태 정리
  - 읽기 전용 상태는 git의 관리 상태 중에 하나임을 의미
  - Staged 상태의 파일은 지속적인 수정과 상태 변경 가능
  - "git add"로 다중 상태를 갖는 파일을 준비 상태로 추가
    - 이전에 저장되었던 상태는 유지되지 않고 삭제됨
    - 마지막에 추가하는 파일의 내용이 준비 상태인 파일의 내용으로 변경
  - 원본과 Staged 파일과 Modified 파일 3개가 존재한다고 볼 수 있음
  - 즉, 한 파일의 관리를 위해 원본과 같은 2개 파일이 동시 존재할 수 있음
    - 실제로 편집 가능한 파일은 Modified 파일 하나만 가능
    - 수정이 발생한 파일은 항상 대상 파일 내용을 확인할 때마다 확인 가능
    - Committed 파일은 관리 중이므로 수정하기 전에만 확인 가능
    - Staged 파일은 별도 명령어를 통해 내용을 확인할 수 있음

Staged 파일 상태 - 13

- **변경된 파일 내용 확인**
  - "git diff" **명령으로** Staged **파일과** Committed **파일 차이 확인 가능**
  - **파일 차이를 통해** Committed **파일의 수정 사항을 미리 확인 가능**

MINGW64:/f/suho

```
unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (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/suho (master)
$ ls list.txt
list.txt

unangel@DESKTOP-I8OQG85 MINGW64 /f/suho (master)
$ git diff --cached list.txt
diff --git a/list.txt b/list.txt
index 9cffded..1210e07 100644
--- a/list.txt
+++ b/list.txt
@@ -1 +1,5 @@
+./
+../
+.git/
+.gitignore
 list.txt

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

정리 – 1

- git 명령어 사용 방법
  - 쉘에서 "git" 다음에 수행할 명령어를 입력하는 형태로 구성
- git은 3가지 파일 상태를 가짐
  - 관리(Committed), 수정(Modified), 준비(Staged) 상태
  - 파일과 디렉터리는 git에 의해 상태가 지속적으로 변경

- git 명령어
  - git status : 저장소의 현재 상태 정보를 확인
  - git add : 수정된(Modified) 파일을 준비(Staged) 상태로 변경
  - git rm : 새로운 파일의 준비 상태 파일을 수정 상태로 변경
  - git restore : 존재하던 파일의 준비 상태 파일을 수정 상태로 변경
  - git commit : 준비 상태의 파일을 관리(Committed) 상태로 변경
  - git log : git 저장소에 commit된 정보를 출력

정리 – 2

- 각 상태의 변경 명령어
  - Staged 파일 확인을 통해 각 상태는 서로 다른 파일로 동작 가능
  - 파일 관리에 최적화된 관리 방법임을 확인할 수 있음