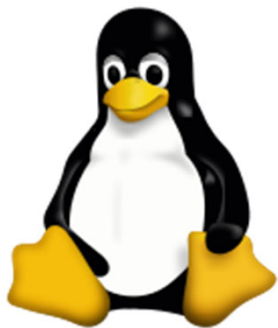


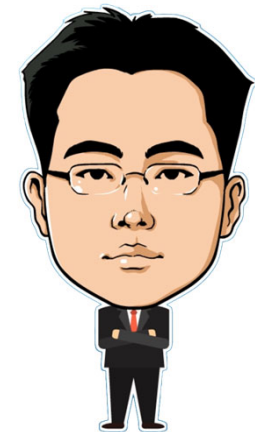
GNU/Linux

Git 호스팅
(Git Hosting)

Seo, Doo-Ok



Clickseo.com
clickseo@gmail.com



목 차



- **원격 저장소**
- **원격 저장소와 Git**



원격 저장소



- 원격 저장소



- GitHub



- 원격 저장소와 Git



- **GitHub** : **github.com**

- 2008년, **Git** 을 사용하는 프로젝트를 지원하는 웹 호스팅 서비스

- Git 의 기본 기능과 버그 추적(bug tracking), 기능 요청(feature requests), 작업 관리(task management), 위키(wiki) 기능 등을 추가적으로 제공한다.

- 영리적인 서비스와 오픈소스를 위한 무상 서비스를 모두 제공
 - 사용자 2,800만명과 8,500만개의 저장소

- 2018년 6월, MS 에서 GitHub 를 75억달러(8조원)에 인수 발표

- 프로그래밍 언어 : Ruby
 - 루비 온 레일즈(Ruby on Rails)로 작성



- 루비 온 레일즈(Ruby on Rails) : **rubyonrails.org**
 - Ruby로 작성된 MVC 패턴을 이용하는 오픈소스 웹 프레임워크
 - 라이선스 : MIT License
 - 저장소 : <https://github.com/rails/rails>



GitLab

- **GitLab** : gitlab.com

- 2011년 설립, 웹 기반의 Git 저장소 관리 서비스

- 위키와 이슈 추적 기능을 갖춘 웹 기반의 **Git** 저장소 관리자
- 영리적인 유료 서비스와 오픈소스를 위한 무료 서비스 모두 제공
 - 설치형(무료)과 호스팅용 서비스 제공
 - 전문 개발자에게 비공개 프로젝트를 호스팅하는 대가로 비용 청구 : 월 2달러
 - **협업자수에 제한없이 비공개 무료 저장소 제공**
- 2018년, Apple 은 자사 개발 툴인 Xcode 10 부터 GitLab 지원한다고 발표

- 라이선스 : 이중 라이선스

- Community Edition : MIT License
- Enterprise Edition : **GitLab EE(Enterprise Edition) License**

- 프로그래밍 언어 : Ruby, Go, Vue.js



GitLab Enterprise Edition : gitlab.com

<https://gitlab.com/gitlab-org/gitlab>

(read-only) GitLab Community Edition

<https://gitlab.com/gitlab-org/gitlab-foss>

Bitbucket

- **Bitbucket** : bitbucket.org

- 2010년 9월, 아틀라시안(Atlassian)에서 매입

- 영리적인 유료 서비스와 오픈소스를 위한 무료 서비스 모두 제공
 - 사용자 600만명
 - 전문 개발자에게 비공개 프로젝트를 호스팅하는 대가로 비용 청구 : 월 2달러(또는 5달러)
 - 최대 5명의 협업자로 구성된 소규모 팀에게 비공개 무료 저장소 허용

- 배포 모델

- Bitbucket Cloud(이전의 Bitbucket) : Django 웹 프레임워크를 사용(Python)
- Bitbucket Server
 - 라이선스 : 사유 소프트웨어
 - 프로그래밍 언어 : Java
- Bitbucket Data Center





원격 저장소

GitHub



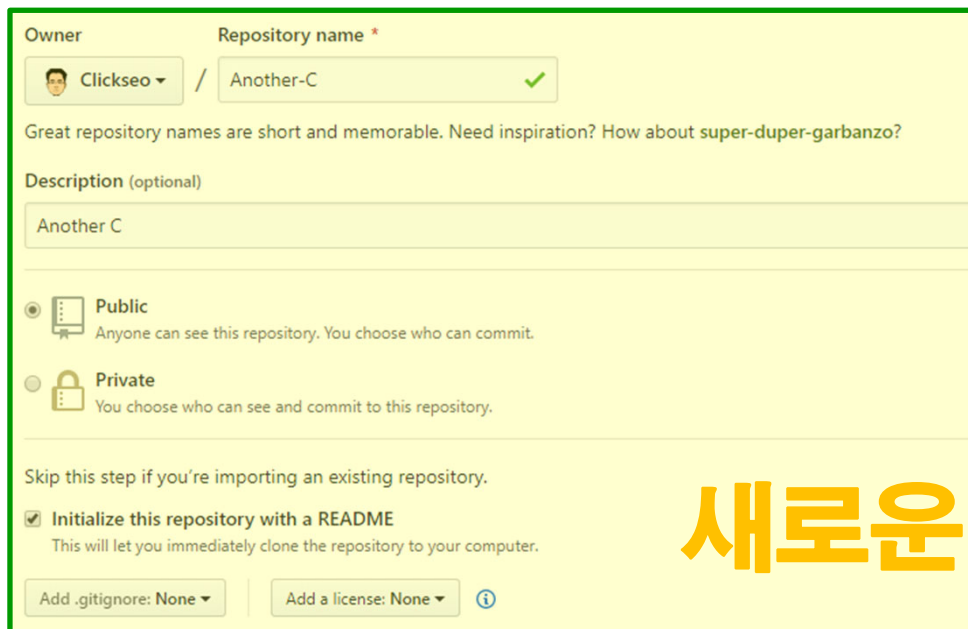
GitHub (1/8)

- **원격 저장소** : 새로운 저장소 생성

- **New repository** : Repositories >> New

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)



Owner: Clickseo / Repository name: Another-C ✓

Great repository names are short and memorable. Need inspiration? How about [super-duper-garbanzo?](#)

Description (optional): Another C

☒ Public: Anyone can see this repository. You choose who can commit.

☐ Private: You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README: This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Create repository

새로운 저장소 생성

GitHub (2/8)

● 원격 저장소 : 새로운 저장소 생성

○ Create a new repository

- **Owner** : 사용자 아이디
 - 협업 환경에서는 다른 사용자의 아이디를 지정할 수도 있다.
- **Repository name** : 새롭게 생성할 원격 저장소의 이름 지정
 - 가능하면 로컬 환경에서 작업할 Git 프로젝트 디렉터리 이름과 같게 하는게 좋다.
- **Description** : (선택) 생성한 원격 저장소가 어떤 역할을 하는지 작성
- **Public / Private** : 원격 저장소의 공개 여부를 선택
- **Initialize this repository with a README**
 - 지정한 Repository name 과 Description 항목의 내용을 담은 [README.md](#) 파일을 생성한다.
- **Add .gitignore**
 - 원격 저장소에 포함하지 않을 파일들의 목록을 만들 때 사용한다.
- **Add a license**
 - 원격 저장소에 저장할 프로젝트가 어떤 라이선스에 속할지를 선택한다.

GitHub (3/8)

- **원격 저장소 : GitHub**

- 외부에서 접속해서 사용하는 저장소

- 로컬에서 작업한 Git 프로젝트 저장소가 외부에 있는 것이다.

The screenshot shows the GitHub interface for a repository named 'Another-C' by user 'Clickseo'. At the top, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security' (0), 'Insights', and 'Settings'. The repository name 'Another C' is displayed with an 'Edit' button. Below the name, it shows '1 commit', '1 branch', '0 packages', '0 releases', and '1 contributor'. There are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table shows one commit: 'Clickseo Initial commit' with the latest commit hash '5b14194' from '12 minutes ago'. Below the table, the 'README.md' file is shown with its content: 'Another-C' and 'Another C'.

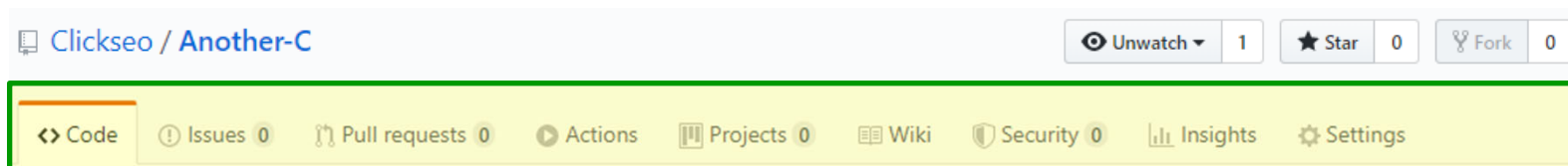
GitHub, 원격 저장소

GitHub (4/8)

● 원격 저장소 : 구조

○ 원격 저장소의 세부 구조

- **Code** : 원격 저장소의 루트 디렉터리
- **Issues** : 원격 저장소의 주요 이슈 사항을 기재한 후 관리한다.
- **Pull Request** : 전체 목록
 - (상호 작용) Fork 한 저장소를 수정해 다시 원본 저장소에 병합해달라는 요청
- **Actions**
- **Projects**
- **Wiki** : 공유할 정보나 개발 문서, 참고 자료 등을 작성
- **Security**
- **Insights**
- **Settings**

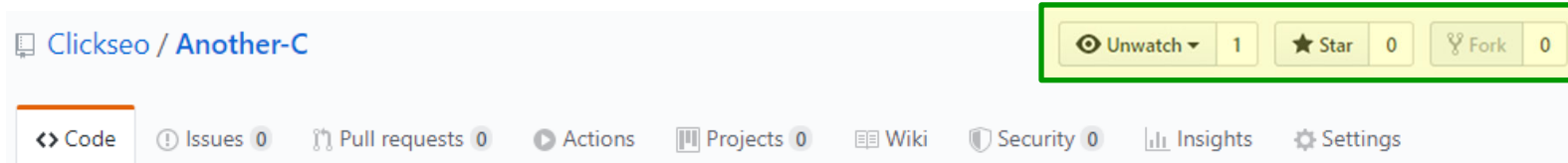


GitHub (5/8)

● 원격 저장소 : 구조

○ 원격 저장소의 세부 구조

- **Watch** : 원격 저장소의 활동 내역
 - **Not Watching** : 댓글이나 이슈 등에서 언급될 때만 알려준다.
 - **Watching** : 모든 활동 내역을 알려준다.
 - **Ignoring** : 모든 알림을 무시한다.
- **Star** : 원격 저장소에 관심이 있을 때...
- **Fork** : 원격 저장소를 분기



GitHub (6/8)

● GitHub : 구조

○ Fork

- 다른 사람의 원격 저장소를 내 계정으로 복사
- Fork 하지 않은 저장소
 - 쓰기 권한이 없는 원격 저장소(아무나 사용할 수 없는 원격 저장소)
 - 자유롭게 파일을 생성하거나 수정하여 원격 저장소에 반영하는게 불가능

The screenshot displays the GitHub interface for the 'torvalds / linux' repository. At the top, the repository name is highlighted in a green box. Below it, the 'Fork' button is also highlighted in a green box. A red arrow points from this 'Fork' button to a forked repository, 'Clickseo / linux', which is also highlighted in a green box. The text '원격 저장소 복제' (Remote Repository Duplication) is written in large yellow characters on the right side of the image. The interface shows various repository statistics like commits, branches, packages, releases, and contributors. The bottom section shows the 'Linux kernel source tree' and a 'Clone or download' button.

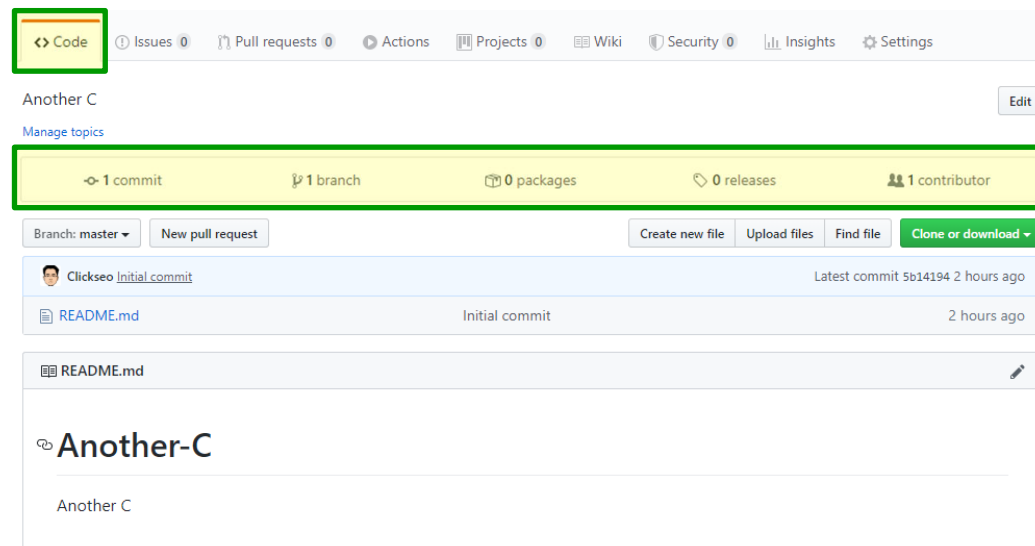
GitHub (7/8)

- **원격 저장소 : 정보**

- **원격 저장소의 세부 정보 : Code**

- **Description** : 원격 저장소에 대한 설명
- **Commits, Branches** : 해당 원격 저장소의 commit / branch 수
- **Packages, Releases**
- **Contributors** : 원격 저장소에 commit 혹은 pull request가 받아들여진 사용자 수
- **View license**

Code



GitHub (8/8)

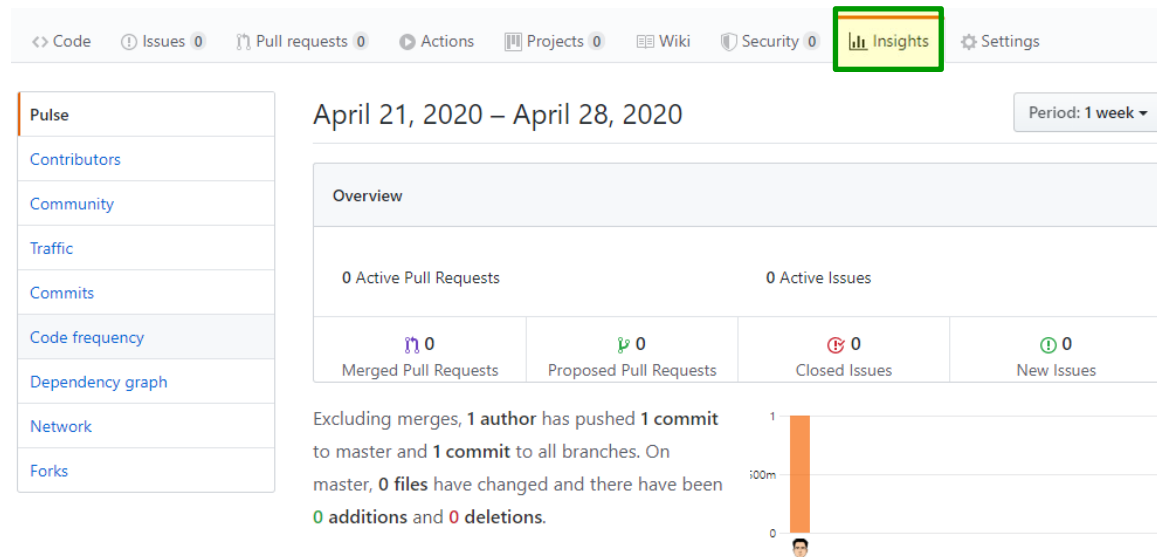
- **원격 저장소 : 정보**

- 원격 저장소의 세부 정보 : **Insights**

- **Pulse** : 원격 저장소에 대한 기본적인 정보

Insights

- Contributors
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Networks, Forks



원격 저장소와 Git



- 원격 저장소



- 원격 저장소와 Git



- Git/GitHub



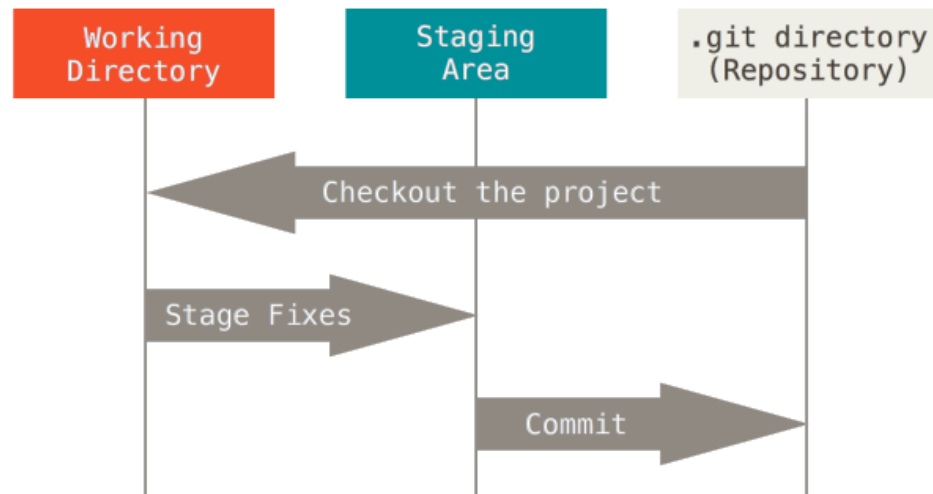
원격 저장소와 Git

● Git : 파일과 프로젝트

○ Git 은 세 가지 상태로 파일 관리

- **Modified** : 수정한 파일을 아직 로컬 데이터 베이스에 Commit 하지 않은 것을 의미
- **Staged** : 현재 수정 파일을 곧 Commit 할 것 이라고 표시한 상태
- **Committed** : 데이터가 로컬 데이터베이스에 안전 하게 저장됐다는 것을 의미

○ Git 프로젝트의 세 가지 단계





원격 저장소와 Git

Git/GitHub



원격저장소와 Git (1/8)

● Git 명령어

○ 원격 저장소와 로컬 저장소 명령어

명령어	설 명
git clone	원격 저장소의 모든 내용을 로컬 저장소 로 복사한다.
git remote	로컬 저장소를 특정 원격 저장소 와 연결 한다.
git push	로컬 저장소의 내용을 보내거나, 로컬 저장소의 변경 사항을 원격 저장소 로 보낸다.
git fetch	원격 저장소의 commit 들을 로컬 저장소 로 가져온다. 로컬 저장소에서 commit 정보들을 확인하고 수동으로 병합해야 한다.
git pull	원격 저장소의 정보를 로컬 저장소로 가져오면서, 자동으로 로컬 저장소의 브랜치에 병합 까지 수행한다. (git push 와 반대 성격의 명령어)

원격저장소와 Git (2/8)

● 원격 저장소 데이터 복사 : Clone

○ Clone

- 원격 저장소와 내 컴퓨터(로컬 저장소)를 연결해 데이터를 복사하는 작업
 - 내가 생성한 원격 저장소를 내 컴퓨터와 연결해서 데이터를 복사하는 작업
 - Fork 한 원격 저장소를 내 컴퓨터와 연결해서 데이터를 복사하는 작업
- 네트워크 프로토콜
 - Clone with HTTPS
 - Clone with SSH

○ Fork

- 원격 저장소 안에서 원격 저장소를 복사하는 작업

원격저장소와 Git (3/8)

● 원격 저장소 데이터 복사 : GitHub

○ 원격 저장소 : GitHub

- Clone or Download

1. 네트워크 프로토콜
 - Clone with HTTPS
 - Clone with SSH
2. Open in Desktop
3. Download ZIP

The screenshot shows the GitHub repository page for 'Clickseo / Another-C'. The repository has 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. The 'Clone or download' dropdown menu is open, showing options to 'Clone with HTTPS', 'Use SSH', 'Open in Desktop', and 'Download ZIP'. The URL 'https://github.com/Clickseo/Another-C.git' is displayed. The repository name 'Another-C' is also visible at the bottom of the page.

Clone or Download

원격저장소와 Git (4/8)

- 원격 저장소 데이터 복사 : `git clone`

- `git clone` : 원격 저장소의 모든 내용을 로컬 저장소로 복사

```
$ git clone cloneURL
```

- 네트워크 프로토콜 : Clone with **HTTPS**, Clone with **SSH**

\$ `git clone https://`

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~$ git clone https://github.com/Clickseo/Another-C.git
'Another-C'에 복제합니다...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
오브젝트 묶음 푸는 중: 100% (3/3), 602 bytes | 602.00 KiB/s, 완료.
clickseo@clickseo-VirtualBox:~$ cd Another-C/
clickseo@clickseo-VirtualBox:~/Another-C$ git status
현재 브랜치 master
브랜치가 'origin/master'에 맞게 업데이트된 상태입니다.

커밋할 사항 없음, 작업 폴더 깨끗함
clickseo@clickseo-VirtualBox:~/Another-C$ ls
README.md
clickseo@clickseo-VirtualBox:~/Another-C$
```

원격저장소와 Git (5/8)

- 원격 저장소 연결 : git remote

- git remote : 로컬 저장소와 원격 저장소를 연결

```
$ git remote add original cloneURL
```

```
$ git remote -v
```

- original : 원격 저장소 별칭
- cloneURL : 원격 저장소 URL

```
$ git remote add original https://
```

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git remote add original https://github.com/Clickseo/Another-C.git
clickseo@clickseo-VirtualBox:~/Another-C$ git remote -v
origin https://github.com/Clickseo/Another-C.git (fetch)
origin https://github.com/Clickseo/Another-C.git (push)
original https://github.com/Clickseo/Another-C.git (fetch)
original https://github.com/Clickseo/Another-C.git (push)
clickseo@clickseo-VirtualBox:~/Another-C$
```

원격저장소와 Git (6/8)

- 원격 저장소 업로드 : git push

- git push : 로컬 저장소의 작업 내용을 원격 저장소에 업로드

```
$ git push original --all
```

- original : 원격 저장소 별칭
- --all 옵션
 - » original 원격 저장소에 로컬의 모든 Branch를 push 한다.

로컬 저장소
새로운 파일 추가 : hello.c

```
$ git push original -all
```

```
clickseo@clickseo-VirtualBox:~/Another-C$ ls
README.md hello.c
clickseo@clickseo-VirtualBox:~/Another-C$ git push original --all
Username for 'https://github.com': clickseo@gmail.com
Password for 'https://clickseo@gmail.com@github.com':
오브젝트 나열하는 중: 4, 완료.
오브젝트 개수 세는 중: 100% (4/4), 완료.
오브젝트 압축하는 중: 100% (3/3), 완료.
오브젝트 쓰는 중: 100% (3/3), 359 bytes | 359.00 KiB/s, 완료.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Clickseo/Another-C.git
9d237d6..5aabf83 master -> master
clickseo@clickseo-VirtualBox:~/Another-C$
```


원격저장소와 Git (7/8)

- 원격 저장소 업로드 : git push

- git push : 로컬 저장소의 작업 내용을 원격 저장소에 업로드

The screenshot shows the GitHub interface for a repository named 'Another-C' by user 'Clickseo'. The repository has 1 branch (master) and 2 commits. The commit history table is as follows:

Commit Message	Initial Commit	Latest Commit
Initial commit	Initial commit	Initial commit
New file "hello.c"	New file "hello.c"	Latest commit 5aafb83 5 minutes ago

The file list shows 'README.md' and 'hello.c'. The 'hello.c' file is highlighted with a green border. Below the file list, the README content is visible, showing the repository name 'Another-C'.

git push 완료 후 GitHub

원격저장소와 Git (8/8)

- 원격 저장소 업로드 : `git push`

- `git push` : 로컬 저장소의 작업 내용을 원격 저장소에 업로드

- 특정 브랜치 반영

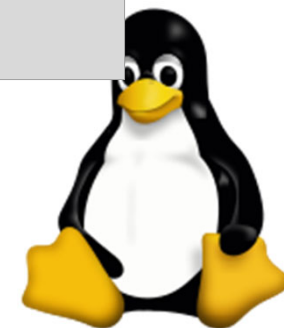
```
$ git push original master
```

```
$ git push original hotfix
```



원격 저장소와 Git

충돌(Conflict)



원격저장소와 Git : 총돌 (1/9)

● Git 명령어

○ 원격 저장소와 로컬 저장소 명령어

명령어	설 명
git clone	원격 저장소의 모든 내용을 로컬 저장소로 복사한다.
git remote	로컬 저장소를 특정 원격 저장소와 연결한다.
git push	로컬 저장소의 내용을 보내거나, 로컬 저장소의 변경 사항을 원격 저장소로 보낸다.
git fetch	원격 저장소의 commit 들을 로컬 저장소로 가져온다. 로컬 저장소에서 commit 정보들을 확인하고 수동으로 병합해야 한다.
git pull	원격 저장소의 정보를 로컬 저장소로 가져오면서, 자동으로 로컬 저장소의 브랜치에 병합까지 수행한다. (git push와 반대 성격의 명령어)

원격저장소와 Git : 충돌 (2/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : git fetch

- git fetch : 원격 저장소의 commit 들을 로컬 저장소로 가져온다.

새로운 파일 생성 : clickseo.c
기존 파일 수정 : hello.c

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ cat hello.c
#include <stdio.h>

int main(void)
{
    printf("Hello World!!! \n");
    printf("local modify Hello World!!! \n");
    return 0;
}

clickseo@clickseo-VirtualBox:~/Another-C$ git push original --all
Username for 'https://github.com': clickseo@gmail.com
Password for 'https://github.com:clickseo@gmail.com@github.com':
To https://github.com/Clickseo/Another-C.git
 ! [rejected]        master -> master (non-fast-forward)
error: 레퍼런스를 'https://github.com/Clickseo/Another-C.git'에 푸시하는데 실패했습니다
힌트: 현재 브랜치의 끝이 리모트 브랜치보다 뒤에 있으므로 업데이트가
힌트: 거부되었습니다. 푸시하기 전에 ('git pull ...' 등 명령어로) 리모트
힌트: 변경 사항을 포함하십시오.
힌트: 자세한 정보는 'git push --help'의 "Note about fast-forwards" 부분을
힌트: 참고하십시오.
clickseo@clickseo-VirtualBox:~/Another-C$
```

\$ git push original -all

원격저장소와 Git : 충돌 (3/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : `git fetch`

- `git fetch` : 원격 저장소의 commit 들을 로컬 저장소로 가져온다.

```
$ git fetch
```

```
$ git fetch
```

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git fetch
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
오브젝트 묶음 푸는 중: 100% (6/6), 1.44 KiB | 70.00 KiB/s, 완료.
https://github.com/Clickseo/Another-C URL에서
9d237d6..8ff90f2 master -> origin/master
clickseo@clickseo-VirtualBox:~/Another-C$
```

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git status
현재 브랜치 master
현재 브랜치와 'origin/master'이(가) 갈라졌습니다,
다른 커밋이 각각 1개와 2개 있습니다.
(리모트의 브랜치를 현재 브랜치로 병합하려면 "git pull"을 사용하십시오)

커밋할 사항 없음, 작업 폴더 깨끗함
clickseo@clickseo-VirtualBox:~/Another-C$
```

원격저장소와 Git : 충돌 (4/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : git fetch

- git fetch : 원격 저장소의 commit 들을 로컬 저장소로 가져온다.

충돌(Conflict) 발생

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git branch
* master
clickseo@clickseo-VirtualBox:~/Another-C$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/original/master
clickseo@clickseo-VirtualBox:~/Another-C$ git merge origin/master
자동 병합: hello.c
충돌 (내용): hello.c에 병합 충돌
자동 병합이 실패했습니다. 충돌을 바로잡고 결과물을 커밋하십시오.
clickseo@clickseo-VirtualBox:~/Another-C$
```

\$ git merge origin/master

원격저장소와 Git : 충돌 (5/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : git fetch

- git fetch : 원격 저장소의 commit 들을 로컬 저장소로 가져온다.

\$ git diff

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git diff
diff --cc hello.c
index 89b9f6b,fb39977..0000000
--- a/hello.c
+++ b/hello.c
@@@ -3,7 -3,7 +3,11 @@@
int main(void)
{
    printf("Hello World!!! \n");
++<<<<<<< HEAD
+    printf("local modify Hello World!!! \n");
++=====
+    printf("Hi~ Clickseo ^..^ \n");
++>>>>>>> origin/master

    return 0;
}
clickseo@clickseo-VirtualBox:~/Another-C$
```

변경 사항 확인

원격저장소와 Git : 충돌 (6/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : git fetch

- git fetch : 원격 저장소의 commit 들을 로컬 저장소로 가져온다.

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ cat hello.c
#include <stdio.h>

int main(void)
{
    printf("Hello World!!! \n");
<<<<<< HEAD
    printf("local modify Hello World!!! \n");
=====
    printf("Hi~ Clickseo ^..^ \n");
>>>>>> origin/master

    return 0;
}
clickseo@clickseo-VirtualBox:~/Another-C$ vi hello.c
clickseo@clickseo-VirtualBox:~/Another-C$ cat hello.c
#include <stdio.h>

int main(void)
{
    printf("Hello World!!! \n");
    printf("local modify Hello World!!! \n");

    return 0;
}
clickseo@clickseo-VirtualBox:~/Another-C$
```

충돌 파일 수정

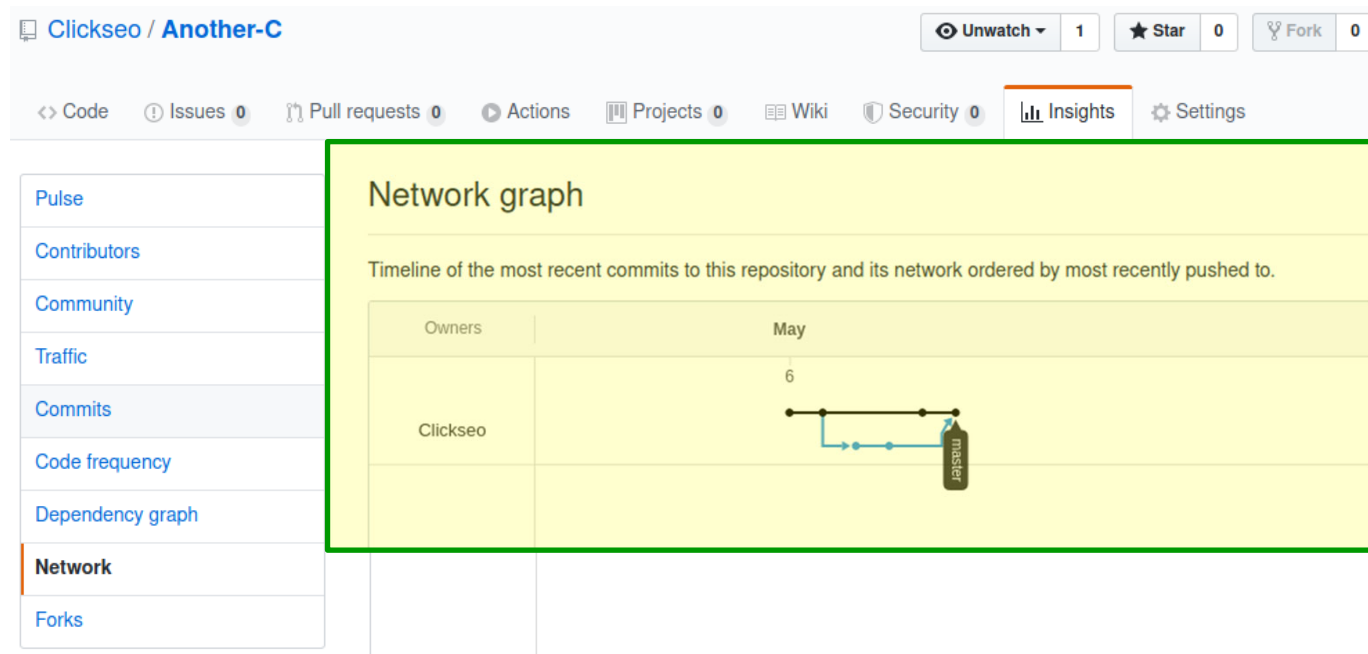
```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git commit -a -m "conflict resolve GitHub"
[master 3a7251c] conflict resolve GitHub
clickseo@clickseo-VirtualBox:~/Another-C$ git status
현재 브랜치 master
브랜치가 'origin/master'보다 2개 커밋만큼 앞에 있습니다.
(로컬에 있는 커밋을 제출하려면 "git push"를 사용하십시오)

커밋할 사항 없음, 작업 폴더 깨끗함
clickseo@clickseo-VirtualBox:~/Another-C$ git push original --all
Username for 'https://github.com': clickseo@gmail.com
Password for 'https://clickseo@gmail.com@github.com':
오브젝트 나열하는 중: 9, 완료.
오브젝트 개수 세는 중: 100% (9/9), 완료.
오브젝트 압축하는 중: 100% (5/5), 완료.
오브젝트 쓰는 중: 100% (5/5), 615 bytes | 205.00 KiB/s, 완료.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Clickseo/Another-C.git
8ff90f2..3a7251c master -> master
clickseo@clickseo-VirtualBox:~/Another-C$
```

```
$ git push original -all
```

원격저장소와 Git : 충돌 (7/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : `git fetch`
 - `git fetch` : 원격 저장소의 commit 들을 로컬 저장소로 가져온다.



GitHub Network graph

원격저장소와 Git : 충돌 (8/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : `git pull`

- `git pull` : 원격 저장소의 commit 들을 로컬 저장소로 가져오면서,
자동으로 로컬 저장소의 브랜치에 병합까지 수행한다.

```
$ git pull
```

```
$ git pull original master
```

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git status
현재 브랜치 master
브랜치가 'origin/master'보다 2개 커밋만큼 앞에 있습니다.
(로컬에 있는 커밋을 제출하려면 "git push"를 사용하십시오)

커밋할 사항 없음, 작업 폴더 깨끗함
clickseo@clickseo-VirtualBox:~/Another-C$ git pull original master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
오브젝트 묶음 푸는 중: 100% (3/3), 658 bytes | 329.00 KiB/s, 완료.
https://github.com/Clickseo/Another-C URL에서
* branch          master      -> FETCH_HEAD
  3a7251c..c3a8f72  master      -> original/master
업데이트 중 3a7251c..c3a8f72
Fast-forward
 hello.c | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
clickseo@clickseo-VirtualBox:~/Another-C$
```

원격저장소와 Git : 총돌 (9/9)

- 원격 저장소의 commit 들을 로컬 저장소로 : **git pull**

- **git pull** : 원격 저장소의 commit 들을 로컬 저장소로 가져오면서,
자동으로 로컬 저장소의 브랜치에 병합까지 수행한다.

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ cat hello.c
#include <stdio.h>

int main(void)
{
    printf("Hello World!!! \n");
    printf("Modify Hello World!!! \n");
    return 0;
}
clickseo@clickseo-VirtualBox:~/Another-C$
```

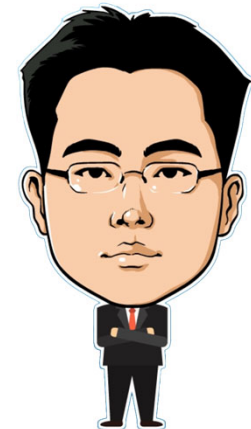
자동 병합

```
clickseo@clickseo-VirtualBox: ~/Another-C
clickseo@clickseo-VirtualBox:~/Another-C$ git commit
현재 브랜치 master
브랜치가 'origin/master'보다 3개 커밋만큼 앞에 있습니다.
(로컬에 있는 커밋을 제출하려면 "git push"를 사용하십시오)

커밋할 사항 없음, 작업 폴더 깨끗함
clickseo@clickseo-VirtualBox:~/Another-C$ git merge
이미 업데이트 상태입니다.
clickseo@clickseo-VirtualBox:~/Another-C$
```

참고문헌

- [1] Scott Chancon, Ben Straub, "Pro Git", Second Edition, Apress, 2014.
- [2] 윤웅식, "만들면서 배우는 Git + GitHub 입문", 한빛미디어, 2015.
- [3] "Git", kernel.org, 2020 of viewing the site, <https://git-scm.com/>.
- [4] "Git Wiki", kernel.org, 2020 of viewing the site, <https://git.wiki.kernel.org/>.
- [5] GitHub, 2020 of viewing the site, <https://github.com/>.



이 강의자료는 저작권법에 따라 보호받는 저작물이므로 무단 전제와 무단 복제를 금지하며,
내용의 전부 또는 일부를 이용하려면 반드시 저작권자의 서면 동의를 받아야 합니다.

Copyright © Clickseo.com. All rights reserved.

