Git

1. 도구 개요
2. 설치 및 실행
3. 주요 기능
4. 활용 예제

1. 도구 개요

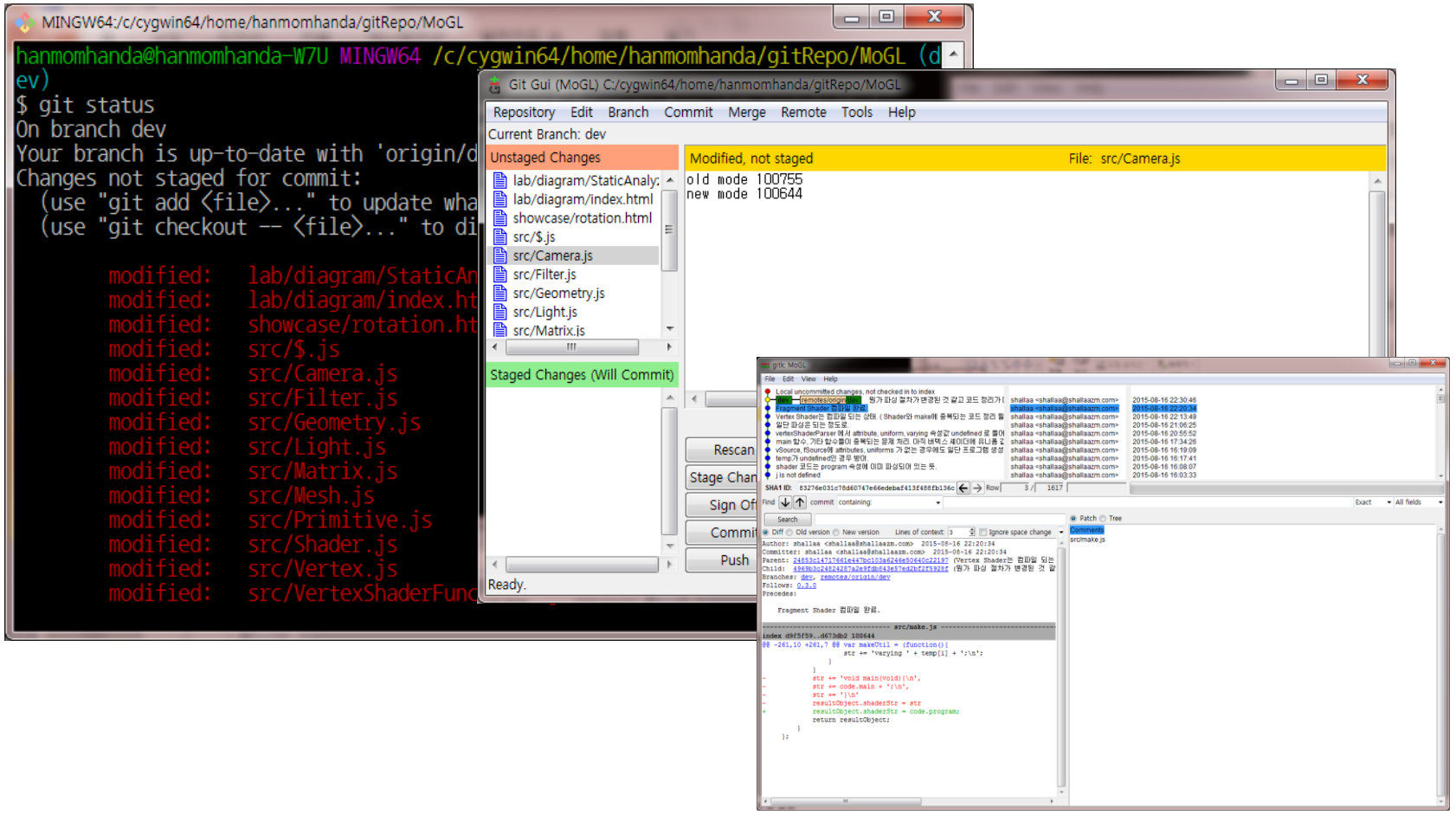
1.1 도구 정보 요약

도구명	Git (http://git-scm.com/)	라이선스	General Public License v2
소개	• 리누스 토발즈가 만든 분산형 버전 관리 시스템 • 대부분의 공개SW가 Git을 이용해서 관리되고 있음 • GitHub, BitBucket, GitLab 등 웹 기반의 다양한 소스 저장소 서비스의 기반 • 대부분의 IDE에서 Git 지원		
특징	• 분산형 버전관리시스템으로 Git사용자가 Git 저장소를 보유하고 원격과 동기화 • 거의 모든 명령을 로컬에서 수행하며, branch의 생성/전환/폐기가 빠르다 • 파일별 변화를 저장하는 다른 버전관리시스템과 달리 파일 시스템 스냅샷을 관리		
주요기능	• 소스 및 문서 버전 관리 서버 및 클라이언트 기능 • add, commit, reset, branch, checkout, merge, rebase 등 로컬에서의 작업 • push, pull. fetch 등 원격 작업		
실행환경	• Windows, Linux, MacOS	사전설치도구	• 해당 없음
카테고리	• 형상관리	최신버전	• v2.6.2 (2015.10)
관련도구	• CVS, SVN, EGit		

1. 도구 개요

1.2 스크린 캡쳐 및 주요 기능

- 터미널에서 git 명령을 실행할 수 있는 Git Bash
- GUI 환경에서 git 명령을 실행할 수 있는 Git GUI



2. 설치 및 실행

세부 목차

2. 설치 및 실행

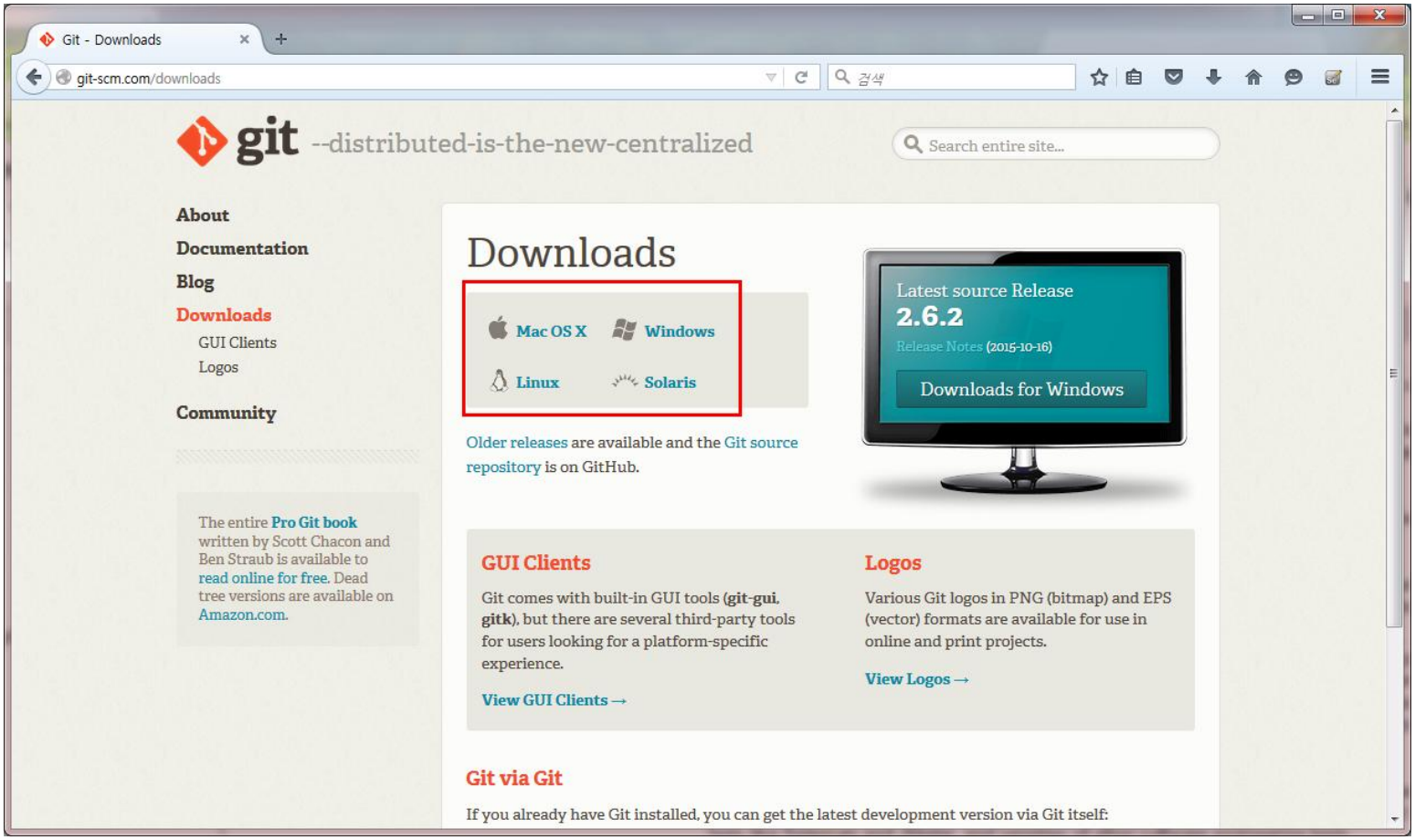
2.1 사전 설치 사항 확인

- Git는 별도의 사전 설치 도구가 필요 없다.
- Linux에서는 배포판 별 설치 도구를 통해 설치할 수 있으며,
- Windows와 MacOSX 용 설치 파일을 다운로드 해서 설치할 수 있다.

2. 설치 및 실행

2.2 다운로드 (1/4)

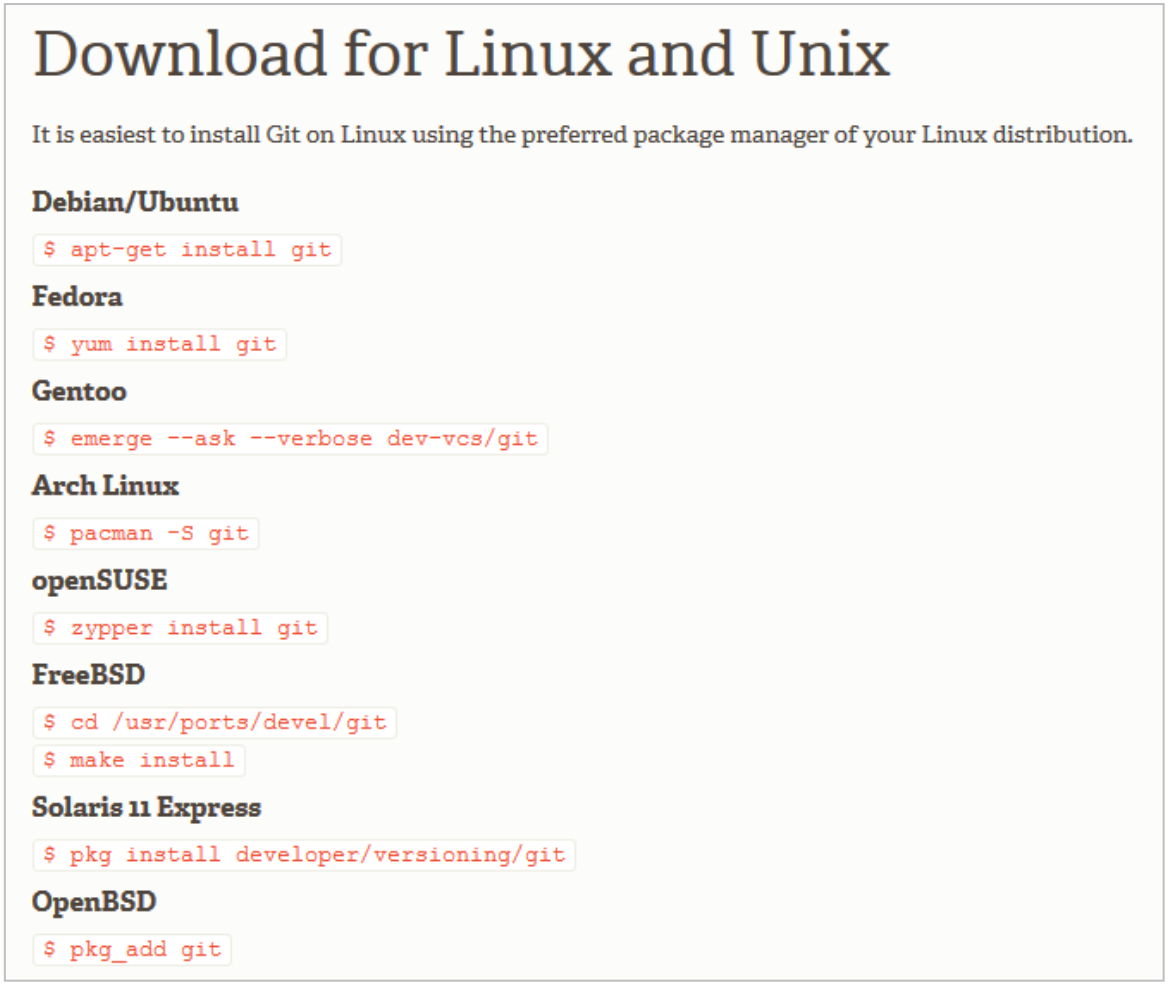
- http://git-scm.com/ 에 접속
- 해당 OS 클릭



2. 설치 및 실행

2.2 다운로드 (2/4)

- Linux 에서의 설치
- 터미널에서 배포판에 맞는 설치 명령 실행



Download for Linux and Unix

It is easiest to install Git on Linux using the preferred package manager of your Linux distribution.

Debian/Ubuntu

```
$ apt-get install git
```

Fedora

```
$ yum install git
```

Gentoo

```
$ emerge --ask --verbose dev-vcs/git
```

Arch Linux

```
$ pacman -S git
```

openSUSE

```
$ zypper install git
```

FreeBSD

```
$ cd /usr/ports/devel/git
$ make install
```

Solaris 11 Express

```
$ pkg install developer/versioning/git
```

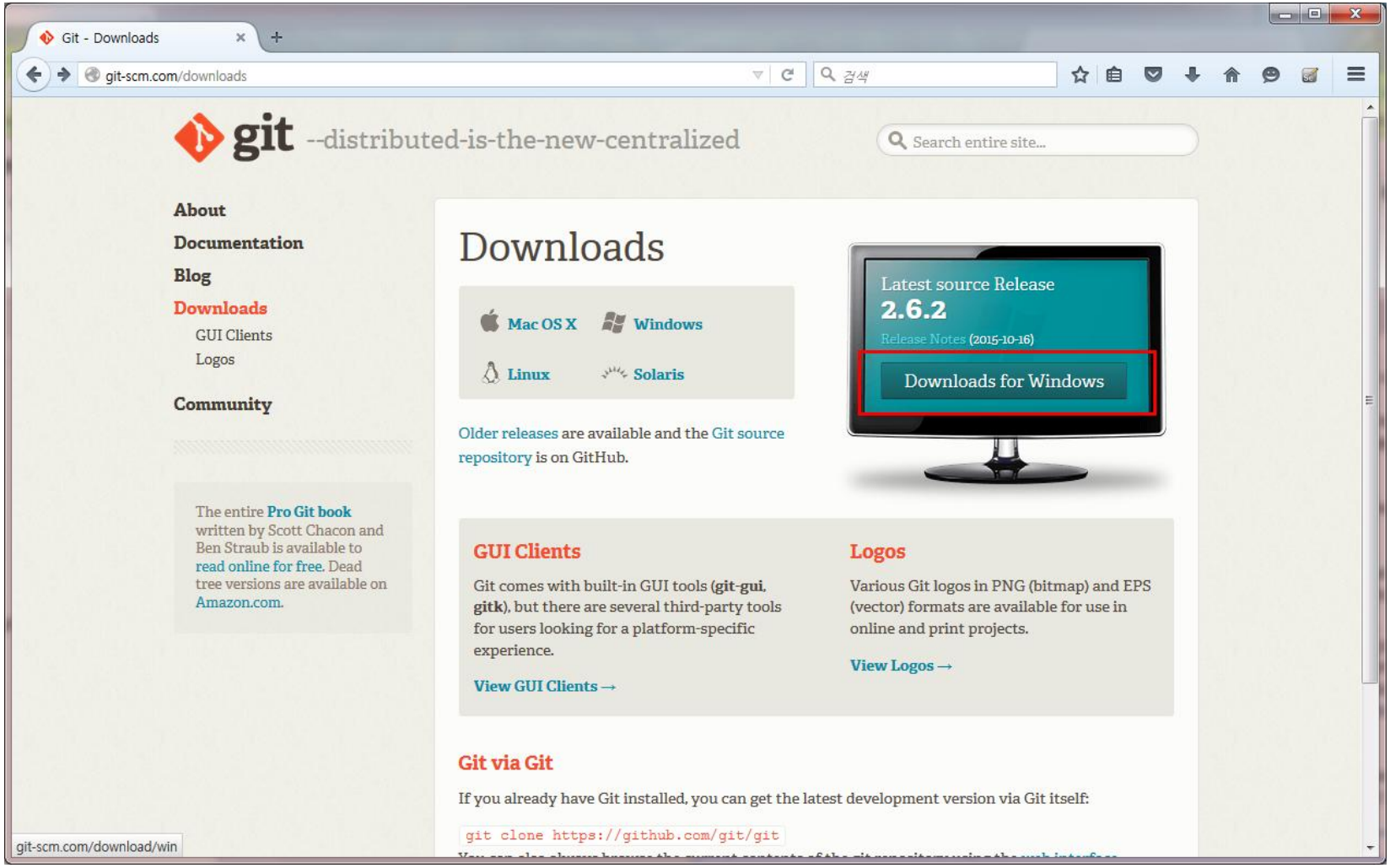
OpenBSD

```
$ pkg_add git
```

2. 설치 및 실행

2.2 다운로드 (3/4)

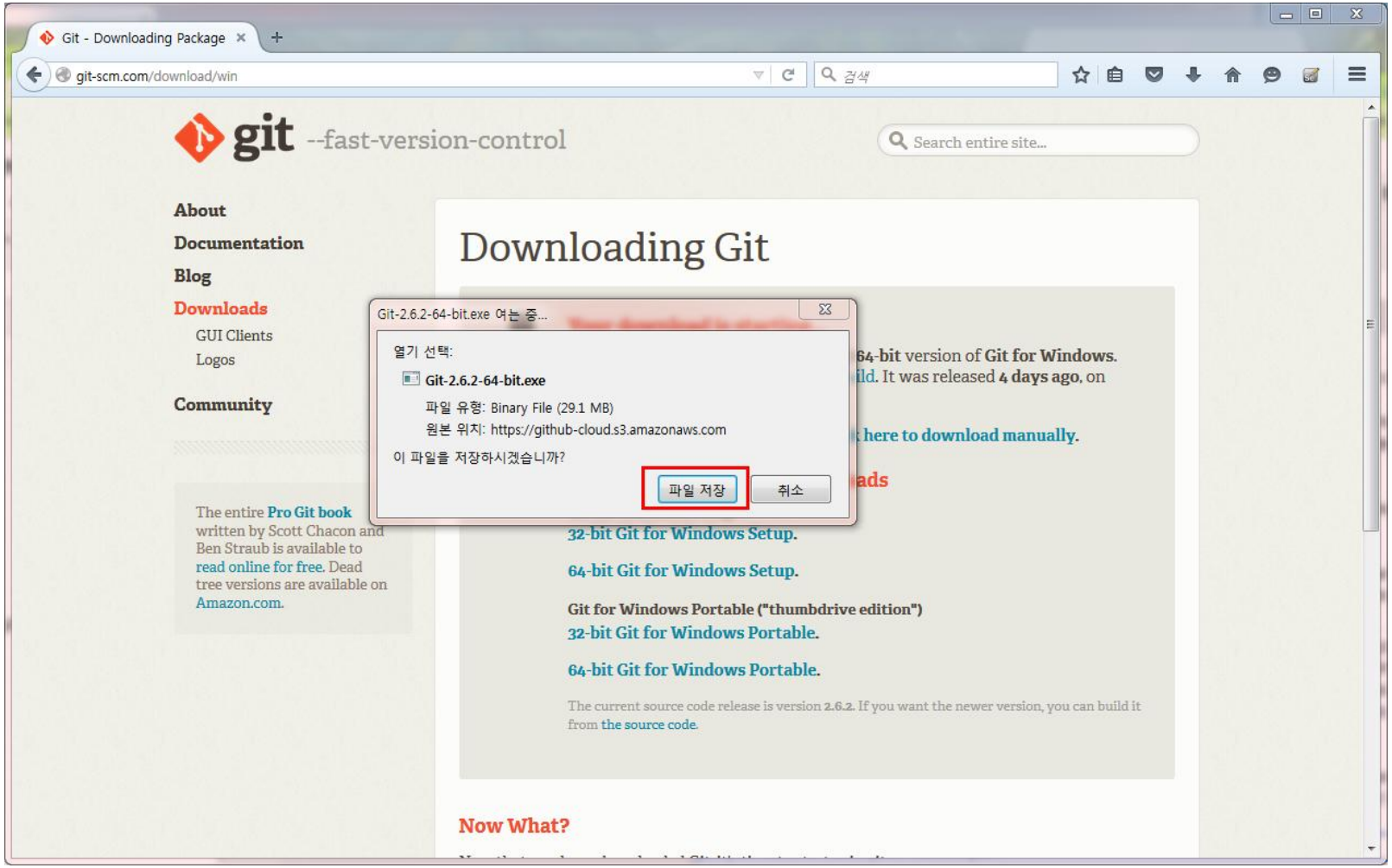
- Windows 에서의 설치



2. 설치 및 실행

2.2 다운로드 (4/4)

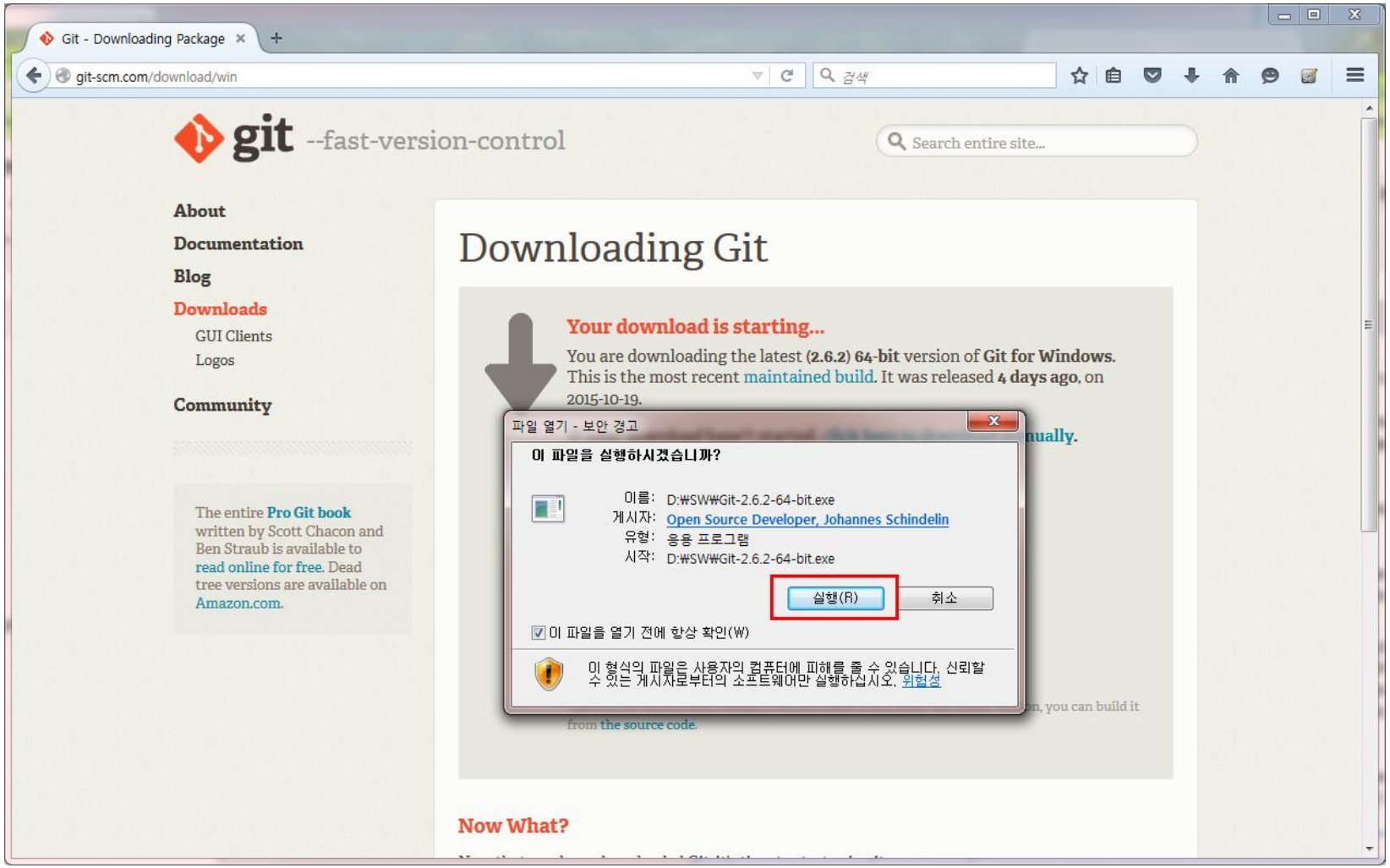
- 설치 파일 다운로드



2. 설치 및 실행

2.2 설치 (1/8)

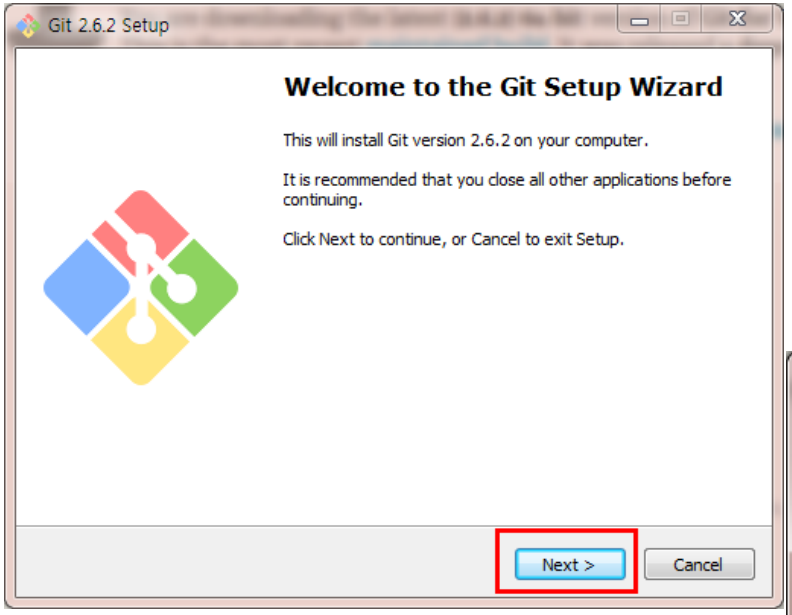
- 설치 파일 실행



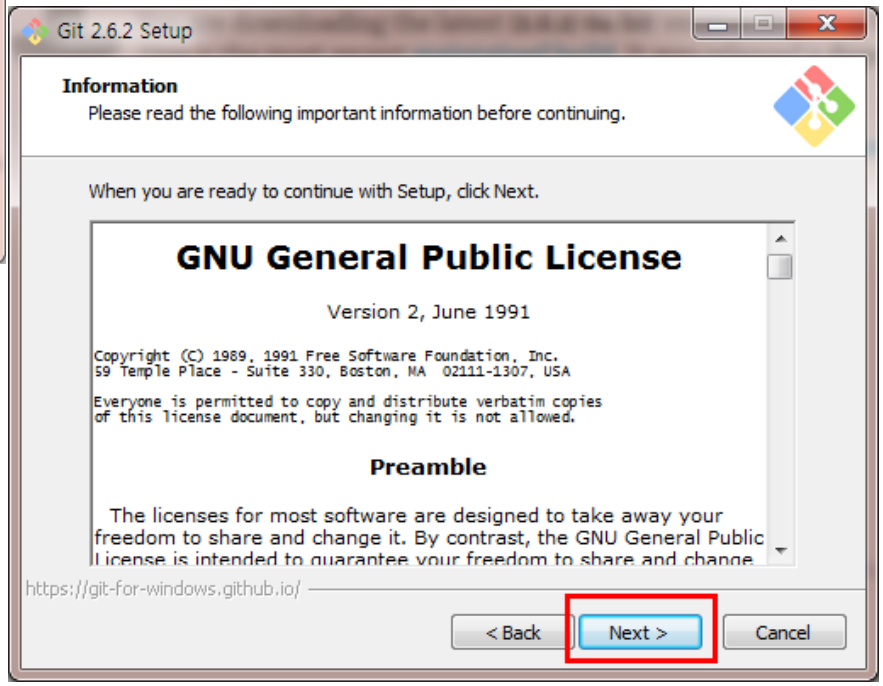
2. 설치 및 실행

2.2 설치 (2/8)

- 설치 마법사 시작



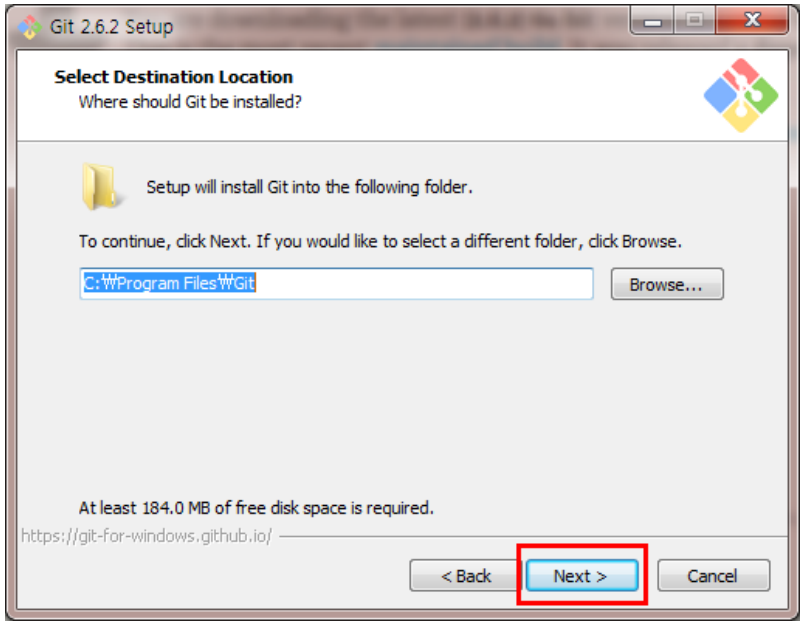
- 라이선스 동의



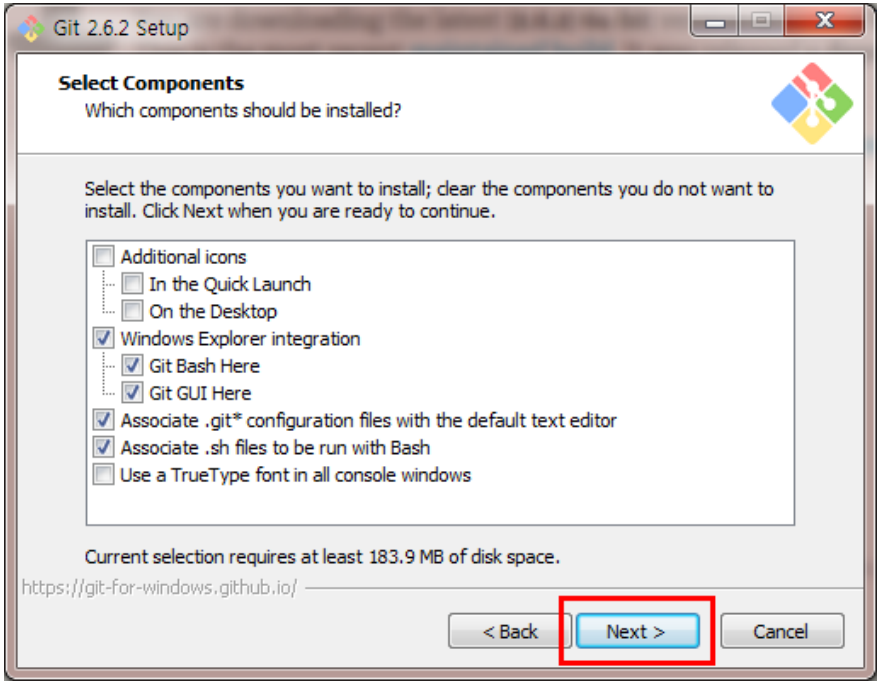
2. 설치 및 실행

2.2 설치 (3/8)

- 설치 위치 지정



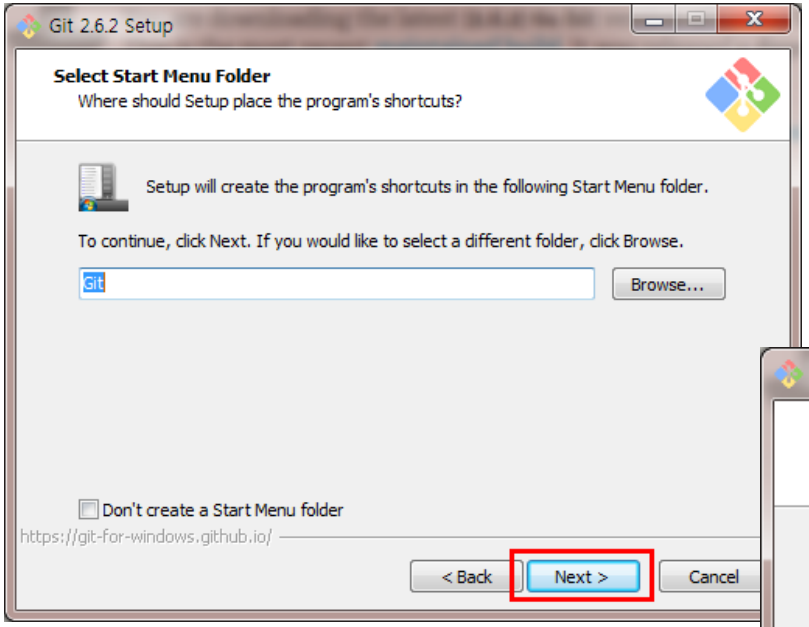
- 설치 옵션 지정 – 기본값으로 설치



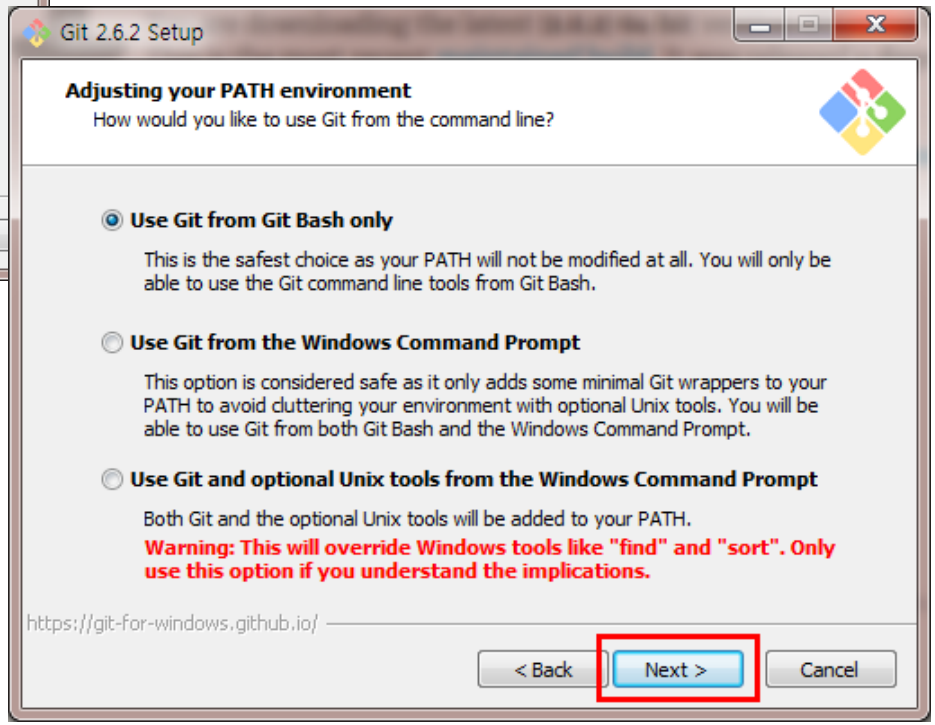
2. 설치 및 실행

2.2 설치 (4/8)

- 시작 프로그램 등록



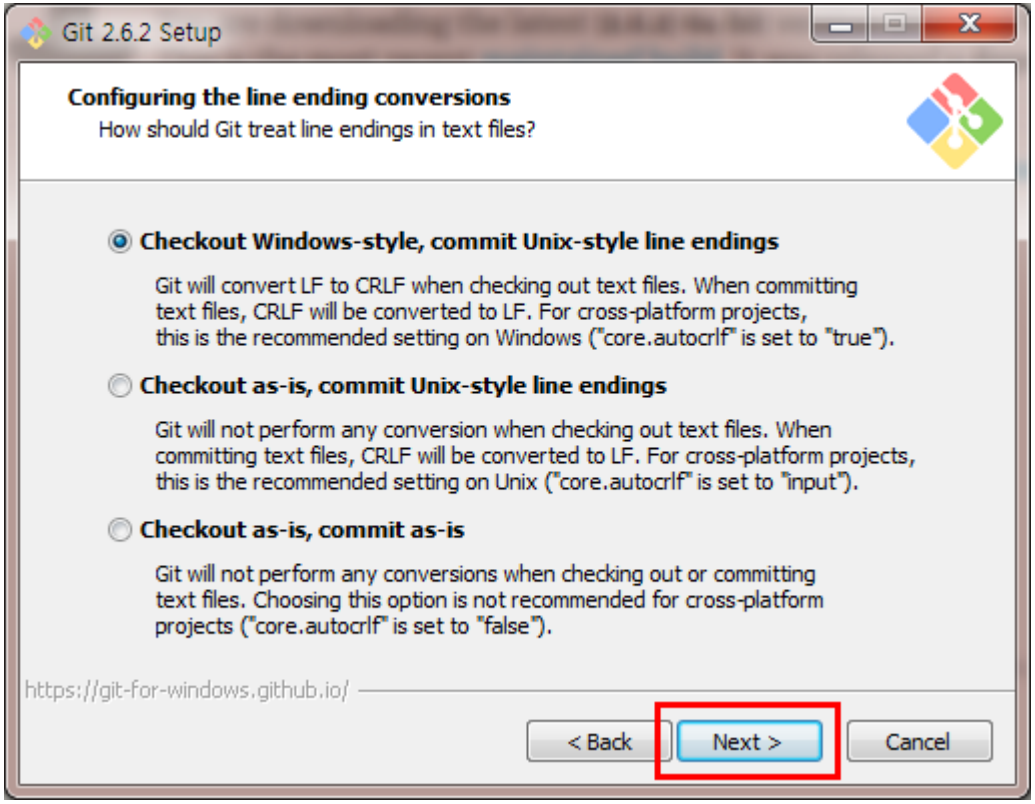
- PATH 환경 변수 설정 – 기본값으로 설치



2. 설치 및 실행

2.2 설치 (5/8)

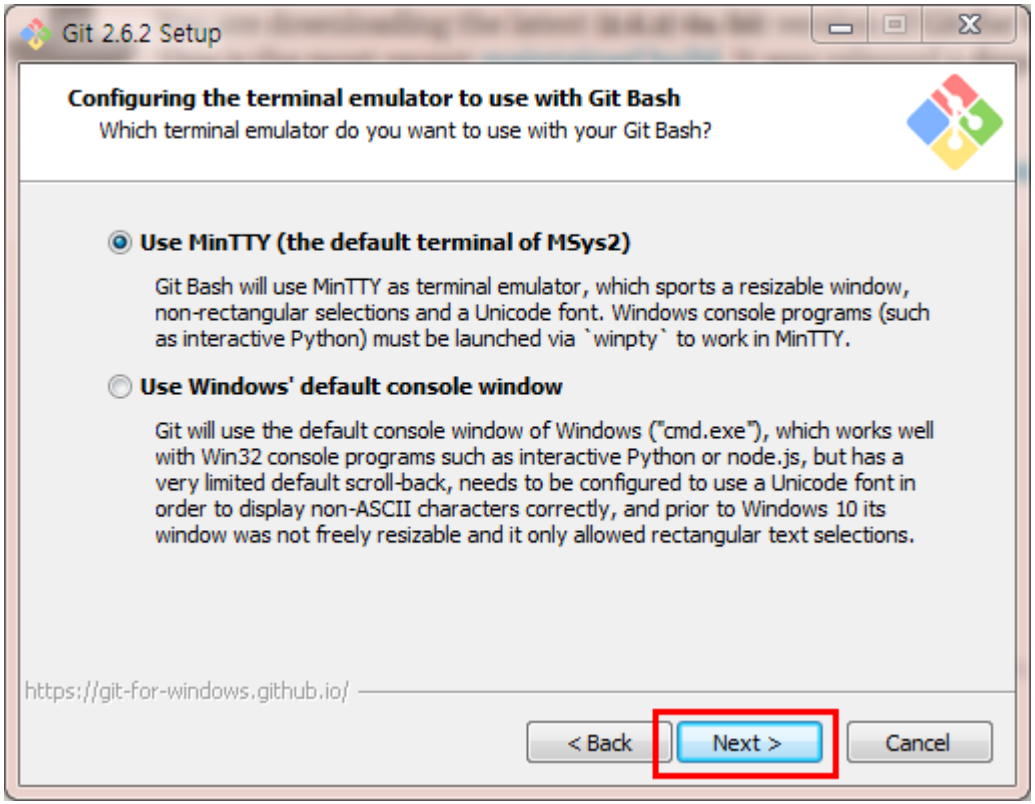
- 개행 문자 설정
  - Checkout Windows-style, commit Unix-style line endings
    - checkout 할 때 CRLF로 가져오고, commit 할 때 LF로 변환



2. 설치 및 실행

2.2 설치 (6/8)

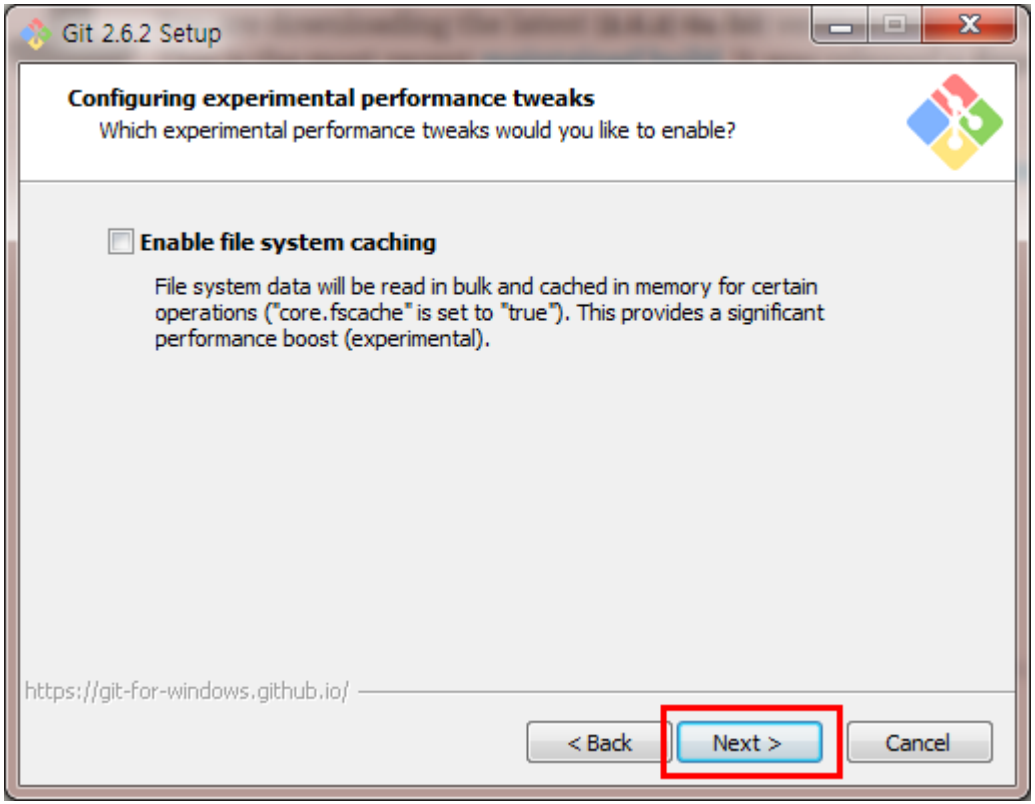
- 실행 터미널 지정
  - Use MinTTY : MinTTY 터미널 새로 설치
  - Use Windows' default console window : 윈도우 기본 명령창에서 git 실행



2. 설치 및 실행

2.2 설치 (7/8)

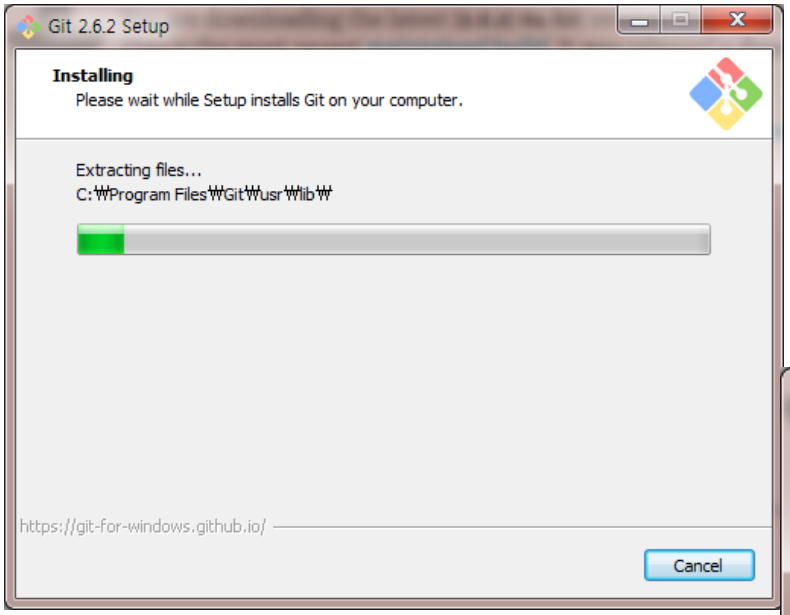
- 파일 시스템 캐쉬 기능
  - 파일 시스템 데이터를 대량으로 읽어서 메모리에 캐쉬
  - 성능을 대폭 향상 시킬 수 있으나 아직 실험단계 기능이므로, 선택하지 않고 기본값 그대로 설치한다.



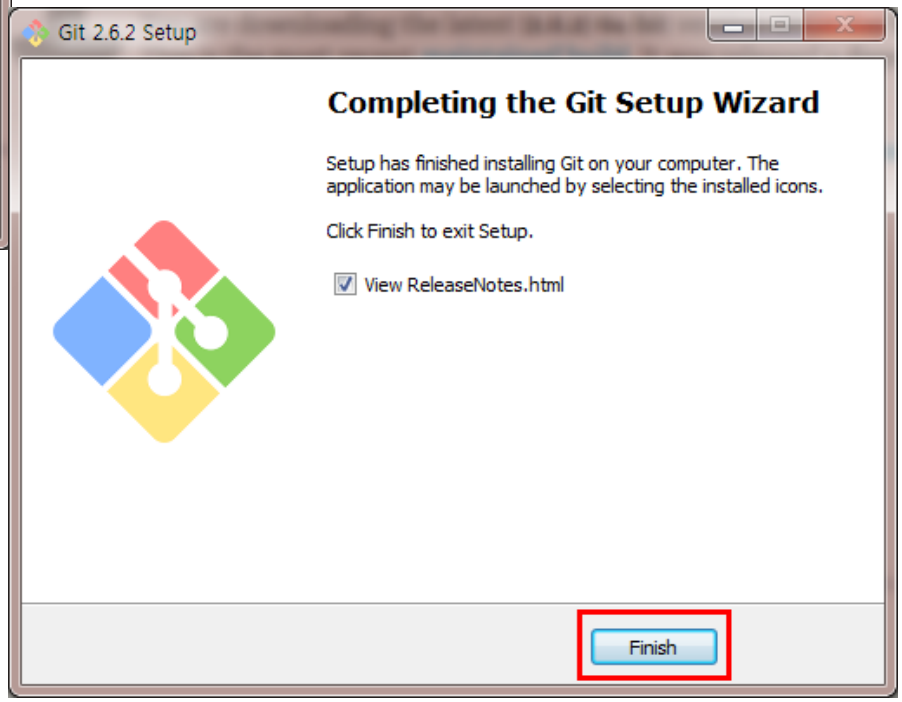
2. 설치 및 실행

2.2 설치 (8/8)

- 설치 시작



- 설치 완료
  - 릴리스 노트를 보지 않으려면 체크를 해제한다.



3. 주요 기능

세부 목차

3. 주요 기능

3.1 git 설정 파일

- git 설정 파일은 3단계로 구성되어 있다.
  - /etc/gitconfig : 시스템 모든 사용자와 모든 git 저장소에 적용
    - git config --system 은 이 파일을 수정
  - ~/.gitconfig 또는 ~/.config/git/config : 특정 사용자에게만 적용
    - git config --global 은 이 파일을 수정
  - git_저장소_디렉토리/.git/config : 특정 git 저장소에만 적용
    - git config 는 이 파일을 수정
  - 설정 파일을 직접 수정 또는 git config 명령으로 설정 가능

- 구체성이 높은 것이 우선 적용된다.
  - git_저장소_디렉토리/.git/config 의 설정 사항이 가장 우선 적용되고
  - /etc/gitconfig 의 설정 사항은 우선순위가 가장 낮다

- 윈도우에서는 $HOME(보통 C:₩Users₩$USER)/gitconfig 와 git_저장소_디렉토리/.git/config 이렇게 두 가지의 설정 파일이 존재한다.

3. 주요 기능

3.2 버전 관리 대상 제외

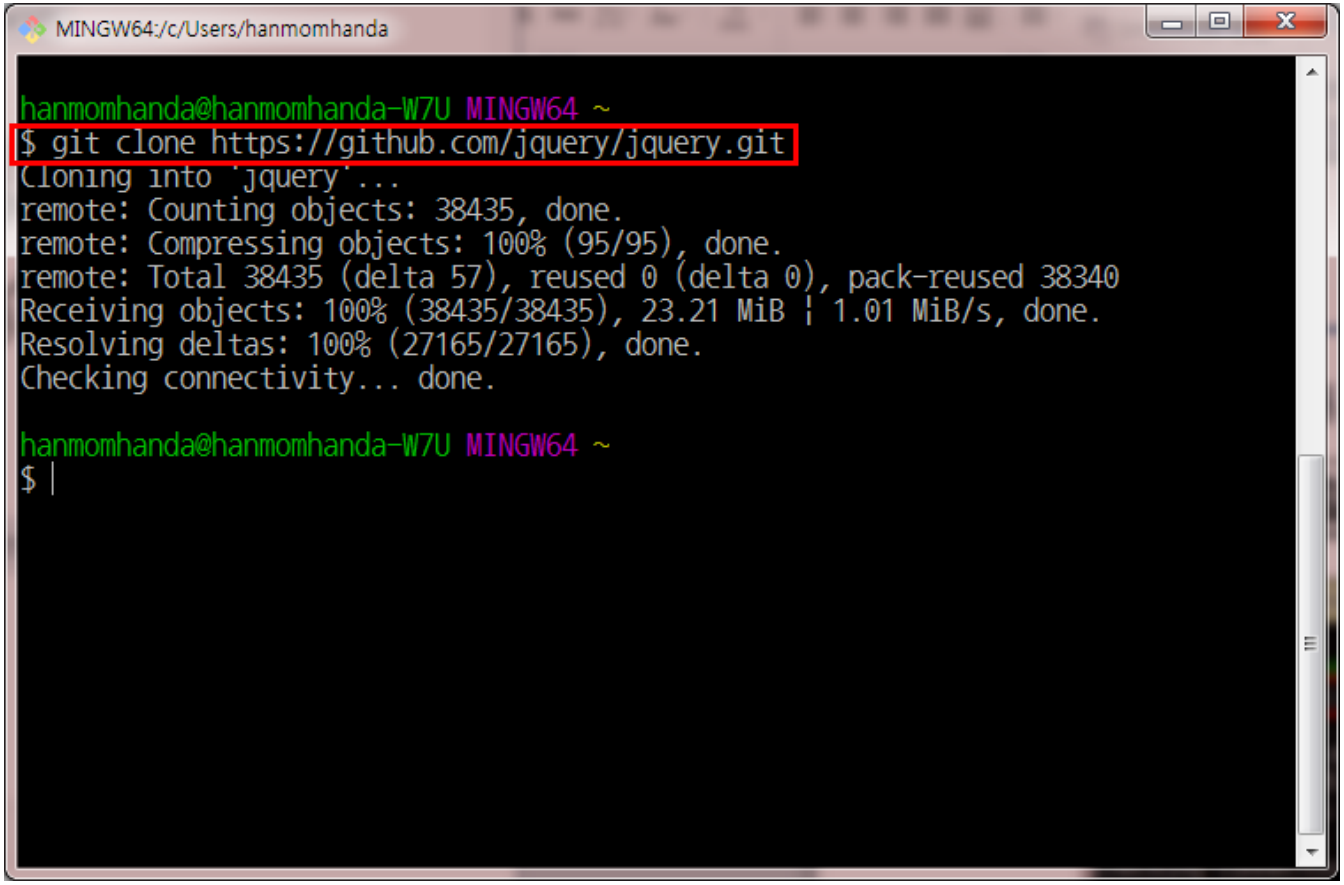
- .gitignore 파일을 통해 특정 파일이나 디렉토리를 버전 관리 대상에서 제외 가능
- 아래의 파일에 설정된 모든 항목은 버전 관리 대상에서 제외됨
  - ~/.config/git/gitignore
  - git_저장소_디렉토리/.gitignore
  - git_저장소_디렉토리/.git/info/exclude
  - git 설정파일의 core.excludesFile 에 명시된 파일
- 주로 class 파일 등 컴파일된 파일이나 컴파일된 파일이 모여있는 디렉토리를 버전 관리 대상에서 제외한다.

3. 주요 기능

3.3 git clone

- git 저장소를 복제해서 새 저장소를 생성한다.

```
git clone 복제할_git_저장소_url
```

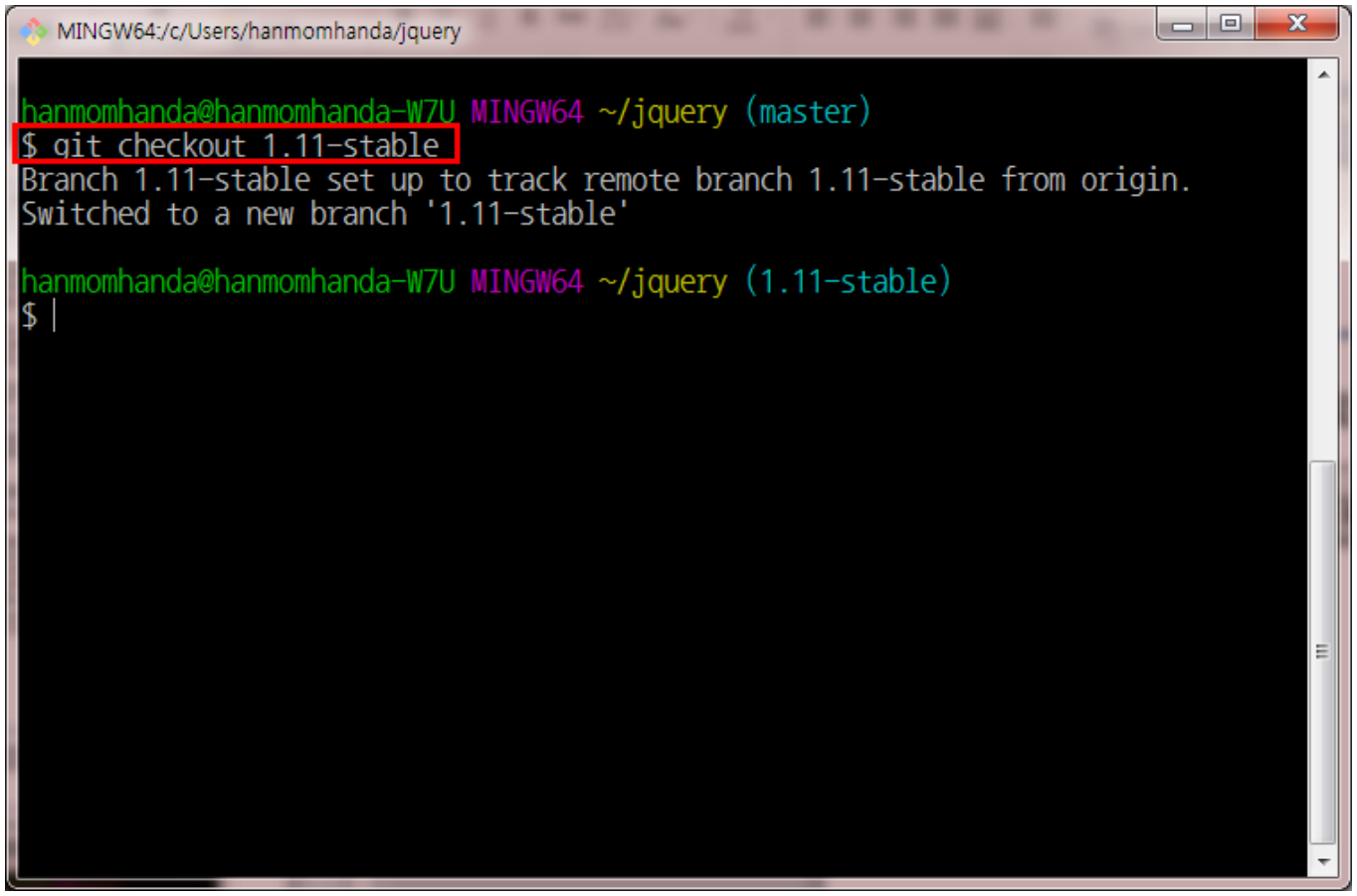


3. 주요 기능

3.4 git checkout (1/2)

- 다른 브랜치나 tag로 이동

```
git checkout branch_또는_tag_이름
```

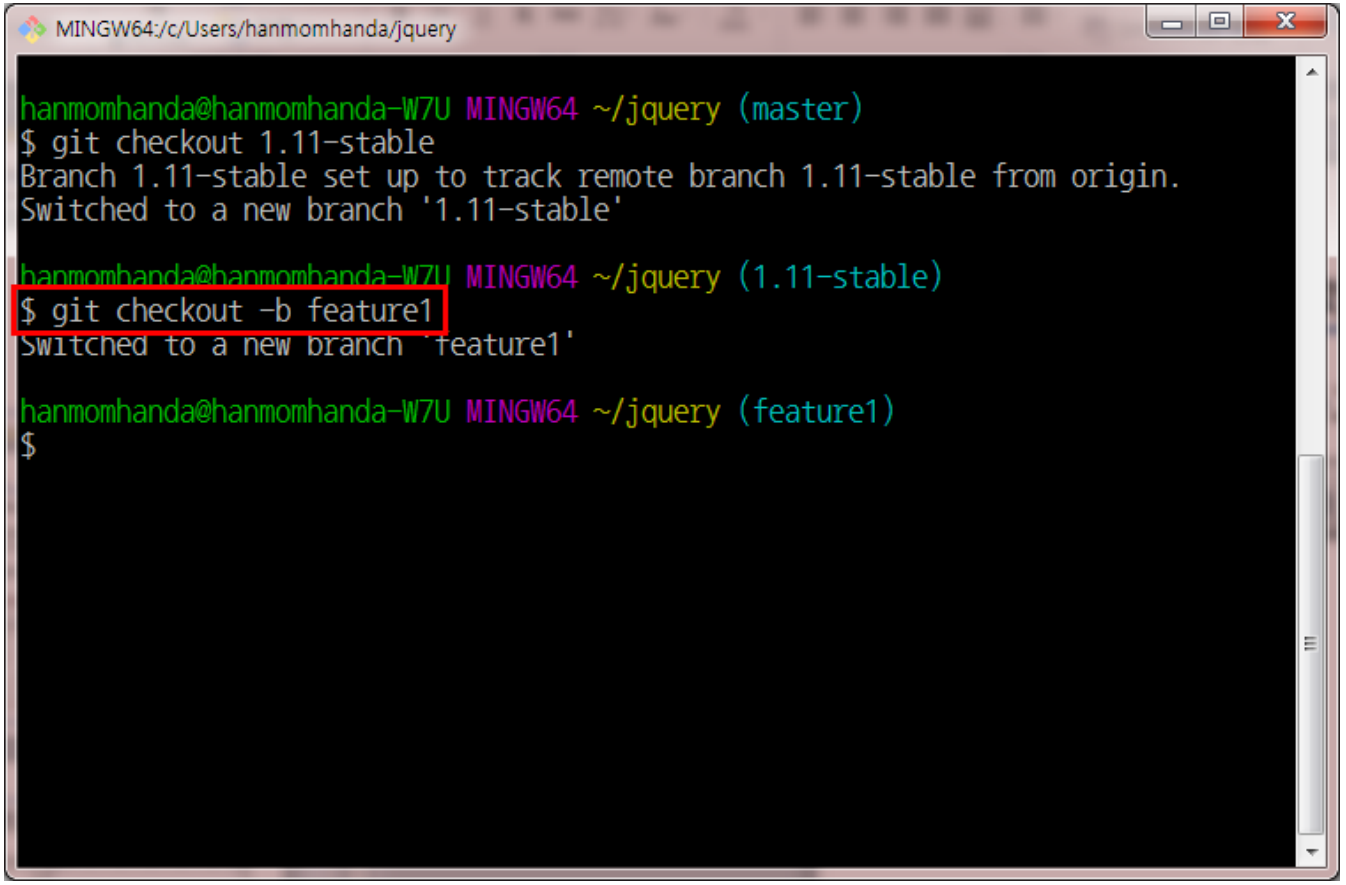


3. 주요 기능

3.4 git checkout (2/2)

- 새 브랜치를 생성하고 새 브랜치로 이동

```
git checkout -b 새branch_이름
```

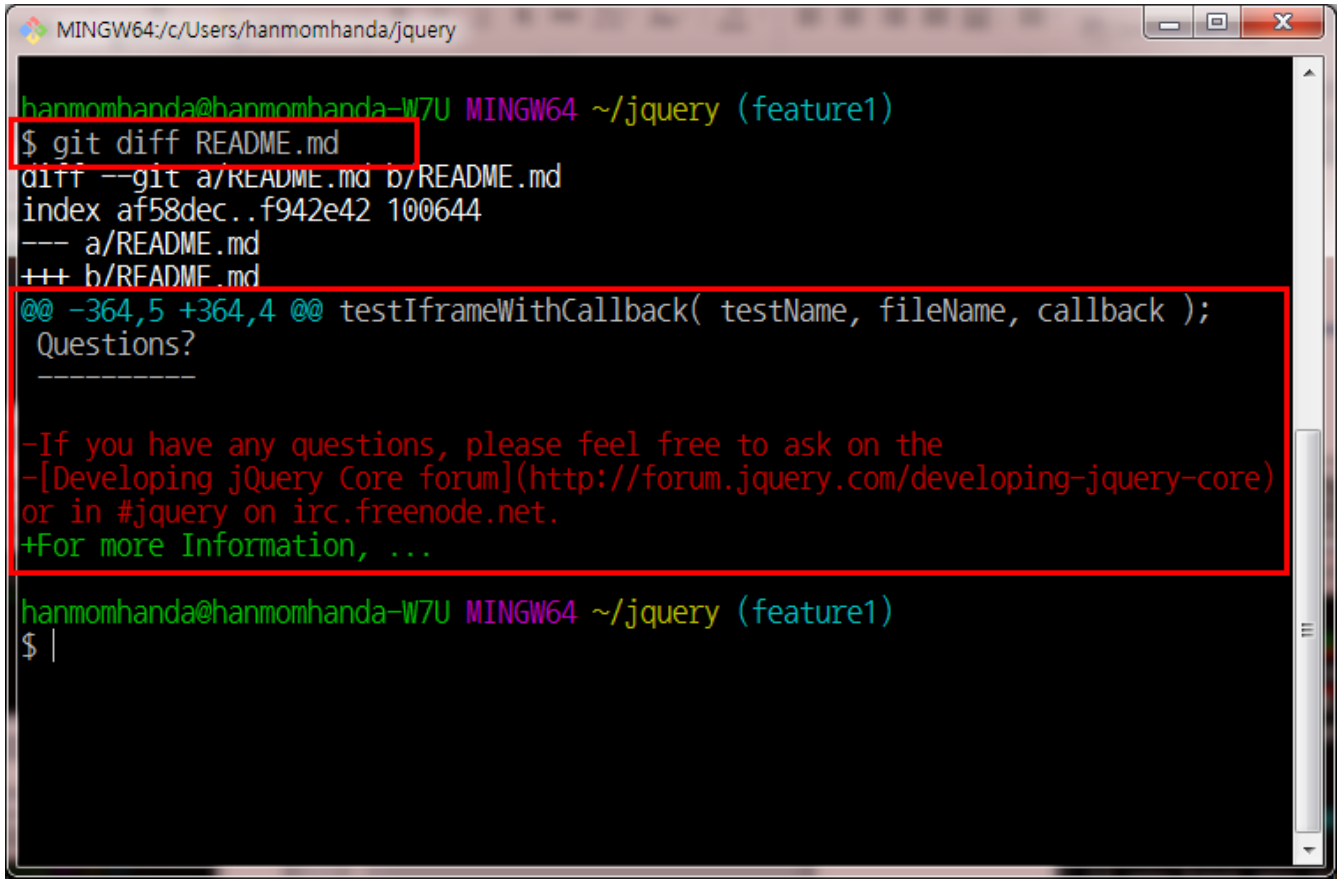


3. 주요 기능

3.5 git diff

- 수정 내용 비교

```
git diff [파일경로]
```

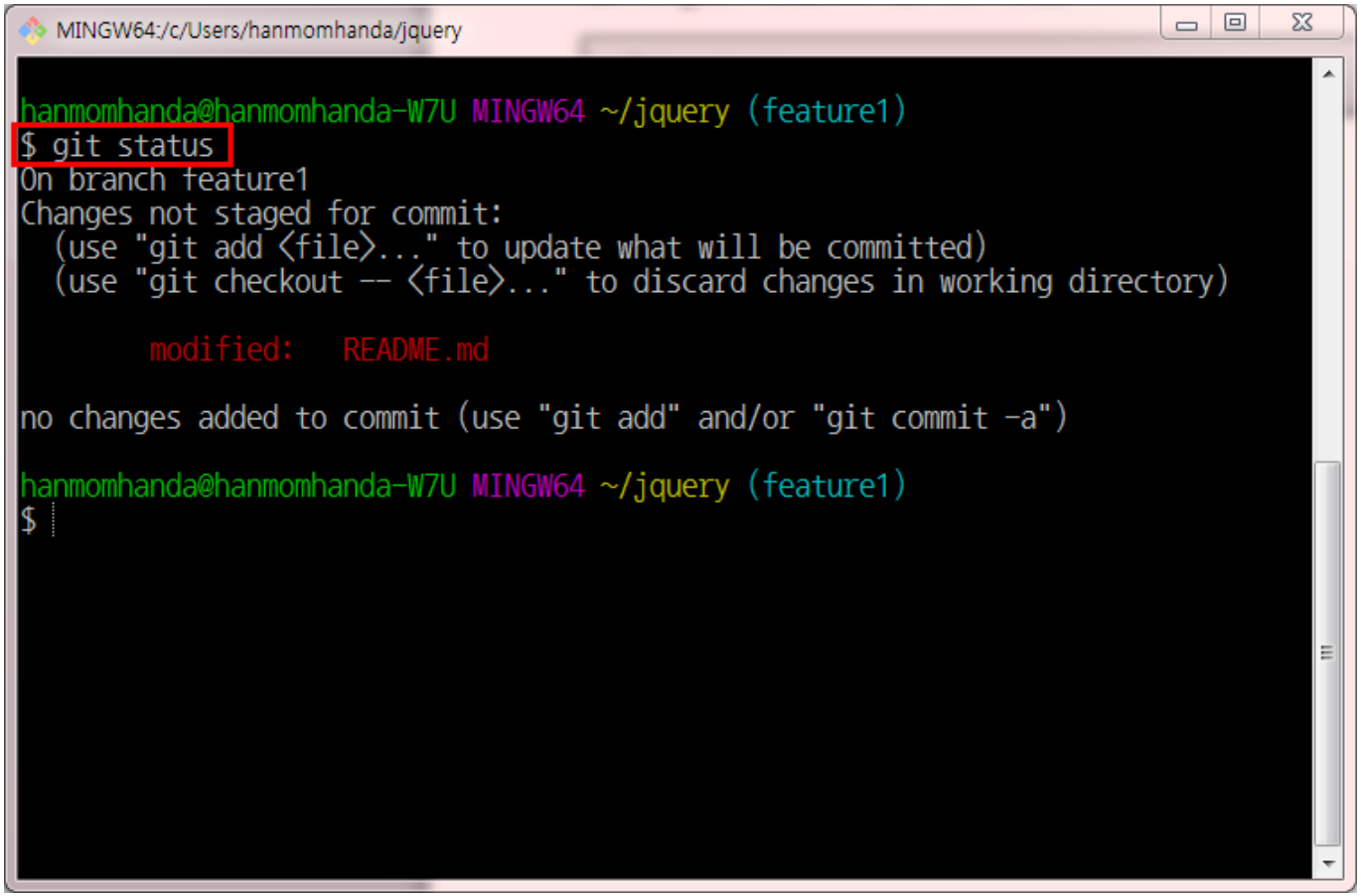


3. 주요 기능

3.6 git status

- 현재 git 저장소 상태 확인

```
git status
```



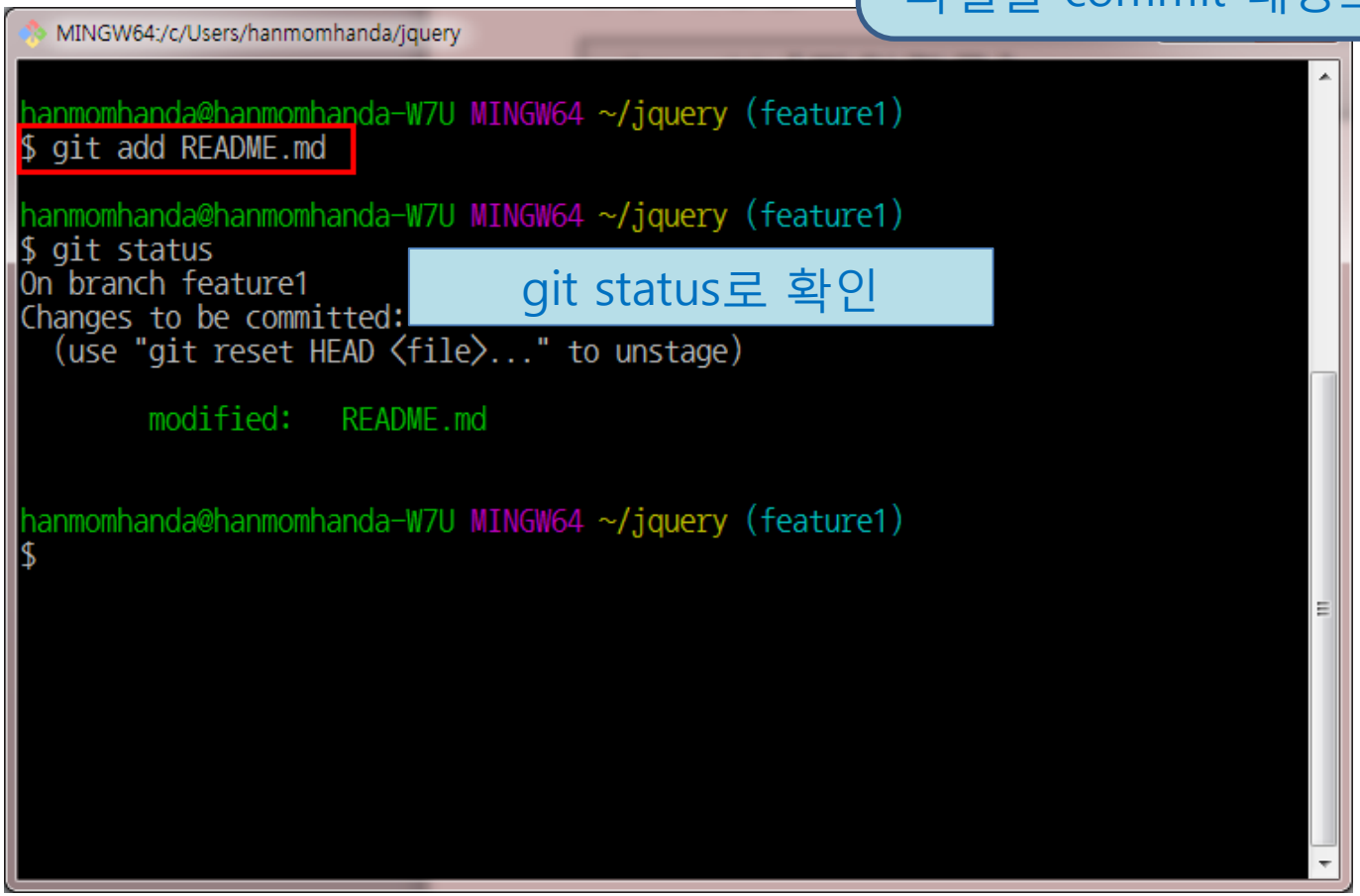
3. 주요 기능

3.7 git add

- 수정 내용을 commit 대상으로 등록

```
git add 파일경로_또는_디렉터리
```

디렉터리를 지정하면
디렉터리 내의 수정 중인 모든
파일을 commit 대상으로 등록

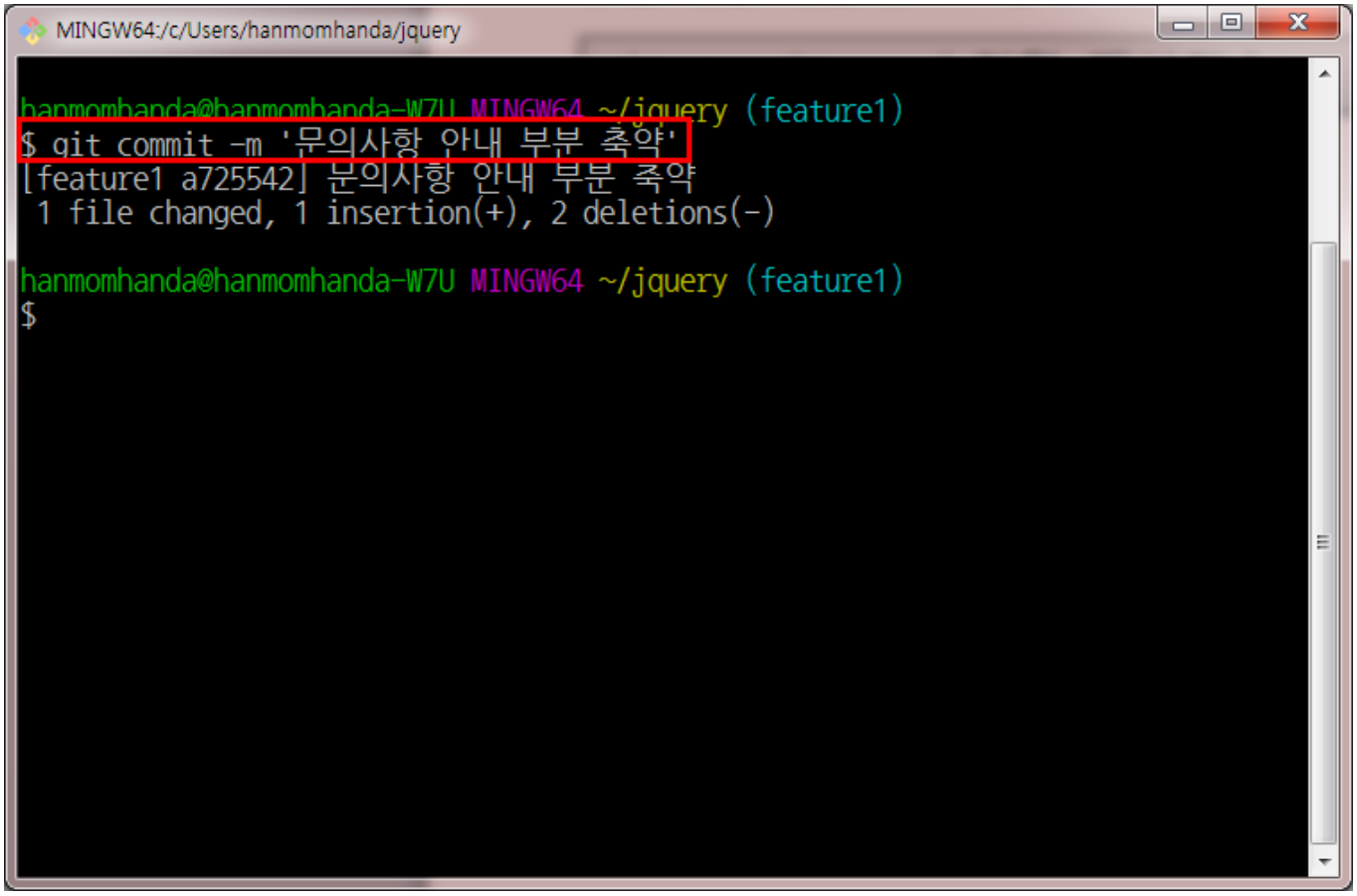


3. 주요 기능

3.8 git commit

- git add로 등록되어 있는 수정 내역을 한 묶음으로 해서 확정적으로 반영

```
git commit -m  '커밋 메시지'
```

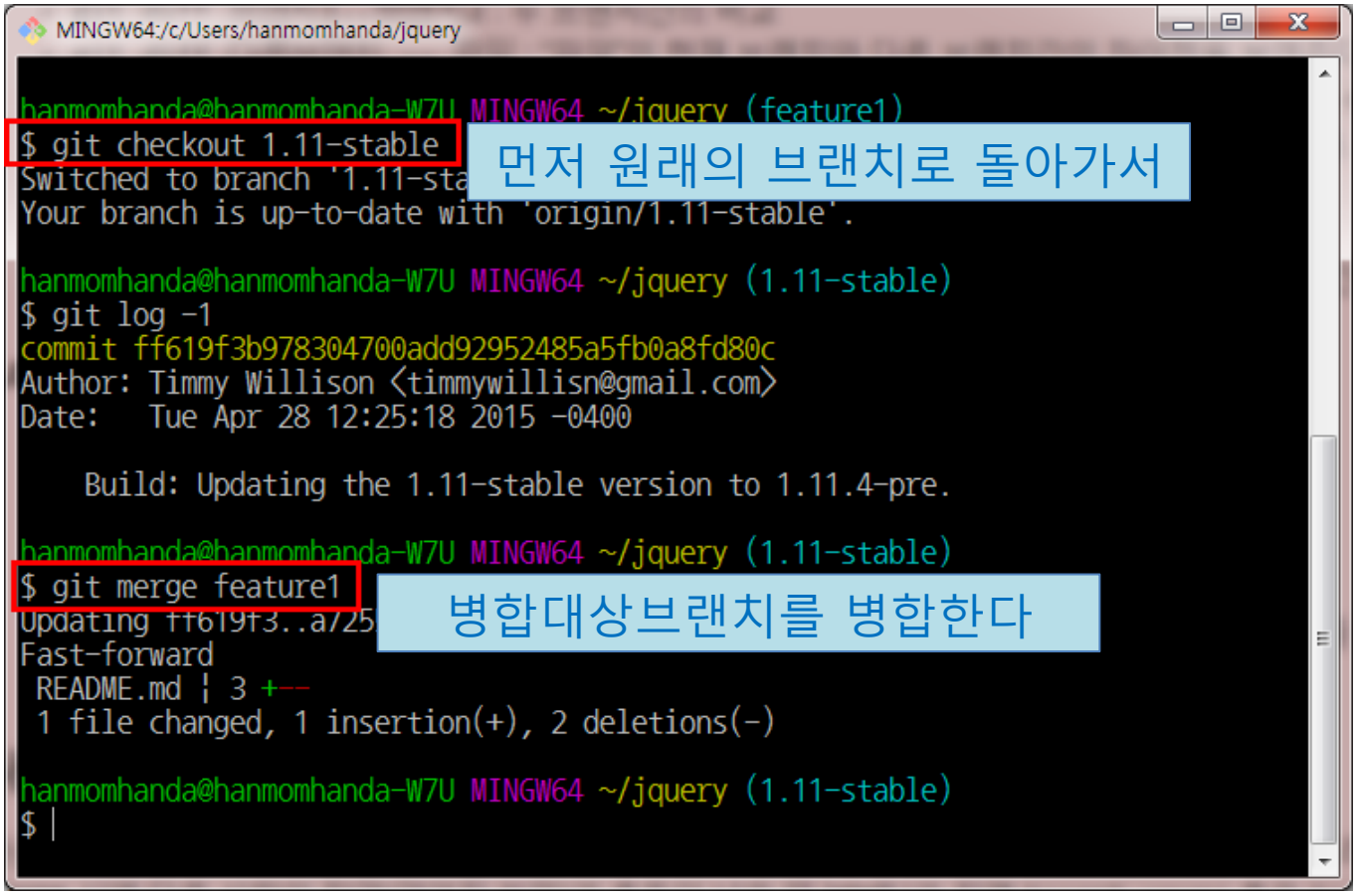


3. 주요 기능

3.9 git merge (1/2)

- 다른 브랜치를 병합

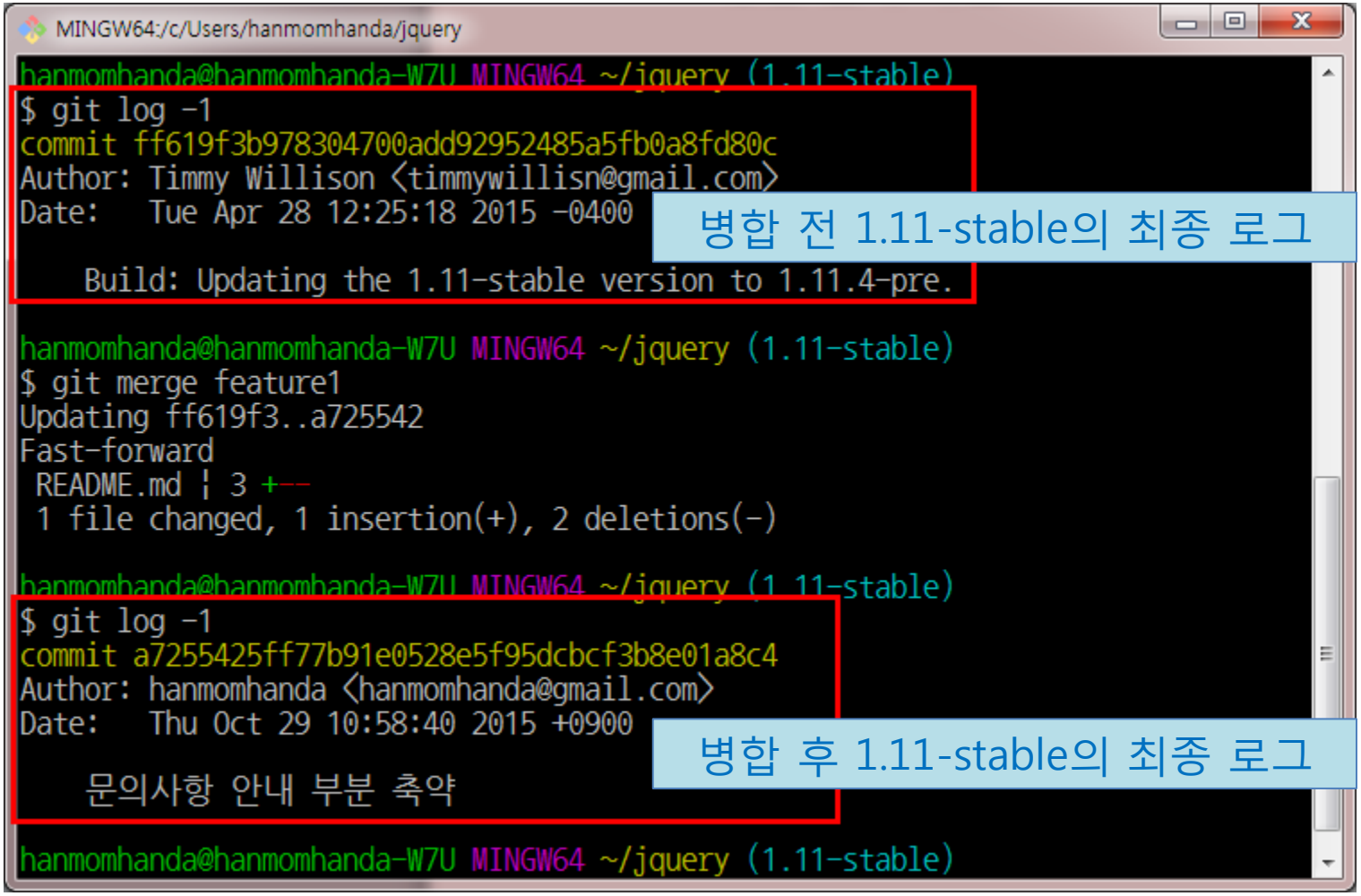
```
git merge 병합대상브랜치명
```



3. 주요 기능

3.9 git merge (2/2)

- 로그로 병합 결과 확인

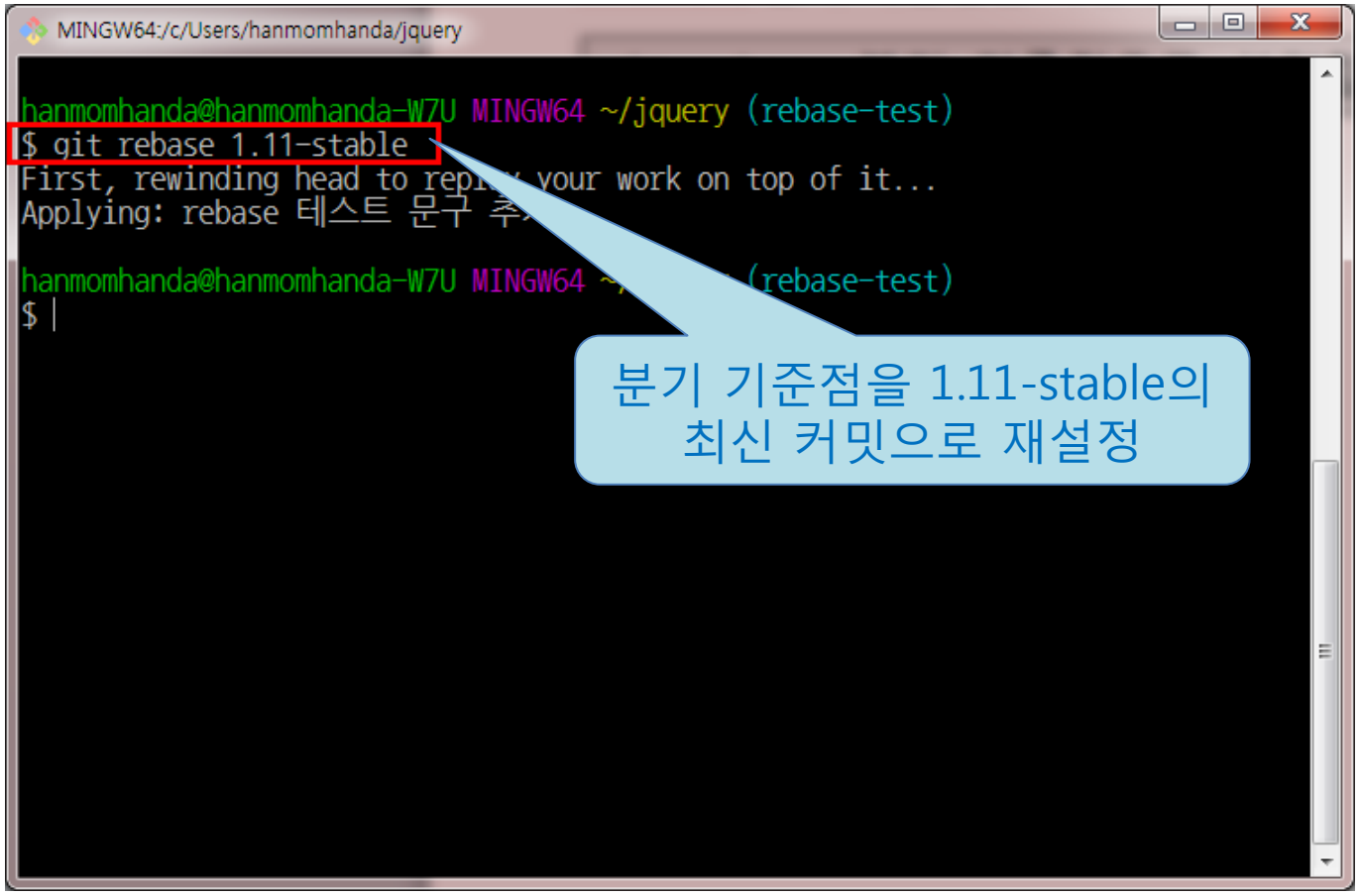


3. 주요 기능

3.10 git rebase (1/2)

- 현재 브랜치의 분기 기준점(base)을 재설정

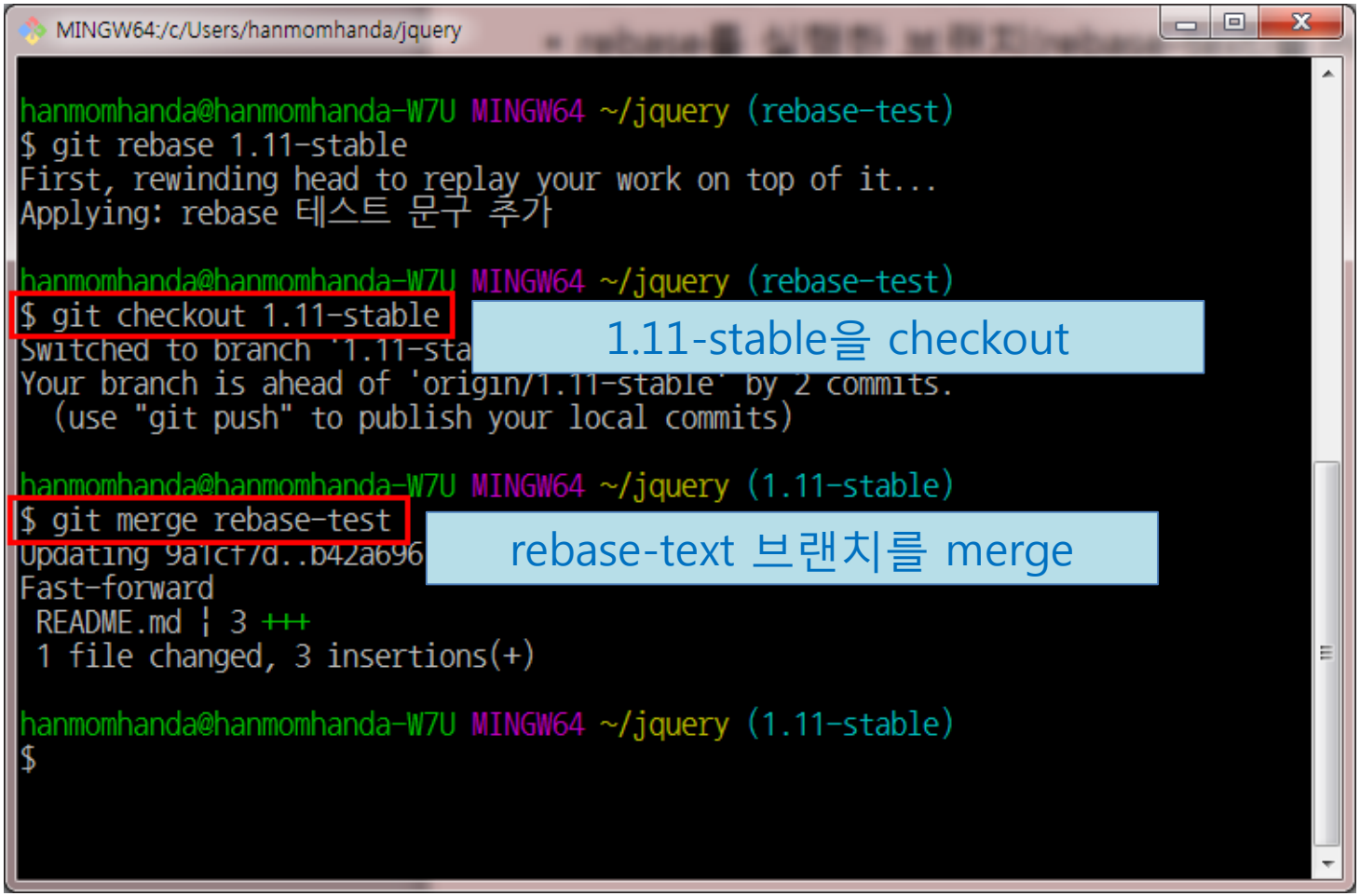
```
git rebase 분기_기준점으로_설정할_브랜치명
```



3. 주요 기능

3.10 git rebase (2/2)

- 분기 기준점으로 재설정된 브랜치(1.11-stable)를 checkout 해서,
- rebase를 실행한 브랜치(rebase-text)를 merge 해야함

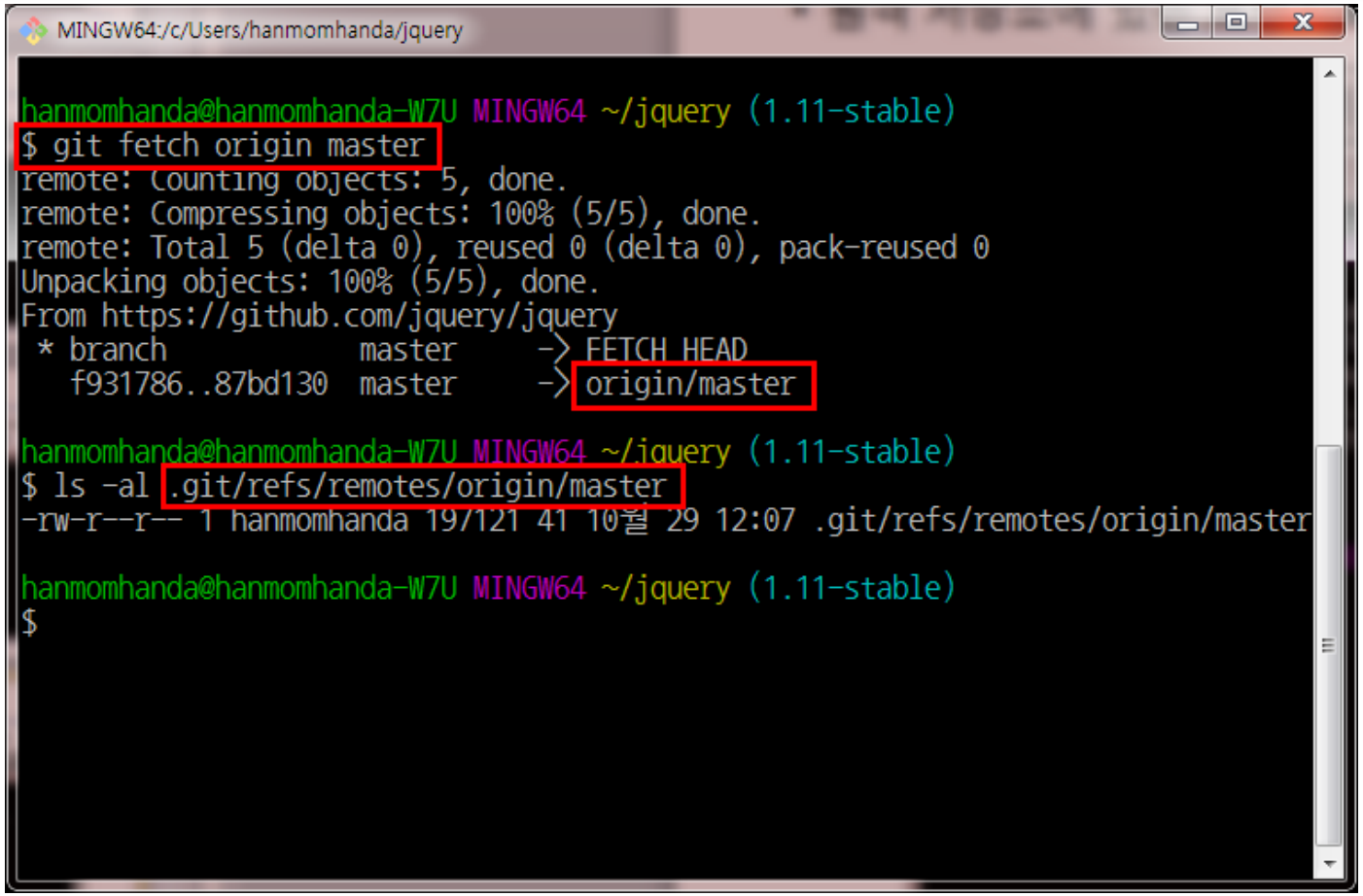


3. 주요 기능

3.11 git fetch

- 원격 저장소에 있는 브랜치를 로컬 저장소로 가져온다.

```
git fetch 원격저장소이름 [원격저장소_내의_브랜치이름]
```

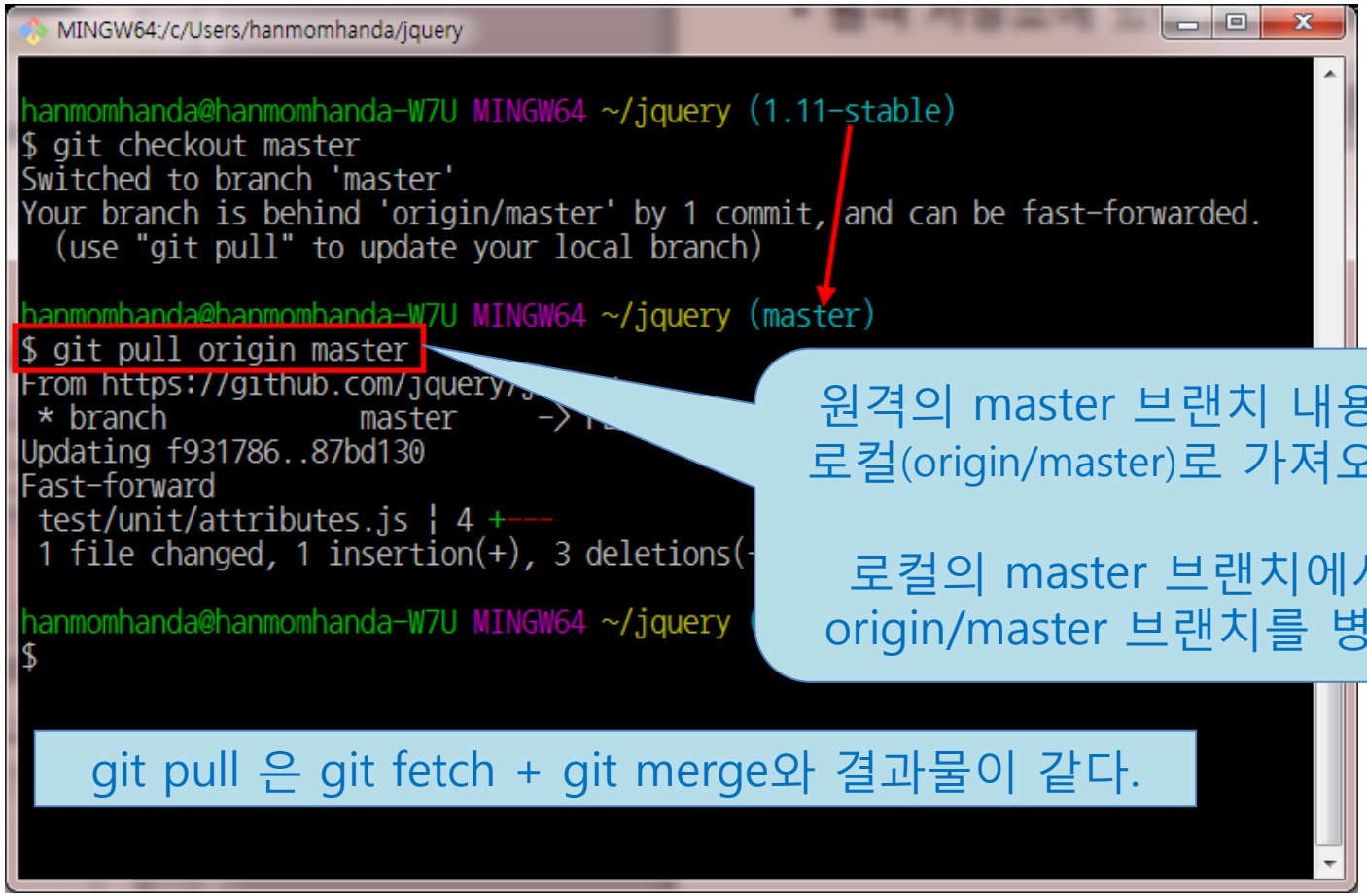


3. 주요 기능

3.12 git pull

- 원격 저장소에 있는 브랜치를 로컬 저장소로 가져와서 merge 한다.

```
git pull 원격저장소이름 원격저장소_내의_브랜치이름
```

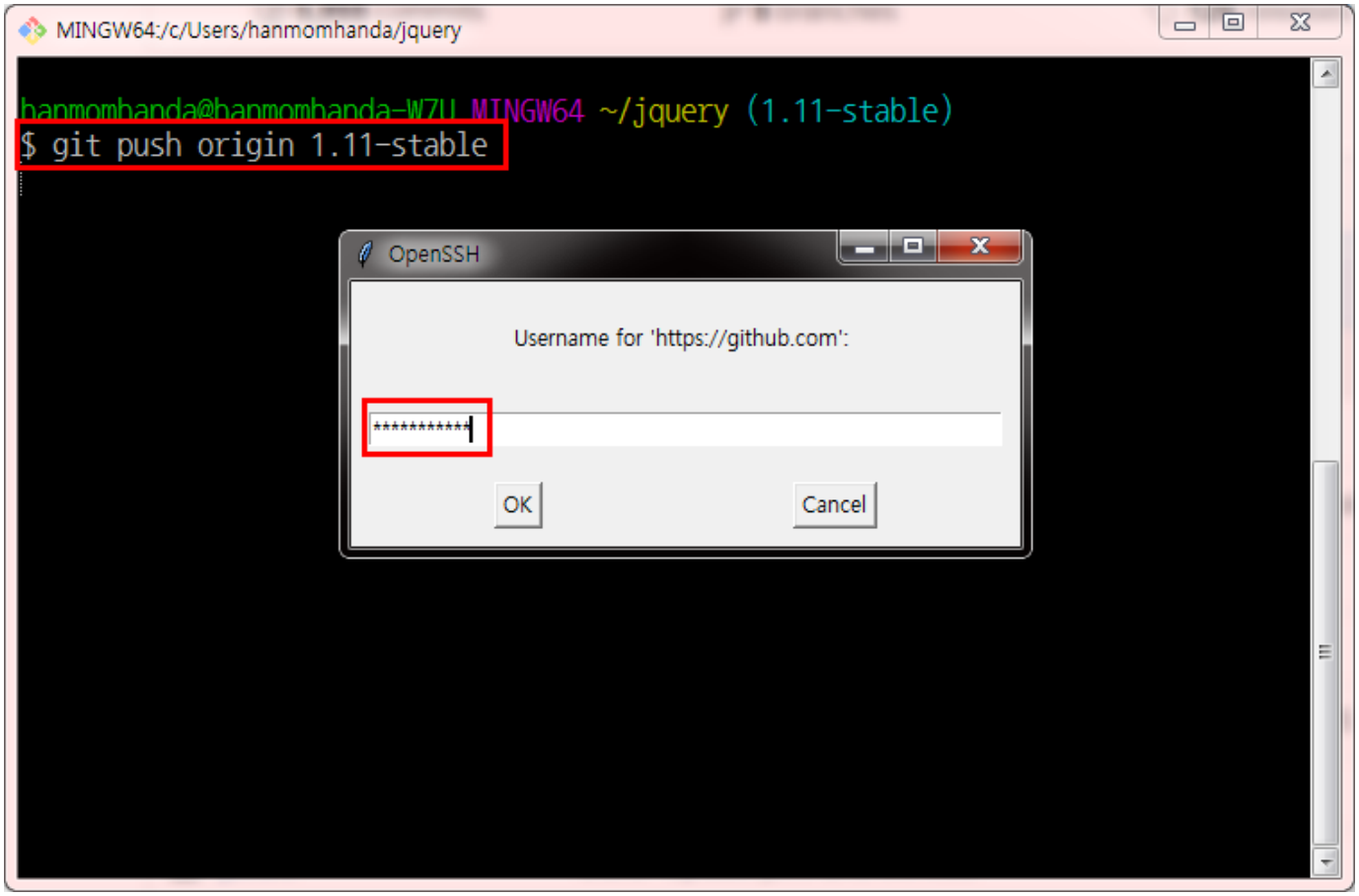


3. 주요 기능

3.13 git push (1/2)

- 로컬 저장소에 있는 브랜치를 원격 저장소에 반영한다.

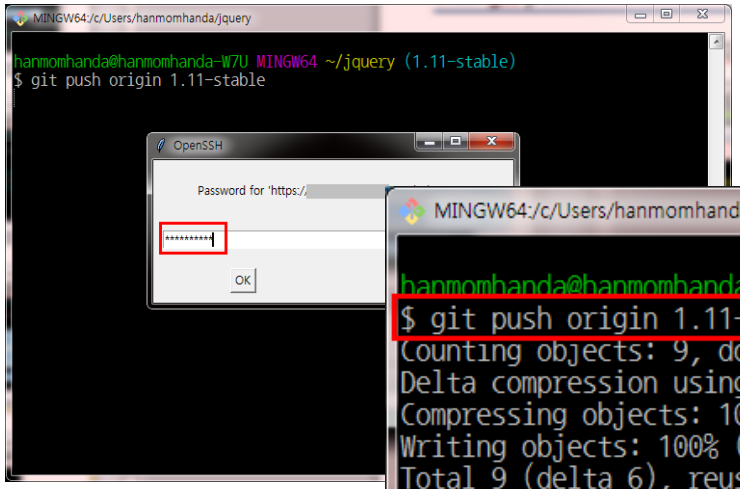
```
git push 원격저장소이름 로컬저장소의_브랜치이름
```



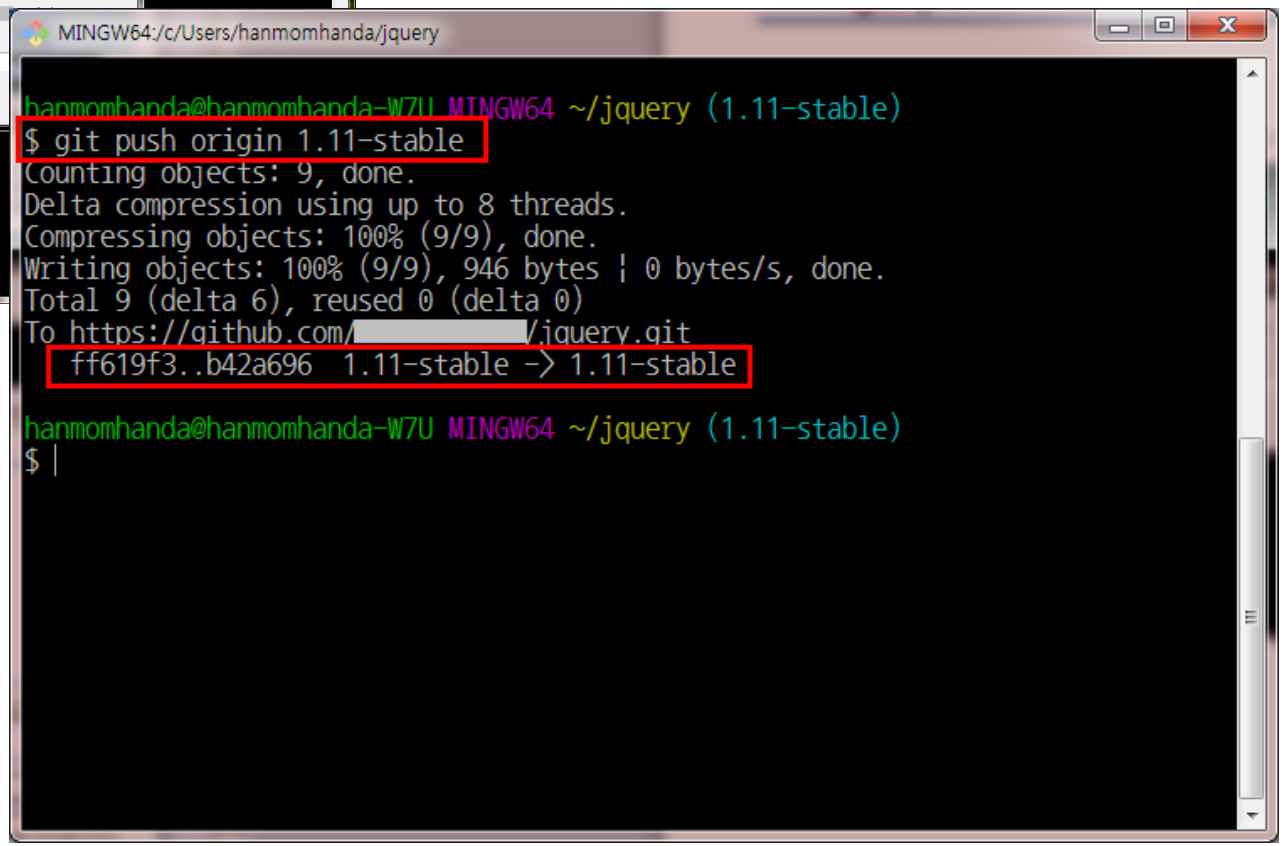
3. 주요 기능

3.13 git push (2/2)

- 로컬 저장소 접근 인증



- push 완료



4. 활용 예제

세부 목차

4. 활용 예제

4.1 예제 소개

- 본 예제는 프로젝트 내부에서 사용할 수 있는 git 서버 저장소를 구축하고,
- git 서버 저장소의 내용을 로컬에 clone 해서 개발 환경을 구축하는 것을 목표로 한다.
- 개발 환경 구축 후 실제의 버전 관리 활동은 3장을 참고한다.

4. 활용 예제

4.2 git 서버 저장소 구축 (1/5)

- 본 예제에서는 Ubuntu 리눅스에 git 서버 저장소를 구축한다.
- 설치 전 권한문제를 없애기 위해 root 권한으로 설치 시작
- git 설치

```
root@test-server:/# apt-get install git
```

- git 서버 저장소를 관리할 계정 생성
  - 계정 : git, 비밀번호 : 1111, 그룹 : git 생성

```
root@test-server:/# adduser --system --shell /bin/bash --gecos 'git version control' --group
--home /home/git git
시스템 사용자 `git' (116) 추가 ...
새 그룹 `git' (126) 추가 ...
새로운 사용자 `git' (116) 을(를) 그룹 `git'(으)로 추가 ...
'/home/git' 홈 디렉터리를 생성하는 중...
root@test-server:/# passwd git
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
root@test-server:/#
```

4. 활용 예제

4.2 git 서버 저장소 구축 (2/5)

- git 서버 저장소를 위한 repositories 폴더 생성

```
root@test-server:/# mkdir /home/git/repositories
```

- 서버에 저장할 프로젝트 생성

```
root@test-server:/# cd /home/git
root@test-server:/home/git# mkdir Project
```

- 파일 추가 및 로컬 저장소 생성

```
root@test-server:/home/git# cd Project/
root@test-server:/home/git/Project# touch README
root@test-server:/home/git/Project# git init; git add .
Initialized empty Git repository in /home/git/Project/.git/
```

4. 활용 예제

4.2 git 서버 저장소 구축 (3/5)

- 현재 상태 확인 후 최초 커밋

```
root@test-server:/home/git/Project# git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   README
#
root@test-server:/home/git/Project# git commit -a -m "Project First Commit"
[master (root-commit) 27d7f53] Project First Commit
 0 files changed
 create mode 100644 README
root@test-server:/home/git/Project#
```

4. 활용 예제

4.2 git 서버 저장소 구축 (4/5)

- 서버 저장소를 만들기 위한 설정

```
root@test-server:/home/git/Project# cd ..
root@test-server:/home/git# git clone --bare Project Project.git
Cloning into bare repository 'Project.git'...
done.
root@test-server:/home/git# touch Project.git/git-daemon-export-ok
```

- Project.git 을 git 계정의 repositories로 이동

```
root@test-server:/home/git# cp -R Project.git /home/git/repositories
root@test-server:/home/git# cd /home/git/repositories/Project.git
root@test-server:/home/git/repositories/Project.git# git --bare update-server-info
root@test-server:/home/git/repositories/Project.git# mv hooks/post-update.sample hooks/post-u
pdate
```

4. 활용 예제

4.2 git 서버 저장소 구축 (5/5)

- 클라이언트에서 ssh로 접근할 수 있도록 ssh-server 설치

```
root@test-server:/home/git/repositories/Project.git# apt-get install ssh-server
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
ssh-server 패키지는 다음이 제공하는 가상 패키지입니다:
  dropbear 2011.54-1ubuntu0.12.04.2
  lsh-server 2.0.4-dfsg-9
  openssh-server 1:5.9p1-5ubuntu1
설치할 패키지를 하나 분명히 지정해야 합니다.

E: 'ssh-server' 패키지는 설치할 수 있는 후보가 없습니다
root@test-server:/home/git/repositories/Project.git# apt-get install ssh
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지를 더 설치할 것입니다:
  openssh-server ssh-import-id
제안하는 패키지:
  rssh molly-guard openssh-blacklist openssh-blacklist-extra monkeysphere
다음 새 패키지를 설치할 것입니다:
  openssh-server ssh ssh-import-id
0개 업그레이드, 3개 새로 설치, 0개 제거 및 192개 업그레이드 안 함.
350 k바이트 아카이브를 받아야 합니다.
이 작업 후 921 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? Y
```

4. 활용 예제

4.3 git 로컬 저장소 구축

- 서버 저장소를 clone 해서 로컬 저장소를 구축한다.

- 3장의 내용을 참고하여 로컬 저장소에서 버전 관리 활동 수행