

# Data Structure

실습1

Hyeongil Nam

# Introduction

---

- 실습 관련 문의
  - 담당 조교: 남현길
    - E-mail: skagusrlf@hanyang.ac.kr
  - 연구실: 공업센터 본관 304호
  - Office time: 금요일(13:00 ~ 17:00)
- 연구 및 관심 분야
  - Computer Vision, Computer Graphics, HCI, AR/VR

# 0. 이번 주 실습 내용

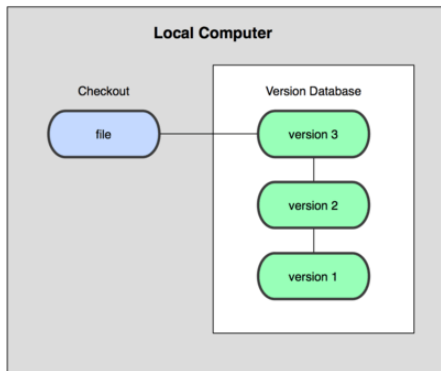
---

- **Git**
  - Why Git?
  - What is Git?
  - How to use Git?
  - **Git 실습**
    - GitLab 로그인 및 초기 설정
    - Git 설치
    - Git 기본 사용법
  - 마무리
  - 과제 (C array code 작성 & Git Push)

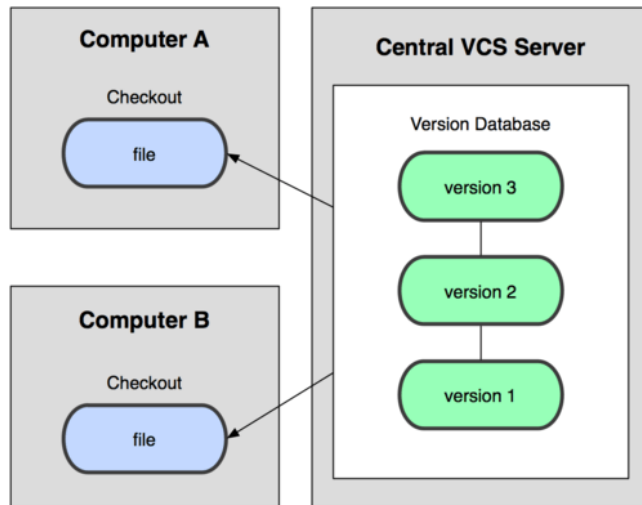
# 1. Why Git?

## - Version Control System(VCS)

151110_코드_(1).txt	2015-11-11 오후...	텍스트 문서
151110_코드_(2) - 수정.txt	2015-11-11 오후...	텍스트 문서
151110_코드_(3) - 기존코드백업.txt	2015-11-11 오후...	텍스트 문서
151110_코드_(4) - 수정2.txt	2015-11-11 오후...	텍스트 문서
151110_코드_(5) - 수정33.txt	2015-11-11 오후...	텍스트 문서
151111_코드_(6) - 완성.txt	2015-11-11 오후...	텍스트 문서
151112_코드_(7) - 완성 - 수정.txt	2015-11-11 오후...	텍스트 문서



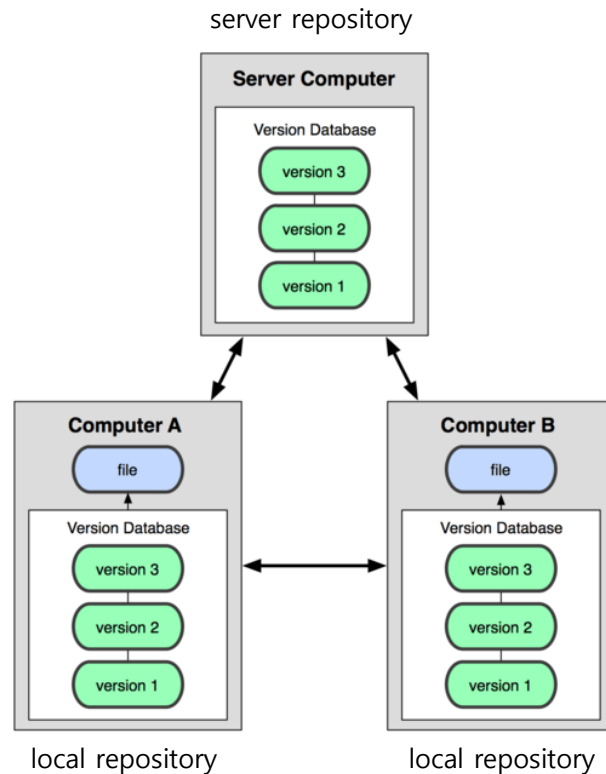
로컬 버전 관리 시스템



중앙집중식 버전 관리 시스템(CVCS)

# 1. Why Git?

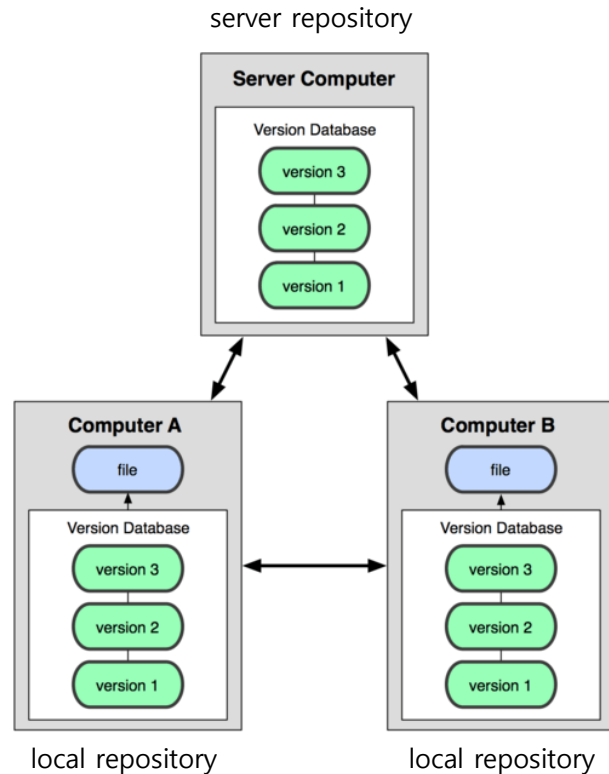
- DVCS(분산 버전 관리 시스템)
  - **Git**, Mercurial, Bazaar, Darcs 등이 존재
- 서버로부터 저장소를 전부 복제함
- 서버에 문제가 생겨도 이 복제물로 다시 작업할 수 있고 서버를 복원할 수 있음
- 여럿이서 협업하는 오픈소스 프로젝트에 활용됨



# 1. Why Git?

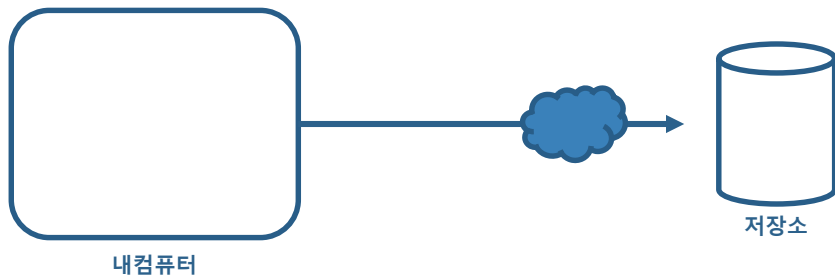
## - Git

- Linux Kernel은 대표적인 오픈소스 프로젝트
- 초창기 BitKeeper라는 DVCS를 사용했음
- BitKeeper를 사용하지 못하게 되자 리눅스 개발 커뮤니티 (특히 리눅스 창시자 리누스 토발즈)가 Git을 개발
- Git의 설계 목표
  - 빠른 속도
  - 단순한 구조
  - 동시다발적인 branch에 대한 지원
  - 완벽한 분산형 시스템
  - 리눅스와 같은 대형 프로젝트 관리도 가능

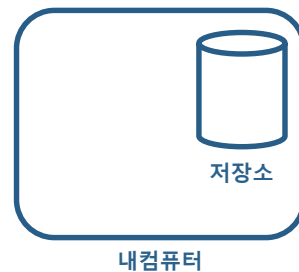


## 2. What is Git?

### - Git이란



CVS는 보통 저장소가 서버에 있음



git은 저장소가 내컴퓨터에 있음

## 2. What is Git?

- Git으로 다른 사람들과 작업을 하고 싶을 때: **원격 저장소**(remote repository)를 만든다



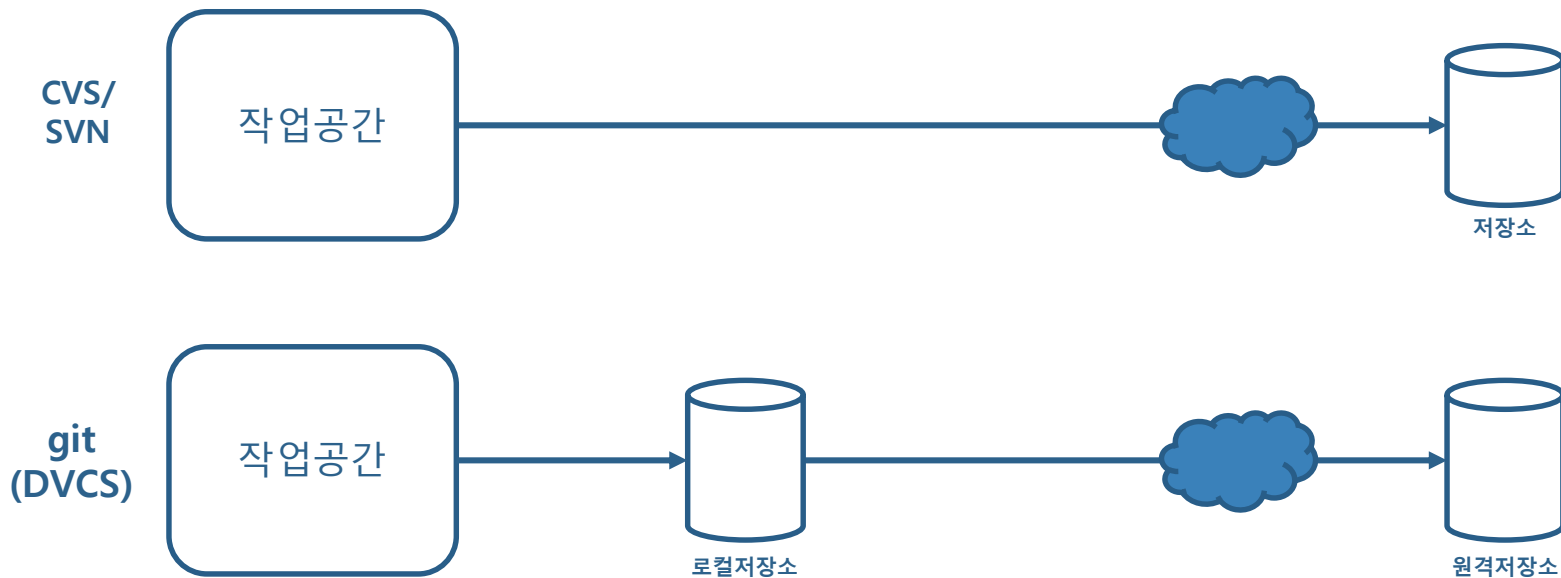
① 내 컴퓨터의 저장소에  
작업 내용을 commit

② 다른 사람과 공유할 때  
원격 저장소에 Push



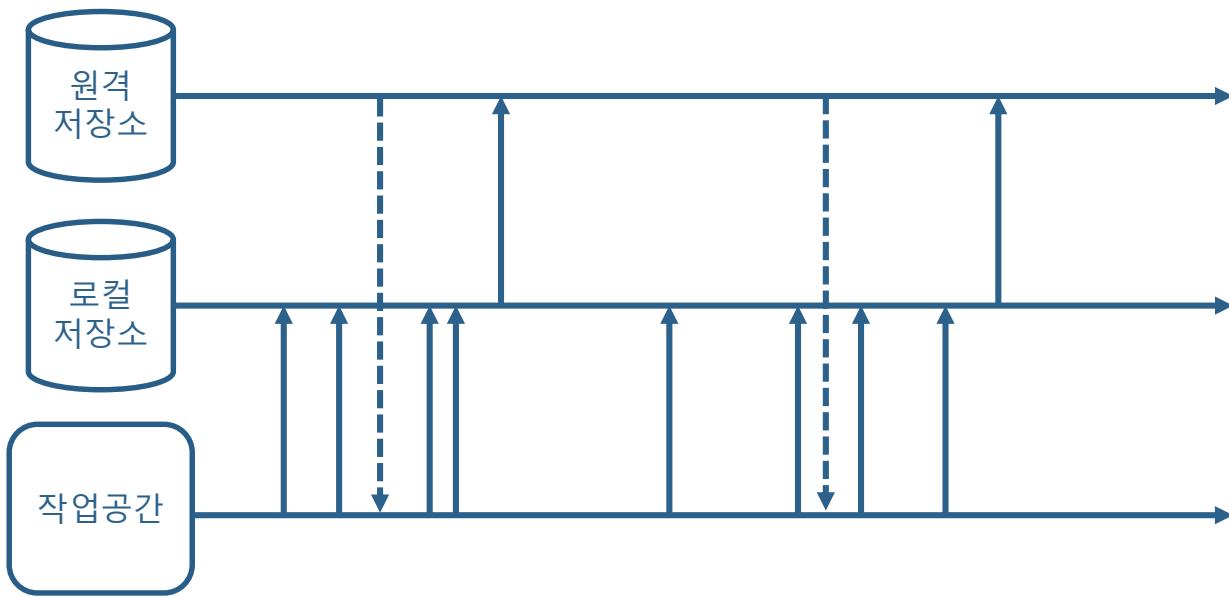
## 2. What is Git?

- Git으로 다른 사람들과 작업을 하고 싶을 때: **원격 저장소**(remote repository)를 만든다



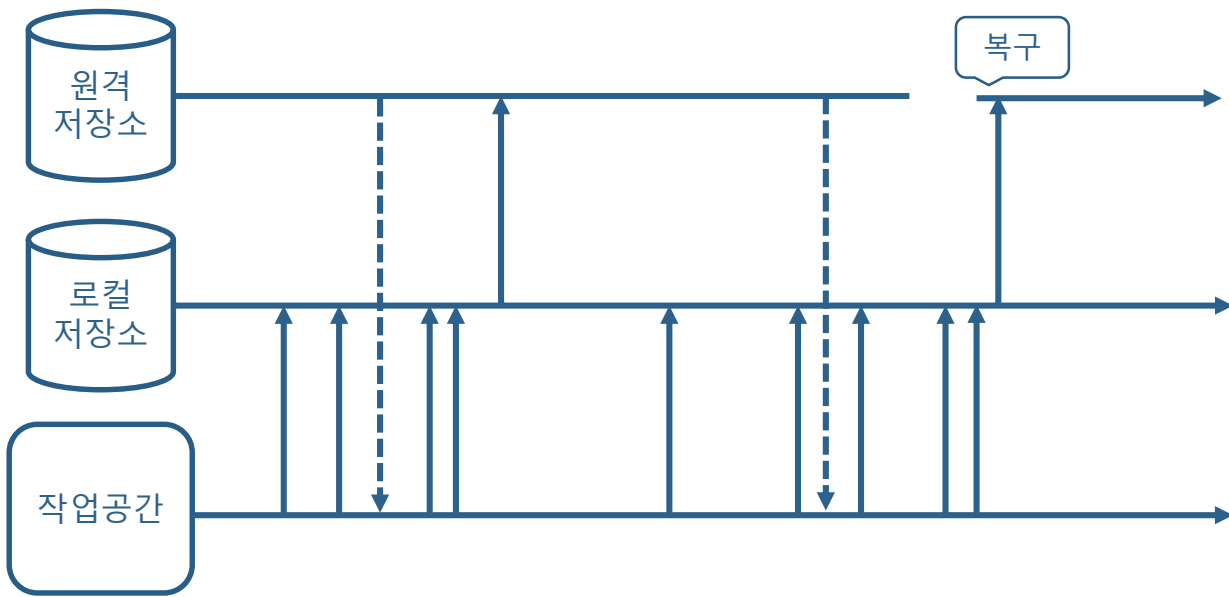
## 2. What is Git?

- commit에 부담이 없음



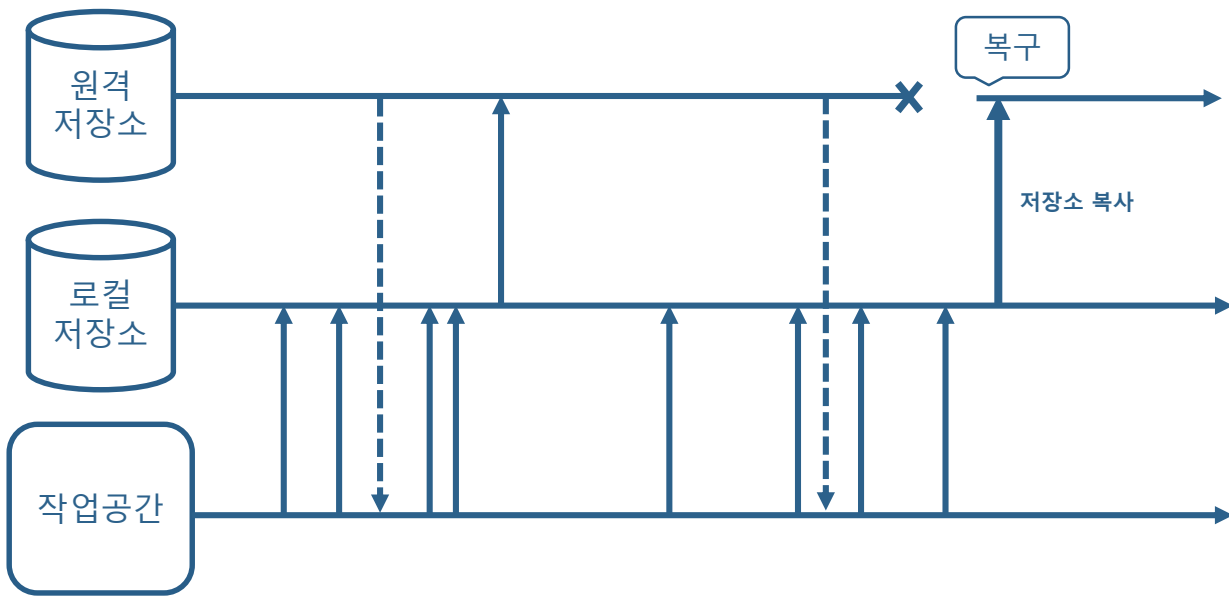
## 2. What is Git?

- 원격 저장소와 연결이 끊겨도 계속 버전관리가 가능



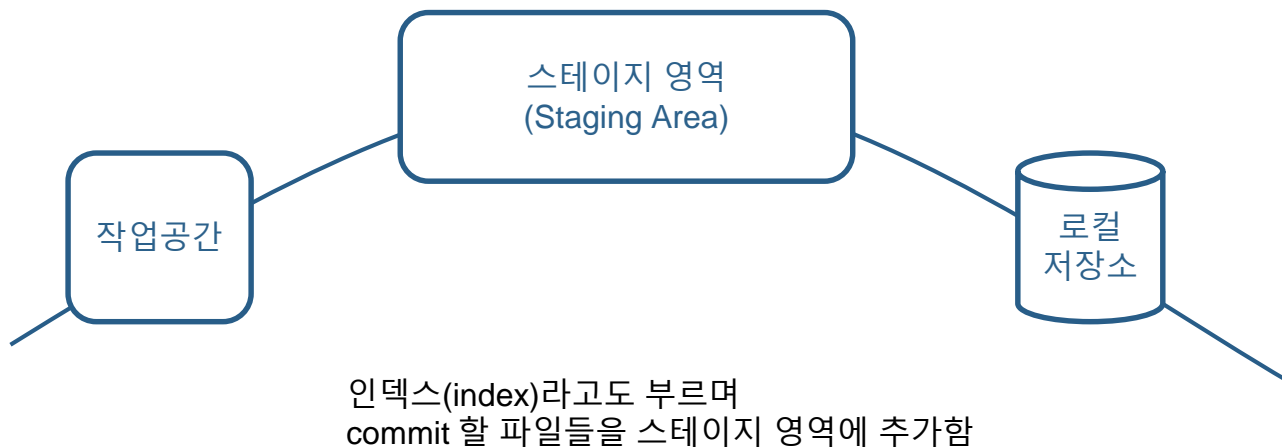
## 2. What is Git?

- 원격 저장소가 사라져도 로컬 저장소로 복원이 가능



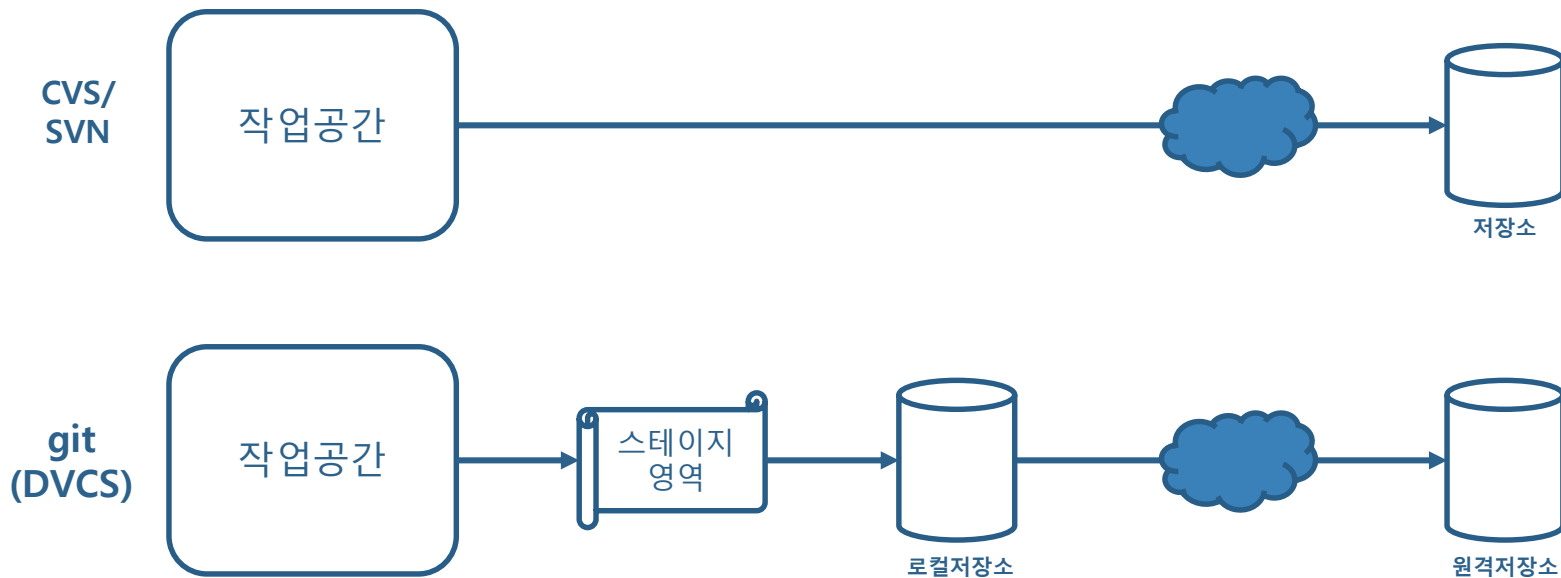
## 2. What is Git?

- Git의 특징: 로컬 저장소에 commit 전, **스테이지 영역**이라는 단계가 존재



## 2. What is Git?

- 스테이지 영역



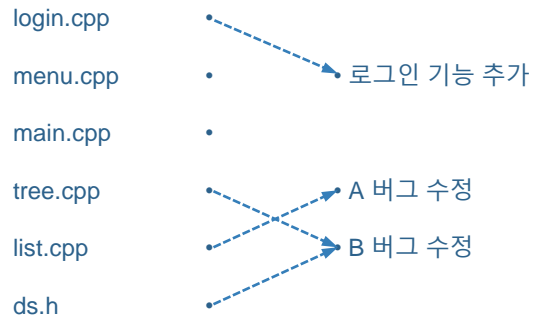
## 2. What is Git?

- 스테이지 영역의 역할

### 빌드 계획

로그인 기능 추가  
A 버그 수정  
B 버그 수정

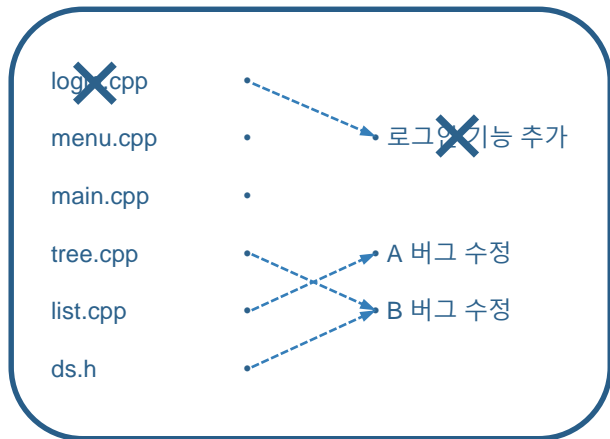
① 이런 빌드 계획을 가지고



② 다음과 같은 파일들을 수정하였는데

## 2. What is Git?

### - 스테이지 영역의 역할



③ 로그인 기능 추가가 다음 빌드로 미루어 졌다면

#### CVS의 경우

- ① 수정된 login.cpp를 어딘가로 백업
- ② 원격 저장소에서 원래 상태로 복원
- ③ 수정 내역 전체를 commit
- ④ 백업해 두었던 login.cpp를 다시 복구

➔ 변경된 파일들은 무조건 commit 대상이 되기 때문

#### Git의 경우

- ① commit할 파일들만 staging area에 추가
- ② 로컬 저장소로 commit

➔ 귀찮거나 특별히 파일 구분이 필요 없을 때 -a 옵션으로 스테이지 추가와 commit을 동시에 할 수 있음

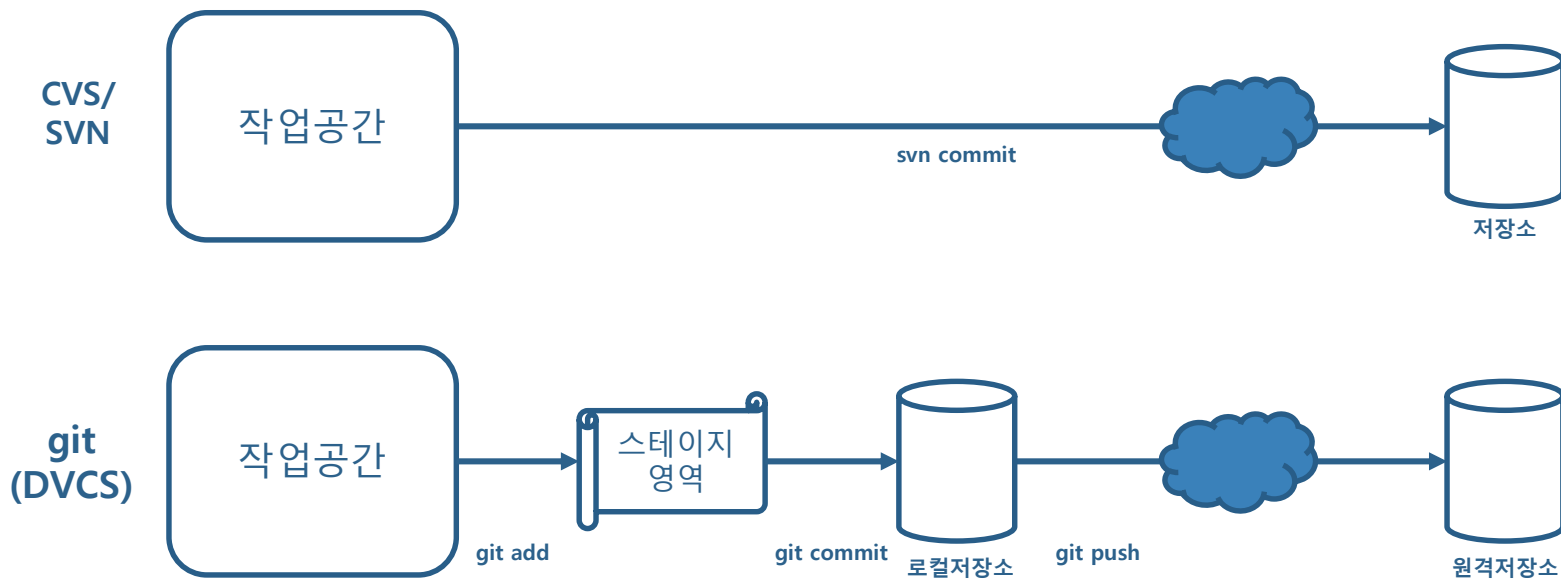
**'git commit -a'**



## 2. What is Git?

**git push**

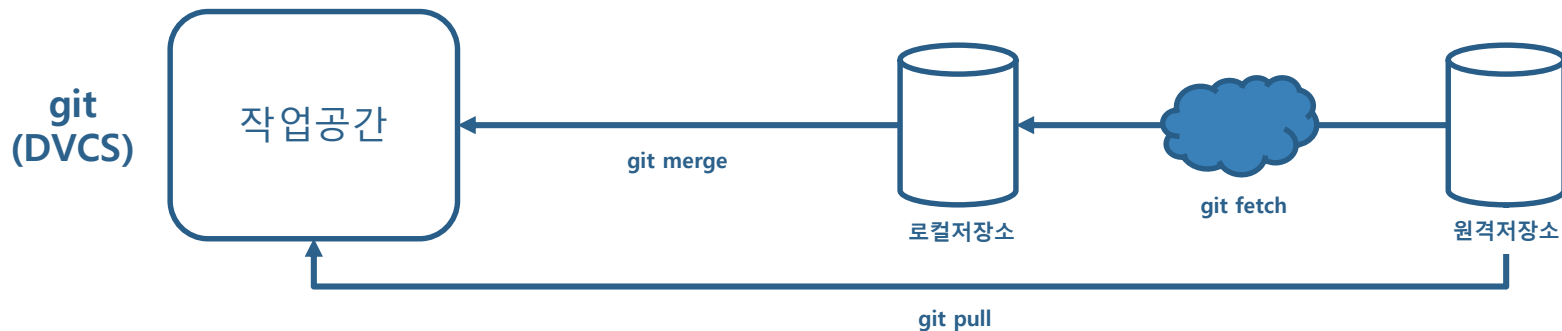
- commit을 하고 난 다음, 다른 사람들에게 작업물을 공유 (원격 저장소에 업로드)



## 2. What is Git?

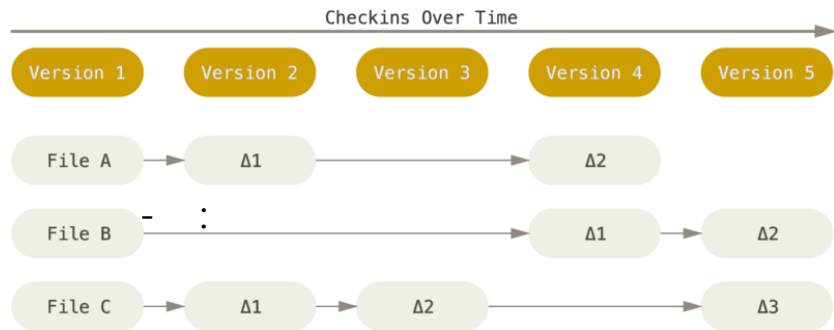
git pull

- 다른 사람들이 작업 한 내용을 내려 받기
  - 원격 저장소로부터 내려 받기: **fetch**
  - 내려 받은 데이터를 병합: **merge**
  - ➔ 위 두가지 과정을 한꺼번에 처리: **pull**

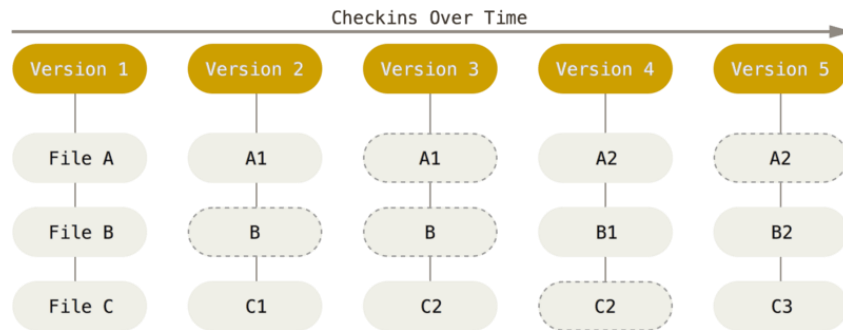


# 3. How to use Git?

- Git의 특징
  - 차이가 아닌 **Snapshot**으로 저장



대부분의 VCS들은 파일의 변화를 저장



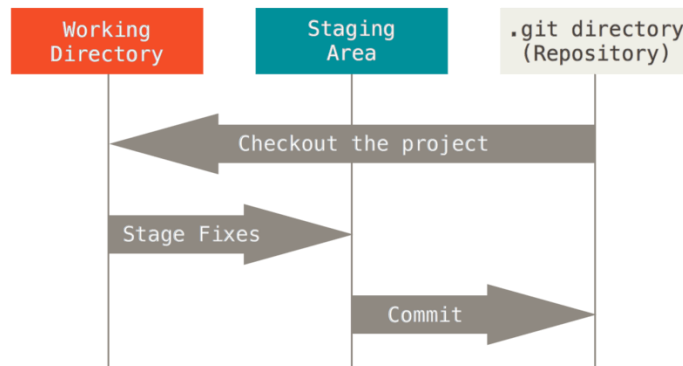
Git은 그 순간의 파일들을 저장

# 3. How to use Git?

## - Git의 특징

### - 파일을 3가지 상태로 관리

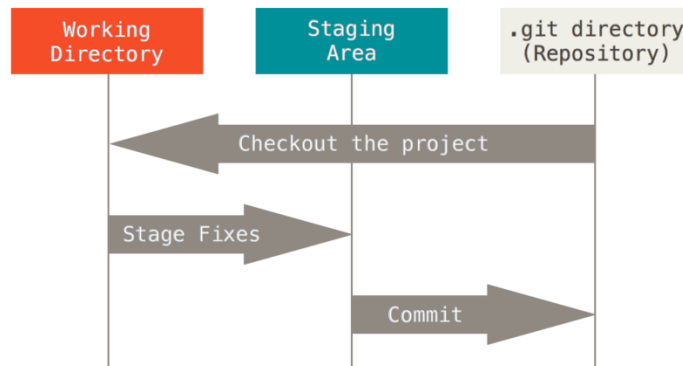
1. **committed**: 파일이 local git repository(로컬 데이터베이스)에 안전하게 저장 된 상태
2. **modified**: 사용자가 파일을 수정하였고 아직 로컬 데이터베이스에 저장(commit)하지 않은 상태  
\* commit 명령을 실행하더라도 commit 되지 않음
3. **staged**: 수정한 파일을 로컬 데이터 베이스에 저장하기 위한 준비 상태  
\* commit 명령은 staged 상태의 파일만을 commit 함



# 3. How to use Git?

## - Git이 기본적으로 하는 일

1. Working directory에서 파일을 수정한다. (modified 상태가 됨)
2. Staging Area로 파일을 Stage 시켜서 commit(커밋)을 위한 준비를 시킨다.  
\* [git add 명령을 통해 Staged 상태로 전환시킴](#)
3. Staging Area에 있는 파일들을 커밋해서 Git directory에 Snapshot(스냅샷)으로 저장한다.  
\* [git commit 명령을 통해 파일을 커밋](#)



# 3. How to use Git?

---

- Git 용어
  - Repository의 버전관리
    - **Add**  
수정된 파일을 staging area에 저장
    - **Commit**  
수정된 파일을 local git repository에 저장
    - **Push**  
Working directory의 파일 데이터를 server git repository로 저장
    - **Fetch**  
local git repository의 파일 데이터를 server git repository에 저장된 특정 버전으로 변경  
\* Working directory의 파일 데이터는 변경시키지 않음(Pull과의 차이)
    - **Merge**  
working directory의 파일 데이터를 local git repository에 저장된 특정 버전으로 변경
    - **Pull**  
Working directory의 파일 데이터를 server(remote) git repository에 저장된 특정 버전으로 변경

### 3. Git 실습 – GitLab 로그인 및 초기 설정

1. [hconnect.hanyang.ac.kr](https://hconnect.hanyang.ac.kr) 접속 후 Sign in with Hanyang 버튼 클릭



#### GitLab Community Edition

##### Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in

Register

Username or email

Password

☐ Remember me

[Forgot your password?](#)

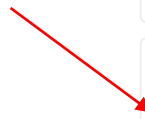
Sign in

Sign in with

Hanyang

☐ Remember me

클릭



### 3. Git 실습 – GitLab 로그인 및 초기 설정

#### 2. 한양대 포털(HY-in) 계정으로 로그인



한양대학교 | 로그인

고객님의 정보에 접근하기 위하여 인증이 필요합니다.  
한양대학교 포털 한양인(HY-in)계정으로 로그인 하시기 바랍니다.

Portal Login

---

ID	<input type="text" value="2018103446"/>	<div>로그인</div>
Password	<input type="password" value="....."/>	



# 3. Git 실습 – GitLab 로그인 및 초기 설정

## 3. 정보 제공 동의 (첫 로그인일 경우)

 **한양대학교 | 개인정보의 제 3자 제공동의 요청**

한양대학교 OPEN API는 아래와 같은 개인정보를 온라인 소프트웨어 교육 지원 시스템 - Real 에 제공합니다.

**제공 받는자**

커넥트재단 (온라인 소프트웨어 교육 지원 시스템 - REAL)

**제공 목적**

웹상에서 학생 실습코드를 저장하고 빌드 하여 채점, 코드 리뷰를 수행하는 시스템 입니다.

실습 코드 저장은 GITLABCE를 활용할 예정이고  
코드 리뷰는 REVIEW BOARD, 빌드 및 채점은 JENKINS를 사용할 예정입니다.

한양대 도메인을 통해 서비스 하고 한양대 학생 인증을 적용할 예정입니다.

**\*\* 실제 서비스를 사용자가 사용하는 환경 입니다.**

**제공 항목**

모든 항목에 동의하시여만 이용 가능합니다.

**로그인사용자 정보조회**

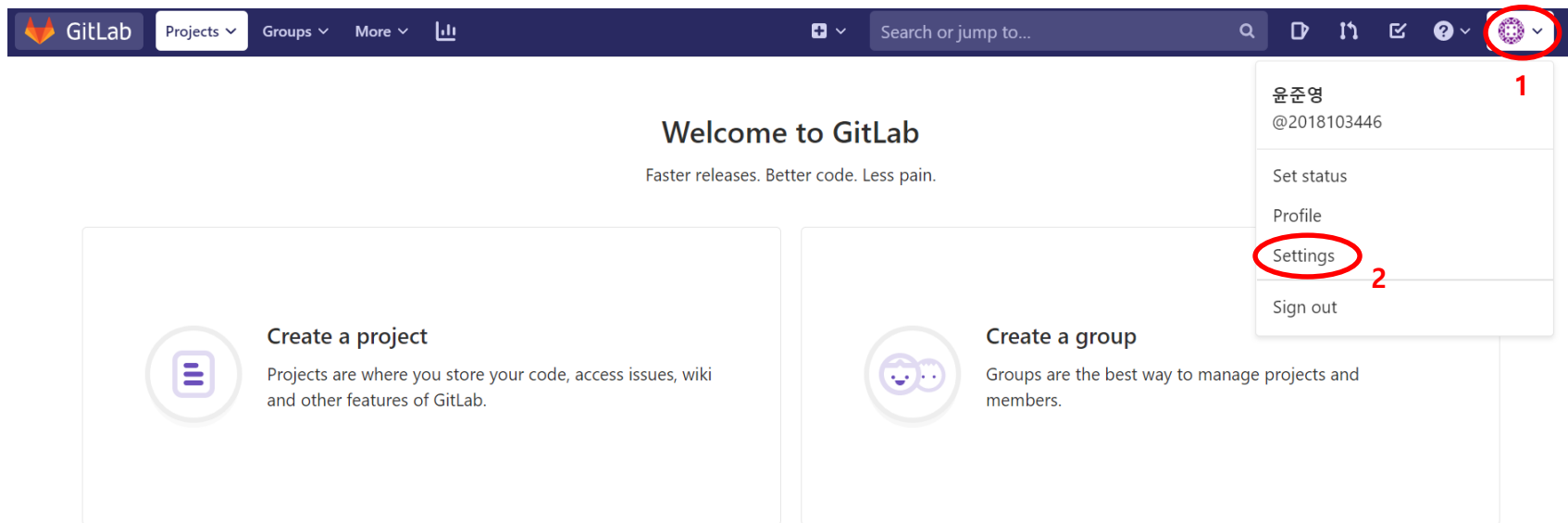
[포털에서 설정한 대표 신분 정보]  
로그인한 사용자의 성명, 사용자ID, 학번(개인번호), 제학(제직) 여부, 소속대학, 소속명, 소속코드, 소속ID, 사용자구분명의 정보를 제공합니다.

☒ 전체 동의합니다. ⓘ

☐ 동의합니다. ⓘ

### 3. Git 실습 – GitLab 로그인 및 초기 설정

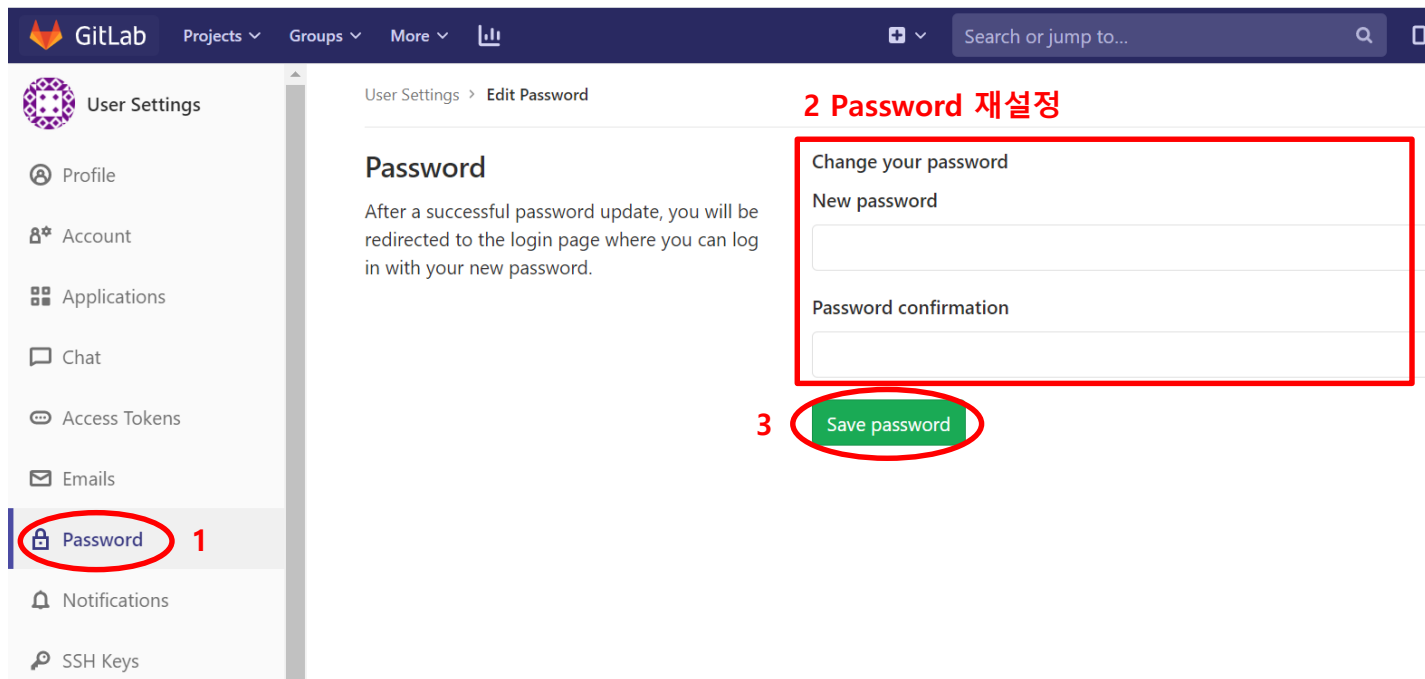
#### 4. Password 설정 (필수)



The screenshot shows the GitLab homepage. The top navigation bar is dark blue with the GitLab logo, 'Projects', 'Groups', and 'More' dropdowns. A search bar is on the right. The main content area has a 'Welcome to GitLab' message and two large cards: 'Create a project' and 'Create a group'. On the right side, a user profile dropdown menu is open, showing the user's name '윤준영' and email '@2018103446'. The 'Settings' option is circled in red and labeled with a red '2'. The profile picture icon in the top right corner is also circled in red and labeled with a red '1'.

# 3. Git 실습 – GitLab 로그인 및 초기 설정

## 4. Password 설정 (필수)



GitLab Projects Groups More

User Settings

- Profile
- Account
- Applications
- Chat
- Access Tokens
- Emails
- Password 1**
- Notifications
- SSH Keys

User Settings > Edit Password

### Password

After a successful password update, you will be redirected to the login page where you can log in with your new password.

### 2 Password 재설정

Change your password

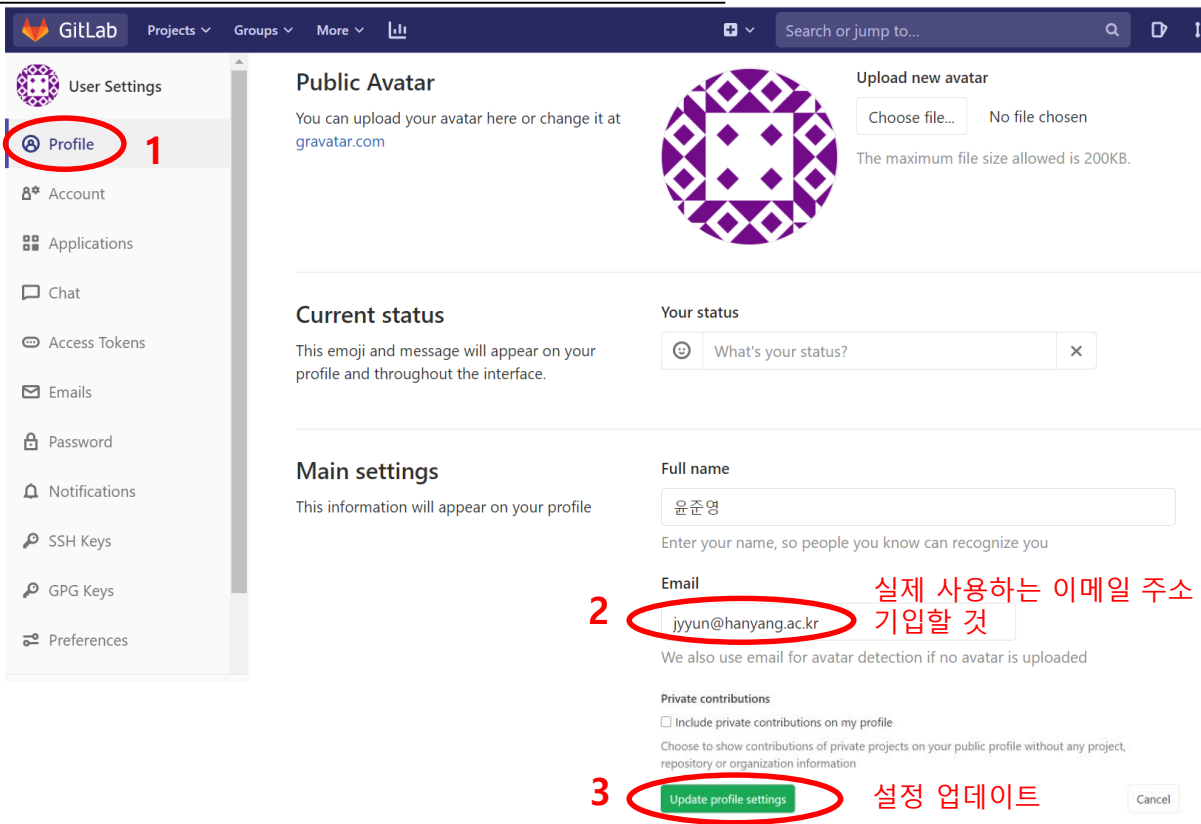
New password

Password confirmation

**3 Save password**

# 3. Git 실습 – GitLab 로그인 및 초기 설정

## 5. Email 설정 (필수)





The screenshot shows the GitLab User Settings page. The left sidebar contains a list of settings: User Settings, Profile (circled in red with a red '1'), Account, Applications, Chat, Access Tokens, Emails, Password, Notifications, SSH Keys, GPG Keys, and Preferences. The main content area is titled 'Public Avatar' and shows a placeholder for a profile picture. Below this is the 'Current status' section. The 'Main settings' section is visible, showing the 'Full name' field with the value '윤준영'. The 'Email' field is circled in red with a red '2' and contains the value 'jyyun@hanyang.ac.kr'. To the right of this field, there is a red annotation: '실제 사용하는 이메일 주소 기입할 것' (Enter the email address you actually use). Below the email field, there is a checkbox for 'Private contributions'. At the bottom of the settings, the 'Update profile settings' button is circled in red with a red '3'. To the right of this button, there is a red annotation: '설정 업데이트' (Update settings). A 'Cancel' button is also visible at the bottom right.

### 3. Git 실습 – GitLab 로그인 및 초기 설정


#### 5. Email 설정 – 변경한 email주소로 전송된 링크에서 변경 승인

HConnect <admin@hconnect.hanyang.ac.kr> 오후 9:58 (3분 전) ☆ ← ⋮  
나에게 ▾

 이 메일이 스팸으로 분류된 이유는 무엇인가요? 이전에 스팸으로 확인된 메일과 유사합니다.




---

 GitLab

[jyyun@hanyang.ac.kr](mailto:jyyun@hanyang.ac.kr)

Click the link below to confirm your email address.

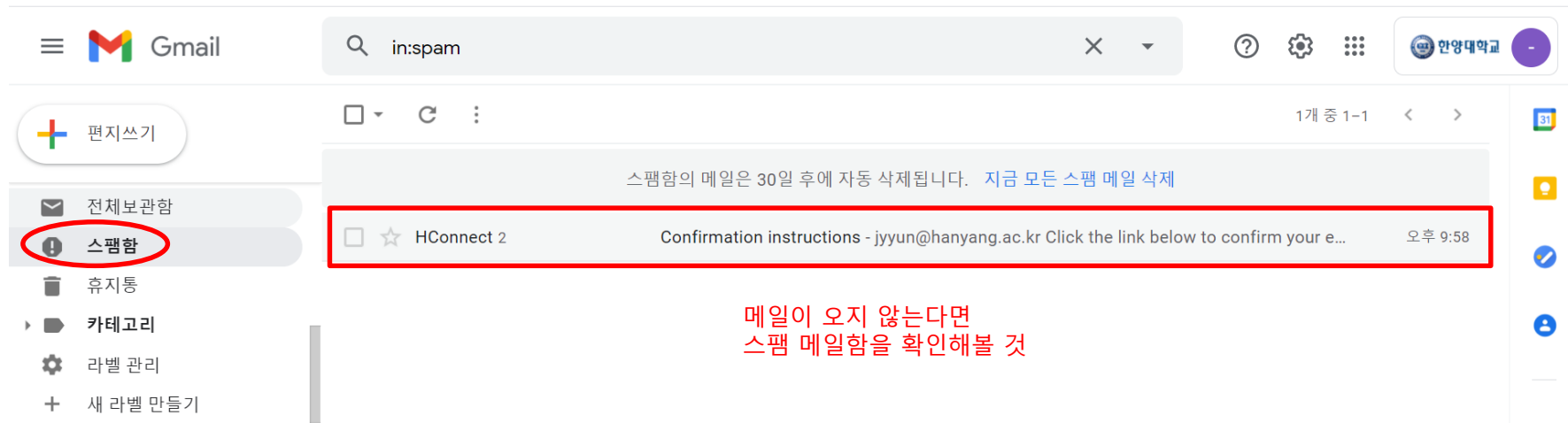
[Confirm your email address](#) 클릭하여 메일계정 등록 승인

 GitLab

You're receiving this email because of your account on [hconnect.hanyang.ac.kr](https://hconnect.hanyang.ac.kr). [Manage all notifications](#) · [Help](#)

# 3. Git 실습 – GitLab 로그인 및 초기 설정

## 5. Email 설정 – 변경한 email주소로 전송된 링크에서 승인



The screenshot shows a Gmail interface with the search bar set to 'in:spam'. The left sidebar shows the 'Spam' folder selected. The main inbox area displays a single email from 'HConnect 2' with the subject 'Confirmation instructions - jyyun@hanyang.ac.kr'. The email content is partially visible, showing a link to confirm the email address. A red box highlights the email, and a red text overlay reads: '메일이 오지 않는다면 스팸 메일함을 확인해볼 것'.

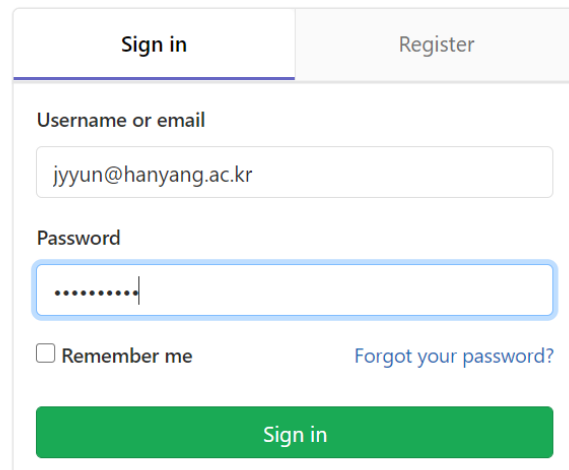
### 3. Git 실습 – GitLab 로그인 및 초기 설정

6. 이후 로그인은 등록한 Email과 변경한 password로 가능  
(Sign in with Hanyang을 거치지 않고)

#### GitLab Community Edition

##### Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

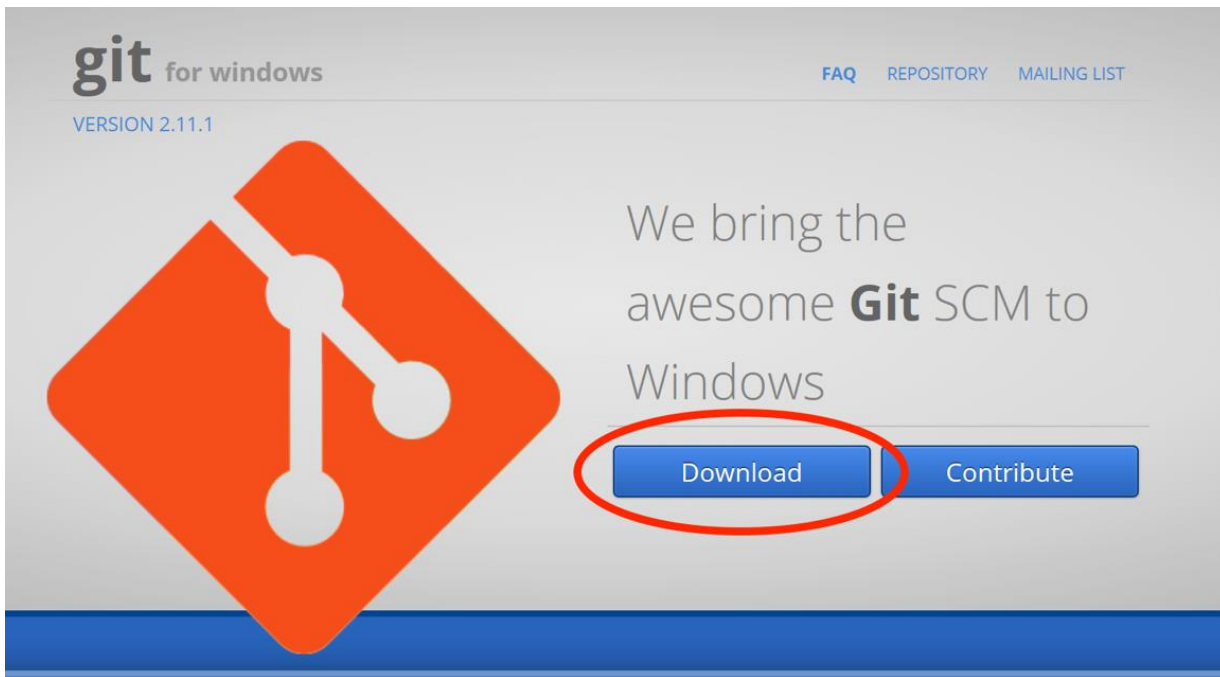


The image shows a web form for signing in to GitLab. It has two tabs at the top: 'Sign in' (active) and 'Register'. Below the tabs, there are two input fields: 'Username or email' containing 'jyyun@hanyang.ac.kr' and 'Password' containing masked characters. Below the password field, there is a checkbox for 'Remember me' and a link for 'Forgot your password?'. At the bottom, there is a green button labeled 'Sign in'.

등록한 이메일 계정 또는 password를 분실한 경우  
Sign in with Hanyang을 통해 접속 후 재설정 가능

### 3. Git 실습 – Git 설치(Windows)

1. <https://git-for-windows.github.io/> 에 접속하여 다운로드 및 설치



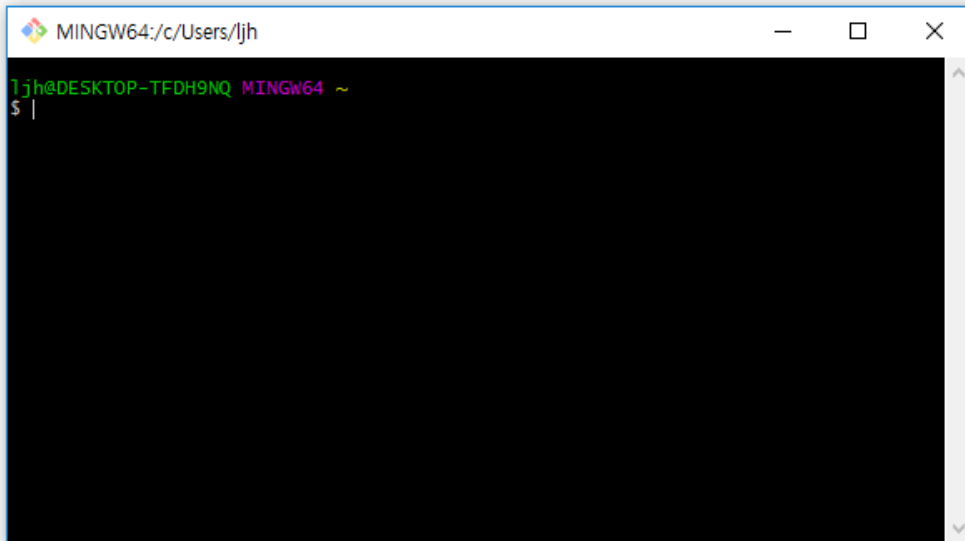


# 3. Git 실습 – Git 설치(Windows)

---

2. Git Bash 실행

3. 작업할 디렉토리로 이동 (ex: \$ cd project)

A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64: c:/Users/ljh'. The terminal content shows the prompt 'ljh@DESKTOP-TFDH9NQ MINGW64 ~' followed by a '\$' prompt and a cursor.

# 3. Git 실습 – Git 설치(Linux)

Ubuntu

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

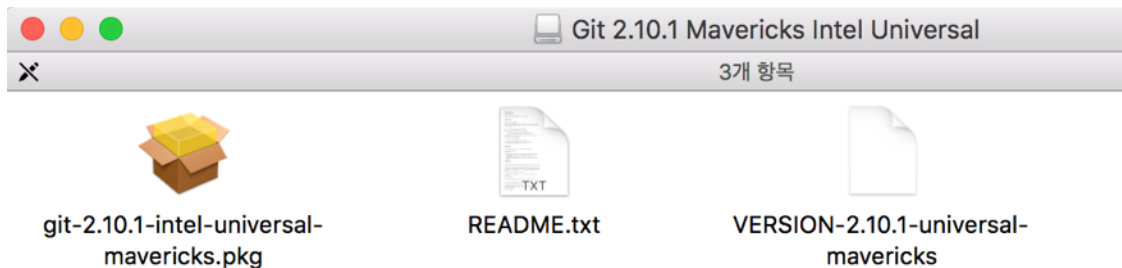
Fedora

```
$ sudo yum install git
```

```
mrbin2002 — mrbin2002@ubuntu: ~ — ssh mrbin2002@10.211.55.7 — 74x21
mrbin2002@ubuntu:~$ sudo apt-get install git
[sudo] password for mrbin2002:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
mrbin2002@ubuntu:~$
```

### 3. Git 실습 – Git 설치(Mac OSX -방법1)

1. <https://git-scm.com/download/mac> 으로 접속 후 다운로드(git-x.x.x-xxx.dmg)
2. 다운받은 파일 실행 후 git-x.x.x-xxx.pkg 파일 실행 후 설치



## 3. Git 실습 – Git 설치(Mac OSX -방법2)

1. Homebrew 설치([https://brew.sh/index\\_ko](https://brew.sh/index_ko))
2. 터미널에 \$ brew install git 명령을 실행하여 git 설치

### Download for macOS

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

#### Homebrew

Install [homebrew](#) if you don't already have it, then:

```
$ brew install git
```

### 3. Git 실습 – Git 기본 사용법 안내

1. 설치 완료 후, Git 사용자를 설정합니다.

```
$ git config --global user.name "2018103446"  
$ git config --global user.email "jyyun@hanyang.ac.kr"
```

( user.name은 학번으로, user.email은 GitLab에 등록해놓은 email로)

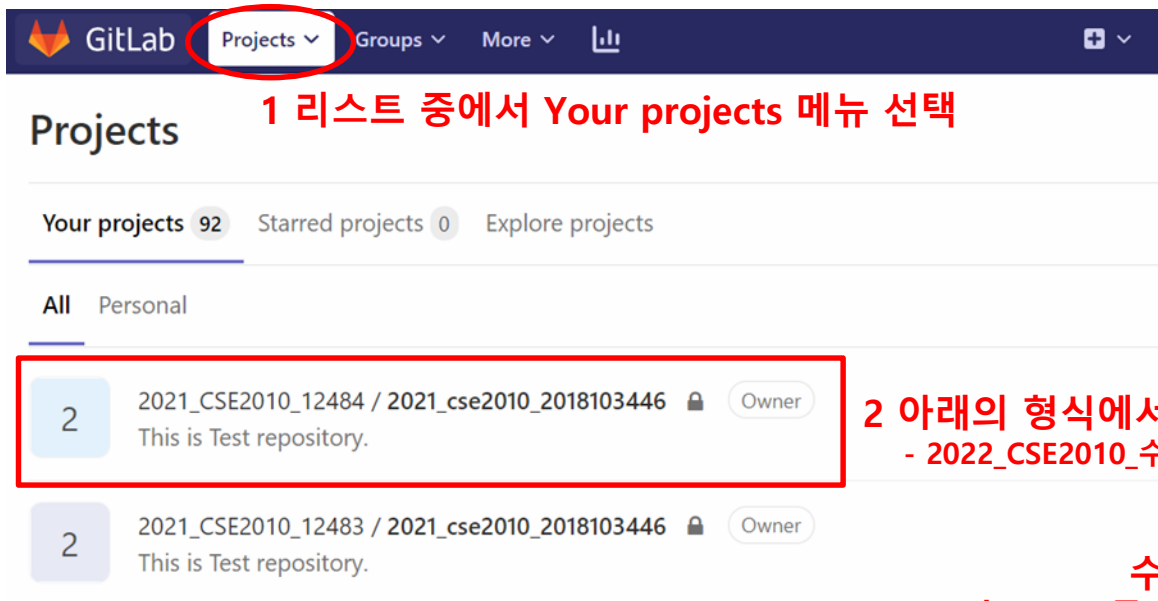
⇒ Email 변경을 하지 않았을 경우 user.email "학번@hanyang.ac.kr" (초기값)

⇒ Email 변경을 하였을 경우 user.email "등록된 email"

⇒ 추후 조교와 교수님으로부터 과제에 대한 comment를 받을 수 있기 때문에 연락이 가능한 email로 변경해야 합니다.

# 3. Git 실습 – Git 기본 사용법 안내

2. 수강중인 수업의 수업번호와 자신의 학번이 적혀 있는 프로젝트에 들어가기



GitLab Projects Groups More

Projects

1 리스트 중에서 Your projects 메뉴 선택

Your projects 92 Starred projects 0 Explore projects

All Personal

2 2021\_CSE2010\_12484 / 2021\_cse2010\_2018103446 Owner  
This is Test repository.

2 2021\_CSE2010\_12483 / 2021\_cse2010\_2018103446 Owner  
This is Test repository.

2 아래의 형식에서 수업번호와 학번 확인  
- 2022\_CSE2010\_수업번호/ 2022\_CSE2010\_학번

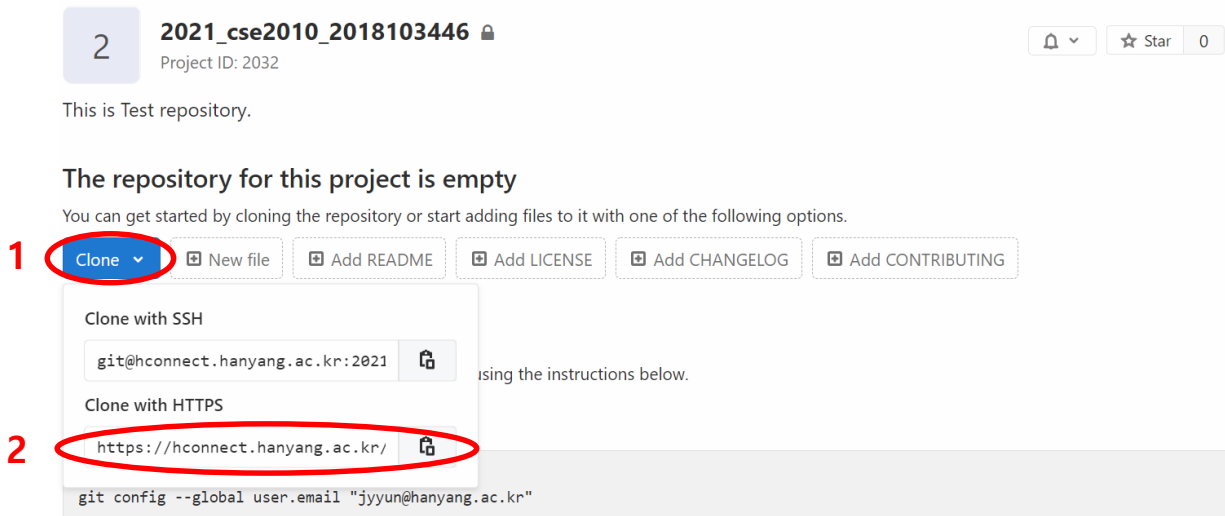
수업번호 참고  
자료구조론 목 오전 09시: 13019  
자료구조론 목 오전 11시: 13020


### 3. Git 실습 – Git 기본 사용법 안내

2. 자신의 학번이 적혀있는 프로젝트에 들어가 Git repository를 clone하여 받습니다.

```
$ git clone https://hconnect.hanyang.ac.kr/2022\_CSE2010\_수업번호/2022\_CSE2010\_학번.git
```

Git clone 주소는 GitLab webpage의 해당 프로젝트 메인화면에서 확인 가능합니다.









2 **2021\_cse2010\_2018103446**   
Project ID: 2032


This is Test repository.

The repository for this project is empty


You can get started by cloning the repository or start adding files to it with one of the following options.

1 **Clone**   New file  Add README  Add LICENSE  Add CHANGELOG  Add CONTRIBUTING

Clone with SSH

`git@hconnect.hanyang.ac.kr:2021`  using the instructions below.

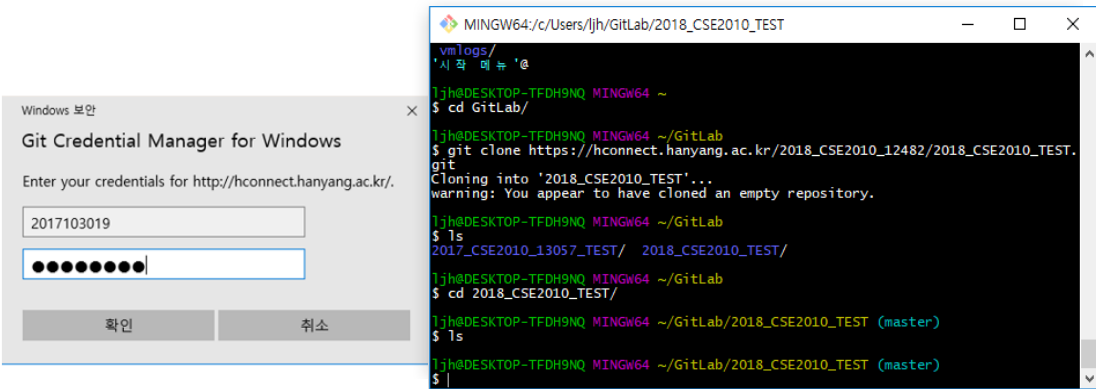
Clone with HTTPS

2 `https://hconnect.hanyang.ac.kr/` 

`git config --global user.email "jyyun@hanyang.ac.kr"`

# 3. Git 실습 – Git 기본 사용법 안내

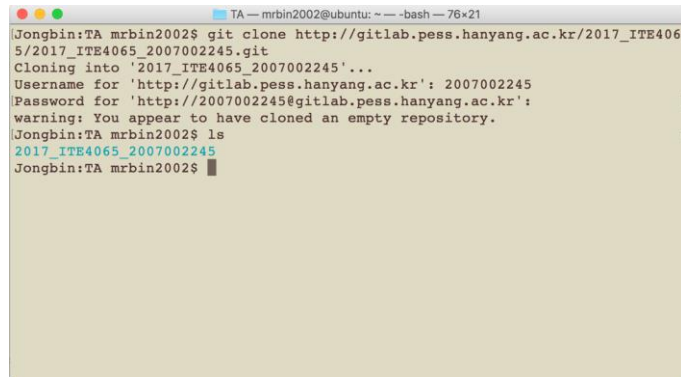
3. git clone 시 요구하는 Username은 학번으로  
Password는 GitLab webpage에서 설정한 password로 입력합니다.



The screenshot shows a Windows 10 environment. On the left, the 'Git Credential Manager for Windows' dialog box is open, prompting for credentials for 'http://hconnect.hanyang.ac.kr/'. The username '2017103019' is entered, and the password field is masked with dots. On the right, a terminal window shows the following commands and output:

```
MINGW64/c/Users/ljh/GitLab/2018_CSE2010_TEST
vmlogs/
's작 메뉴 '
$ cd GitLab/
[jh@DESKTOP-TFDH9NQ MINGW64 ~]
$ git clone https://hconnect.hanyang.ac.kr/2018_CSE2010_12482/2018_CSE2010_TEST.git
Cloning into '2018_CSE2010_TEST'...
warning: You appear to have cloned an empty repository.
[jh@DESKTOP-TFDH9NQ MINGW64 ~]
$ ls
2017_CSE2010_13057_TEST/ 2018_CSE2010_TEST/
[jh@DESKTOP-TFDH9NQ MINGW64 ~]
$ cd 2018_CSE2010_TEST/
[jh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)]
$ ls
```

Windows10



The screenshot shows a Linux terminal window with the following commands and output:

```
Jongbin:TA mrbin2002@ubuntu: ~ -- bash -- 76x21
Jongbin:TA mrbin2002$ git clone http://gitlab.pess.hanyang.ac.kr/2017 ITE4065/2017 ITE4065_2007002245.git
Cloning into '2017 ITE4065_2007002245'...
Username for 'http://gitlab.pess.hanyang.ac.kr': 2007002245
Password for 'http://2007002245@gitlab.pess.hanyang.ac.kr':
warning: You appear to have cloned an empty repository.
Jongbin:TA mrbin2002$ ls
2017 ITE4065_2007002245
Jongbin:TA mrbin2002$
```

Linux / MacOS



### 3. Git 실습 – Git 기본 사용법 안내

#### 4. Clone 받은 폴더로 이동

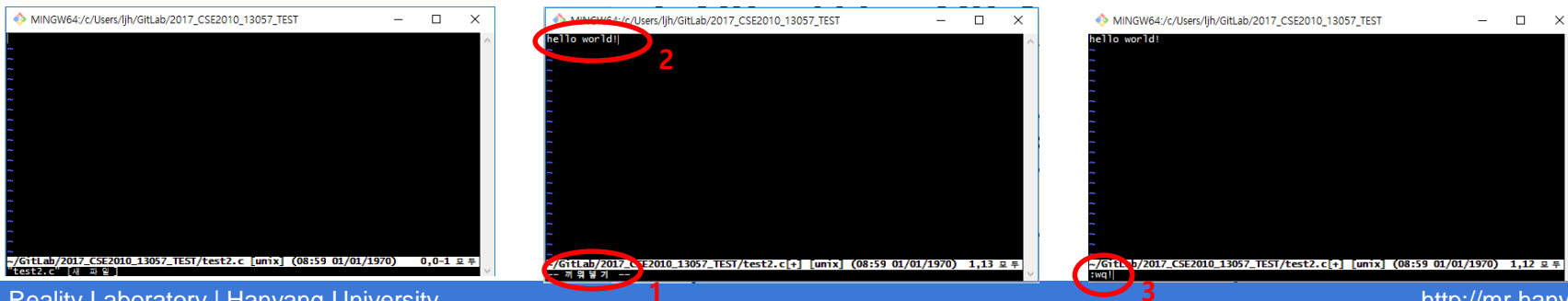
```
$ cd 2022_CSE2010_20XXXXXXXXX
```

#### 5. 작업 파일 생성

```
$ vi test2.c
```

#### 6. 파일 작성 ('i'또는 'a'키를 누르면 작성이 가능)

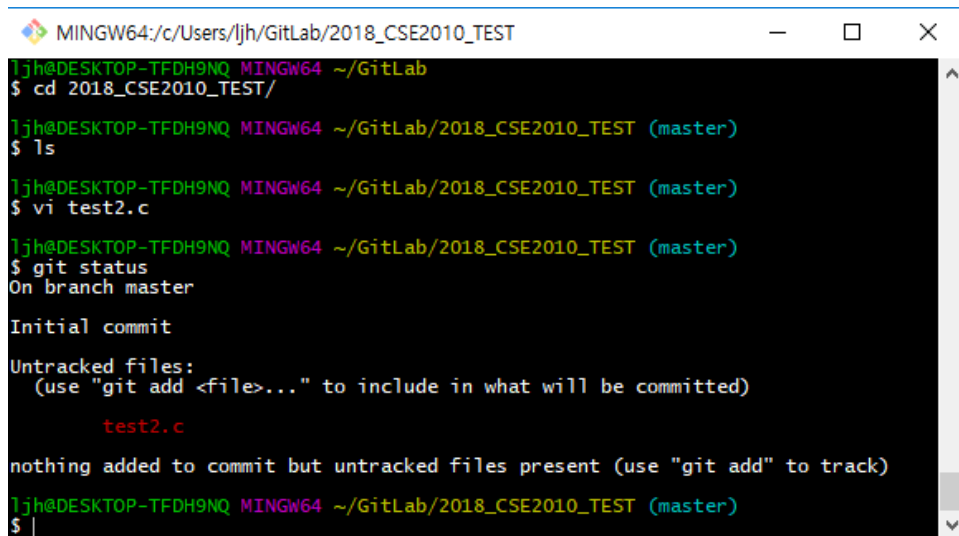
#### 7. 작성이 끝나면 'ESC'키를 누르고 ":wq!"를 입력하고 엔터키를 누르면 저장 완료



### 3. Git 실습 – Git 기본 사용법 안내

8. 현재 git 관리 상태를 확인하면 test2.c가 관리되지 않는 상태로 표시됩니다.

```
$ git status
```



```
MINGW64:/c:/Users/ljh/GitLab/2018_CSE2010_TEST
ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab
$ cd 2018_CSE2010_TEST/

ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ ls

ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ vi test2.c

ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ git status
On branch master

Initial commit

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

        test2.c

nothing added to commit but untracked files present (use "git add" to track)
ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$
```

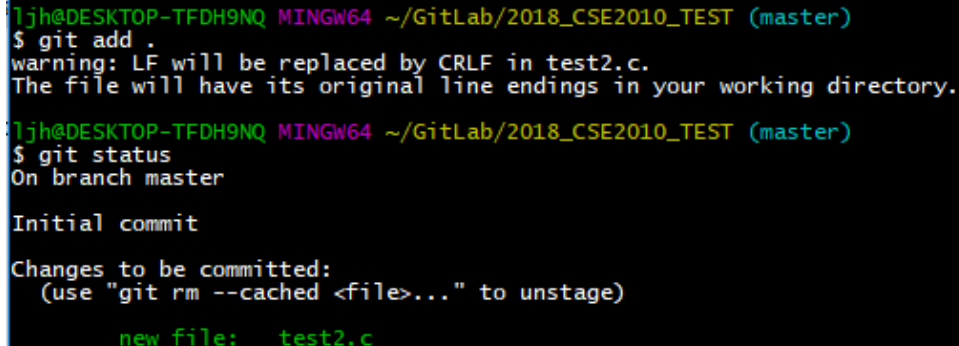
### 3. Git 실습 – Git 기본 사용법 안내

9. 현재 working directory에 있는 모든 추가/수정된 파일을 Stage 영역으로 이동 (test2.c가 git에 의해 관리됨)

```
$ git add .    또는  $ git add specific_file
```

10. Git 관리 상태를 다시 확인

```
$ git status
```



```
ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ git add .
warning: LF will be replaced by CRLF in test2.c.
The file will have its original line endings in your working directory.
ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test2.c
```

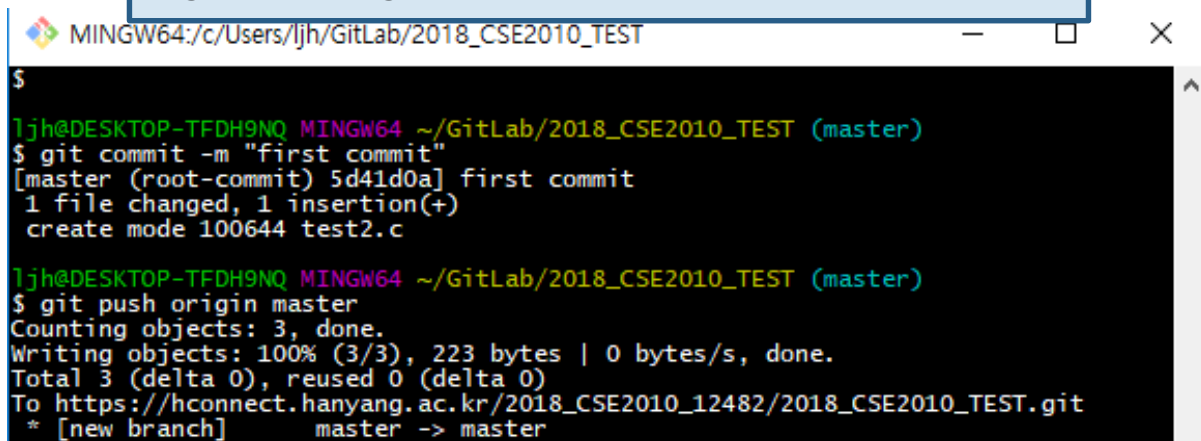
# 3. Git 실습 – Git 기본 사용법 안내

11. 추가/수정된 파일을 커밋(Local repository에 저장)

```
$ git commit -m "first commit"
```

12. 커밋된 내용을 Server repository로 전송

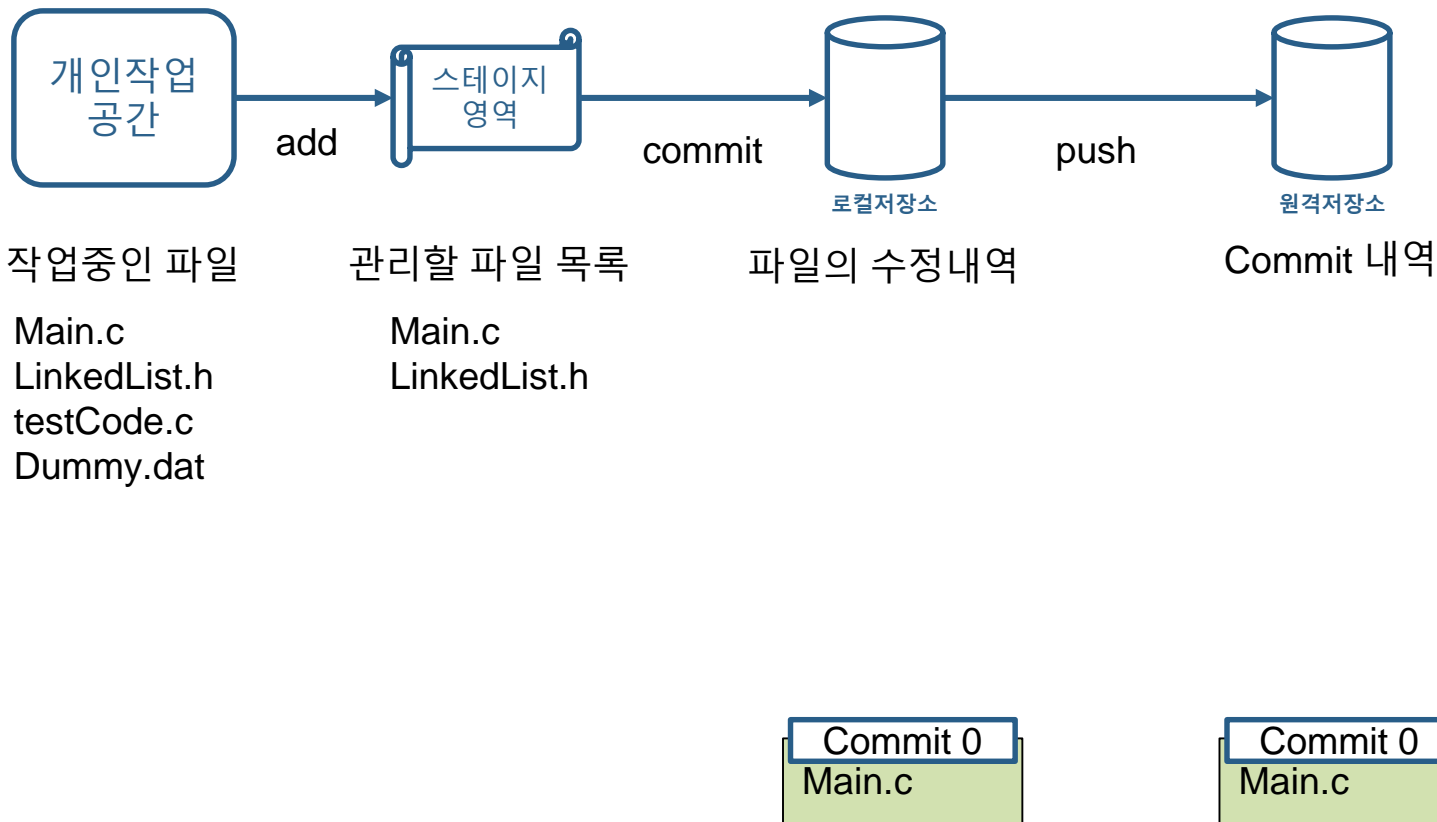
```
$ git push origin master
```



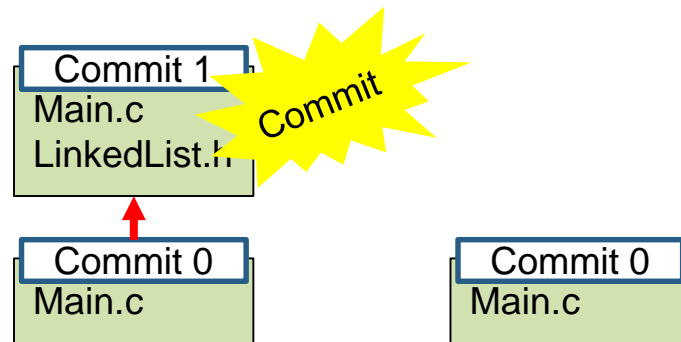
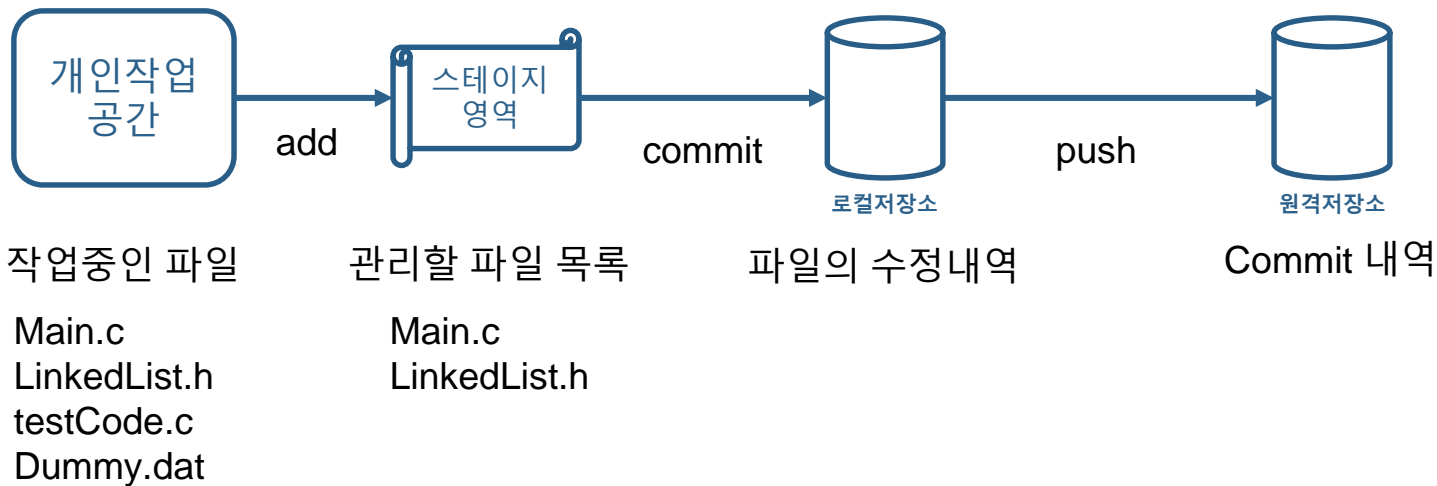
```
MINGW64:/c:/Users/ljh/GitLab/2018_CSE2010_TEST
$
ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ git commit -m "first commit"
[master (root-commit) 5d41d0a] first commit
1 file changed, 1 insertion(+)
create mode 100644 test2.c

ljh@DESKTOP-TFDH9NQ MINGW64 ~/GitLab/2018_CSE2010_TEST (master)
$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 223 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://hconnect.hanyang.ac.kr/2018_CSE2010_12482/2018_CSE2010_TEST.git
* [new branch]      master -> master
```

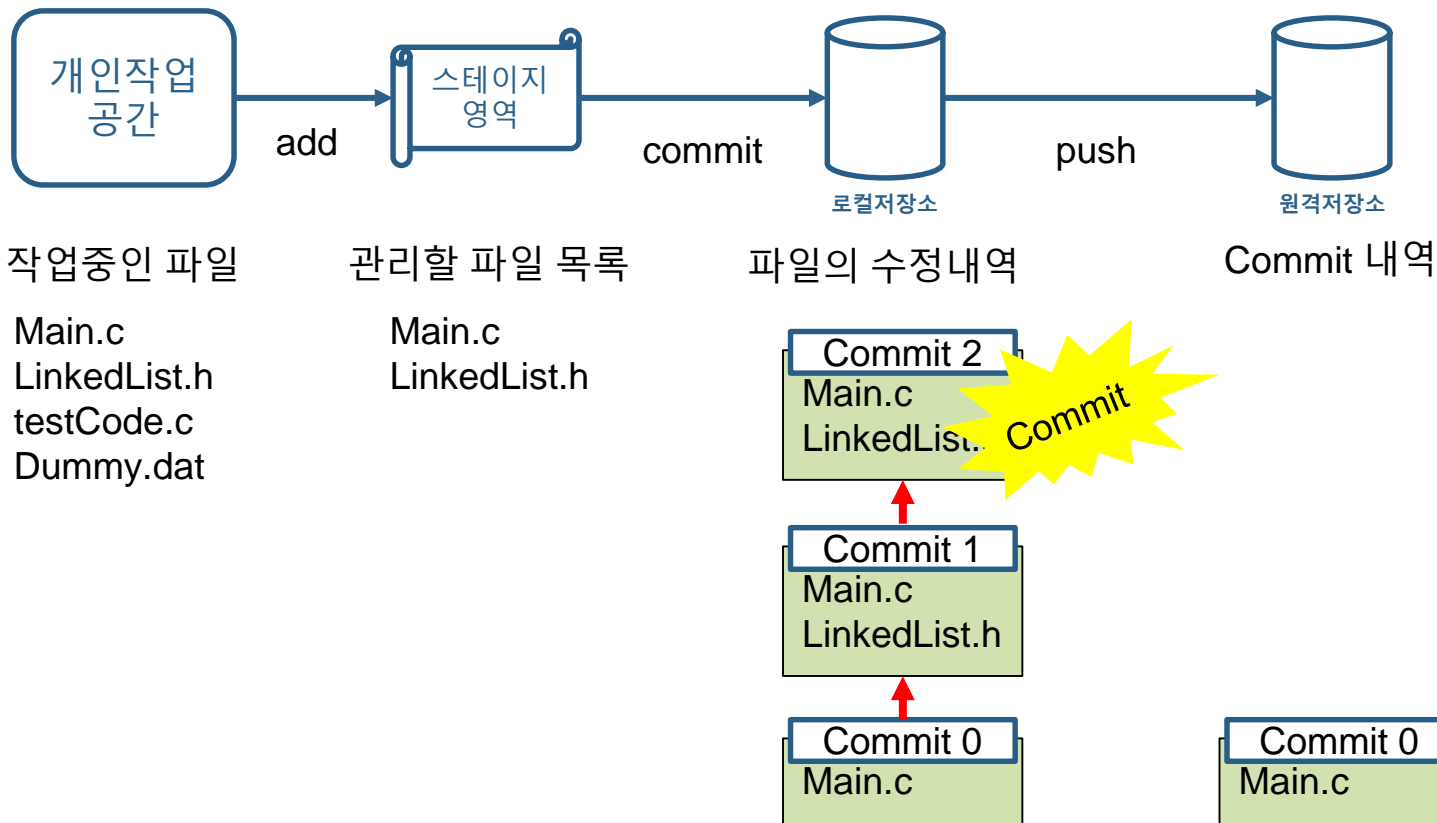
# 3. Git 실습 – Git 기본 사용법 안내



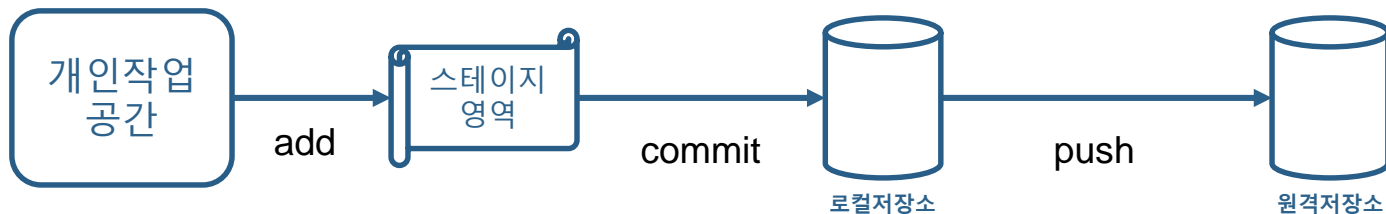
# 3. Git 실습 – Git 기본 사용법 안내



# 3. Git 실습 – Git 기본 사용법 안내



# 3. Git 실습 – Git 기본 사용법 안내



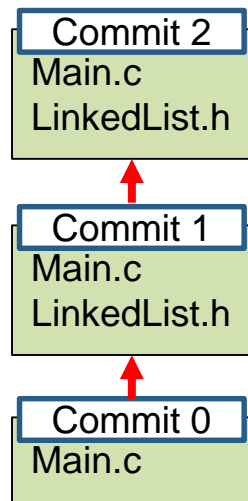
작업중인 파일

Main.c  
LinkedList.h  
testCode.c  
Dummy.dat

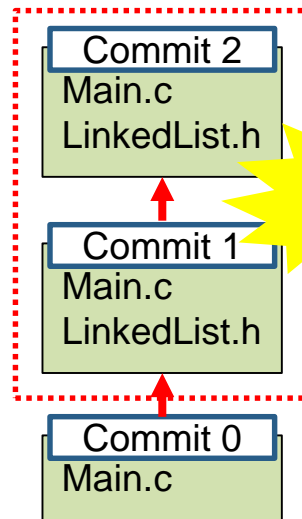
관리할 파일 목록

Main.c  
LinkedList.h

파일의 수정내역

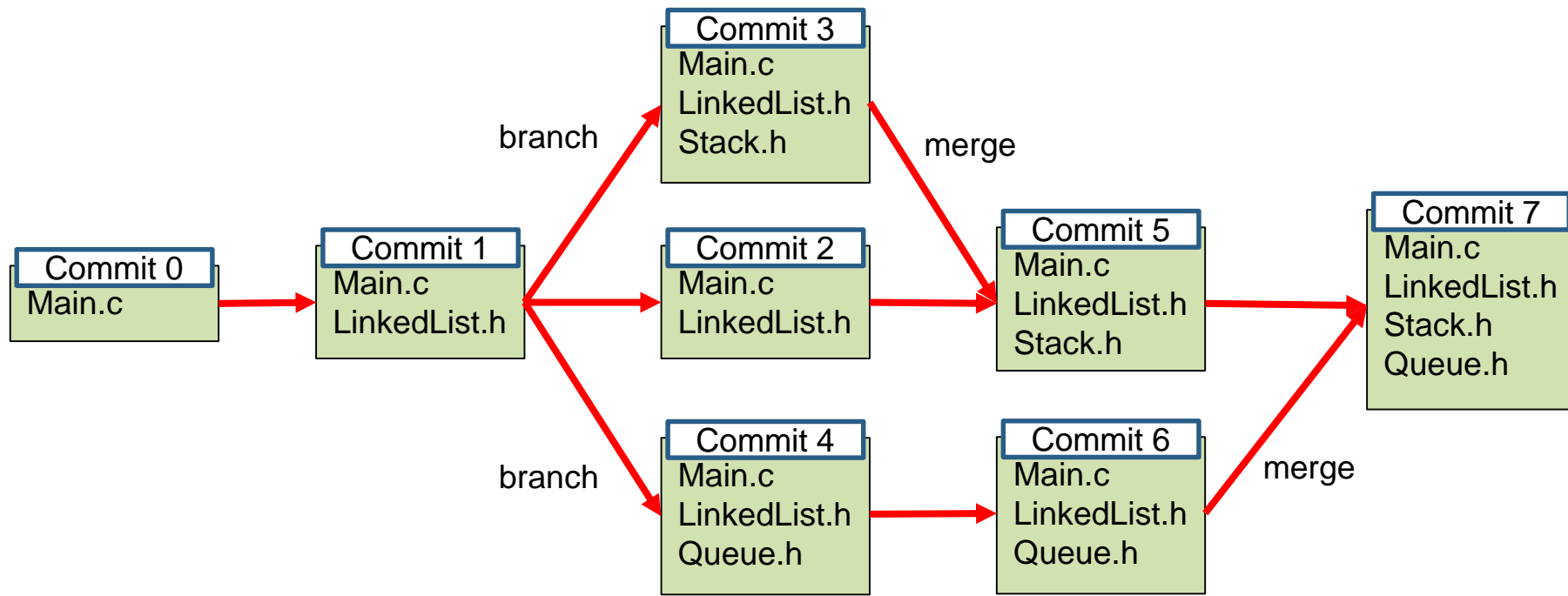


Commit 내역

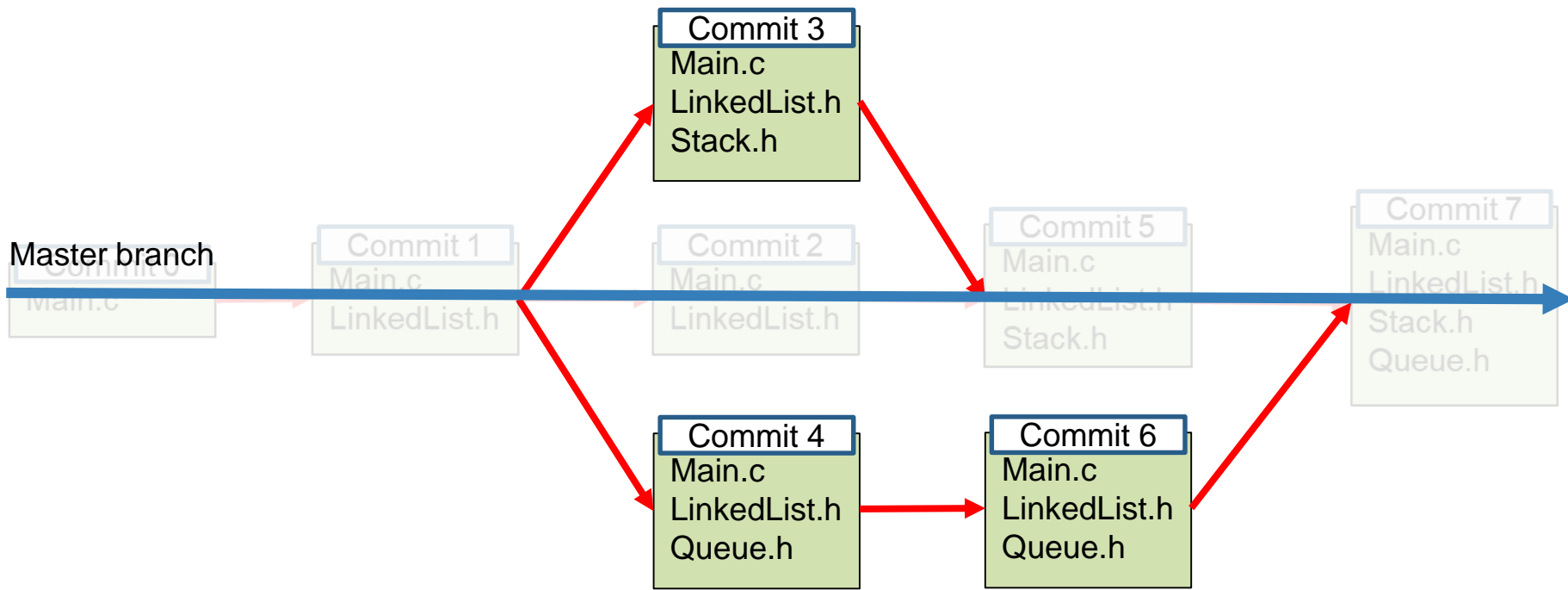




### 3. Git 실습 - Git 기본 사용법 안내



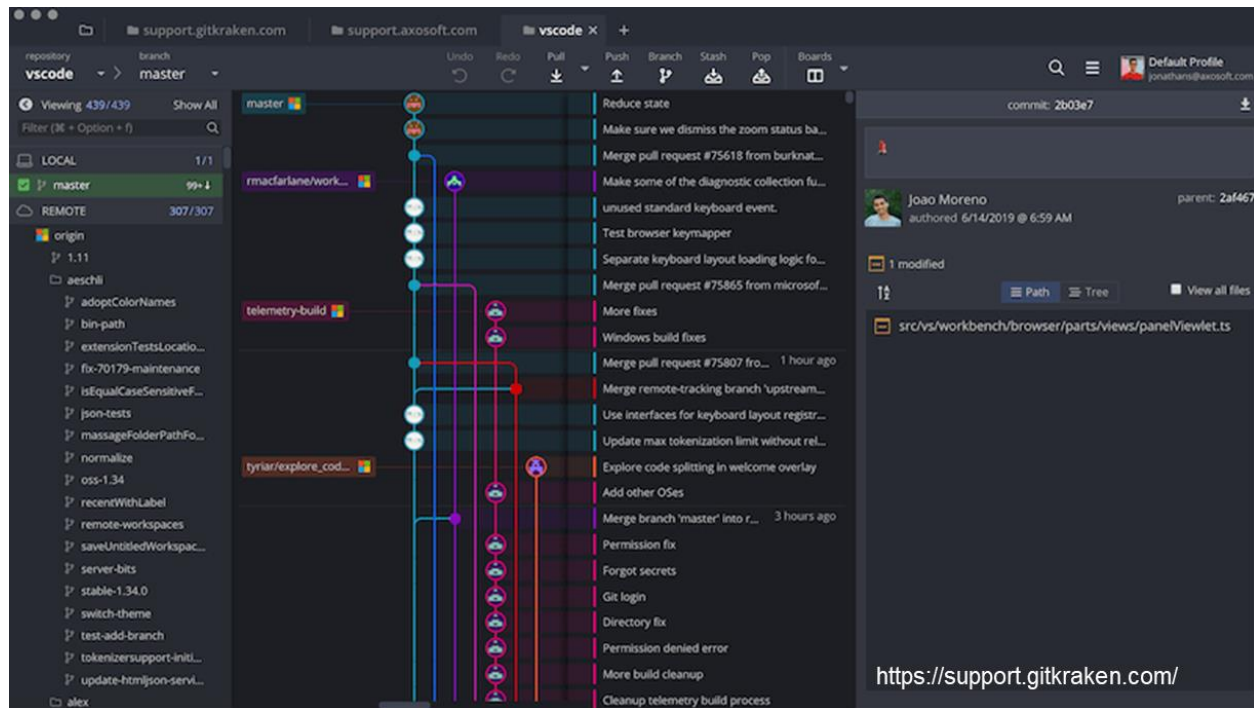
# 3. Git 실습 - Git 기본 사용법 안내



\*Master branch: 맨 처음 생성된 branch

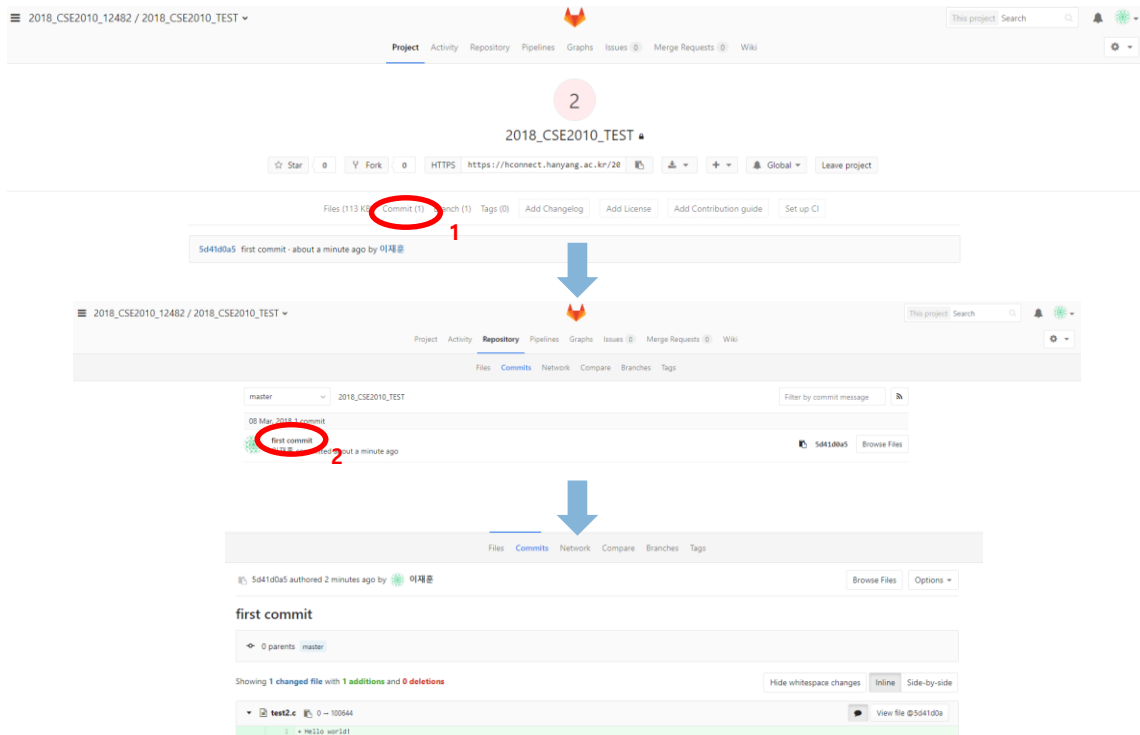
# 3. Git 실습 – Git 기본 사용법 안내

\*여러 개발자가 협업하며 개발한 내역의 예시(GitKraken을 사용한 GUI 화면)



# 3. Git 실습 – Git 기본 사용법 안내

13. Git push를 통해 Server로 전송된 파일은 GitLab webpage에서 확인 가능합니다.



The screenshot illustrates the steps to view a commit on the GitLab webpage:

1. On the repository page, click on **Commits (1)** to view the commit history.
2. On the commit details page, click on **first commit** to view the file changes.

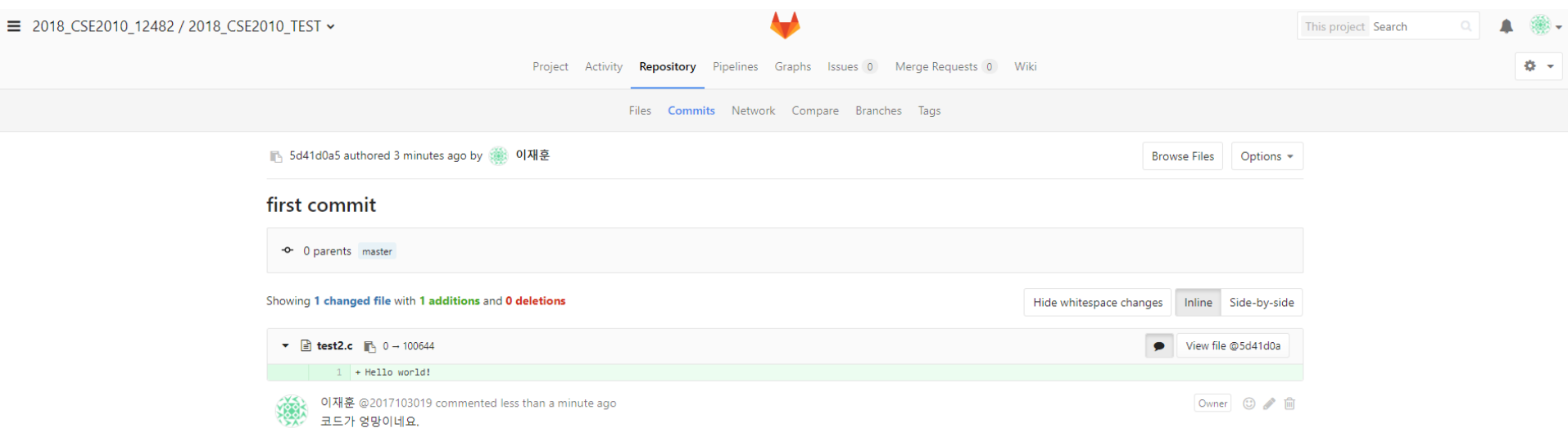
The final view shows the file `test2.c` with the following content:

```
1  // Hello world!
```

## 4. 마무리

Git으로 과제 소스코드를 올리게 되면 조교와 교수님께서 code review를 통해 comment를 남기게 되어 feedback을 받을 수 있습니다.

⇒ 학생은 email이나 GitLab webpage를 통해 해당 comment를 확인할 수 있습니다.



2018\_CSE2010\_12482 / 2018\_CSE2010\_TEST

Repository

5d41d0a5 authored 3 minutes ago by 이재훈

first commit

0 parents master

Showing 1 changed file with 1 additions and 0 deletions

test2.c 0 → 100644

```
1 + Hello world!
```

이재훈 @2017103019 commented less than a minute ago  
코드가 엉망이네요.

## 5. 과제

---

1. C 기본 문법(#define, Array, for, function)들을 활용한 코드 작성
2. 로컬 저장소에 작성한 코드 Update (Add, Commit)
3. 원격 저장소(GitLab repository)에 작성한 코드 Update (Push)

# 5. 과제

- 작성할 코드 ➡➡➡
  - 이 코드를 line by line으로 설명할 수 있을 정도
    - 자료구조론 수강에 큰 어려움이 없음
  - 만일 이 코드를 이해하는 데 어려움이 있다
    - selectionSort 함수의 인자 부분이 잘 이해가 안된다.
    - for문의 작동 방식이 머릿속으로 잘 안그려진다
    - #define??
- ➔ C언어 함수/배열/포인터/동적할당 부분 빠르게 복습 필요
- ➔ 그래도 잘 이해가 안간다면 조교를 적극 활용

```
#include <stdio.h>
#define SIZE 10

int list[SIZE] = { 38,27,1,43,3,9,82,10,16,60 };

void selectionSort(int* list, int n) {
    int i, j, least, temp;

    for (i = 0; i < n - 1; i++) {
        least = i;
        for (j = i + 1; j < n; j++)
            if (list[j] < list[least]) least = j;
        temp = list[least];
        list[least] = list[i];
        list[i] = temp;
    }
}

int main()
{
    int i;

    printf("Before Sorting\n");
    for(i=0; i<SIZE; i++)
        printf("%d ", list[i]);
    printf("\n");

    selectionSort(list, SIZE);

    printf("After Sorting\n");
    for(i=0; i<SIZE; i++)
        printf("%d ", list[i]);
    printf("\n");

    return 0;
}
```

# 감사합니다.

과제 제출 기한: 2022년 3월 16일 23:59분

\*기한 내에 본인의 원격 저장소에 push된 건에 대해서만 인정

궁금한 것이 생기면 언제든지 질문하시면 됩니다 ☺

- 공업센터본관 304호로 방문하시거나
- skagusrlf@hanyang.ac.kr 로 연락 바랍니다.
- 담당조교: 남현길