

시스템 프로그래밍

2주차: C언어 리뷰 실습



T A

강범우, IT/BT 701호

qjadr0630@hanyang.ac.kr



함수 포인터

함수의 이름을 저장하기 위한 포인터이며, 함수 호출 연산의 피연산자가 될 수 있음

선언 형식: 반환 자료형 (*변수_이름) (매개변수 자료형)

```
#include <stdio.h>
```

```
int TestFunc(int nData)
{
    puts("TestFunc() - call");
    return 0;
}
```



```
void main(void)
{
    int (*pfTestFunc)(int) = NULL;
    pfTestFunc = TestFunc;
    pfTestFunc(10);
}
```

TestFunc() - call



메모리 동적 할당

일반적으로 변수를 선언하면 명시한 자료형의 크기만큼 자동으로 메모리가 할당 및 해제되지만

필요한 메모리의 크기가 불확실한 경우 크기가 이미 정해진 변수나 배열을 선언하는 방법이 불가능할 수 있음

따라서, 이런 경우 필요한 메모리를 동적으로 할당 받아야함

자동 변수 -> Stack 영역

동적 할당 -> Heap 영역

```
void *malloc( size_t size );
```

인자

size : 할당받을 메모리의 크기(바이트 단위)이다.

반환값

힙 영역에 할당된 메모리 중 첫 번째 메모리의 주소이다. 오류가 발생하면 NULL을 반환한다.

설명

할당받은 메모리는 반드시 free() 함수를 이용하여 반환해야 하며, 메모리를 초기화하려면 memset() 함수를 이용해야 한다. 기본적으로는 쓰레기 값이 들어 있다.



메모리 동적 할당

동적 할당 `pszData = (char*)malloc(sizeof(char) * nInput);`

할당 해제 `free(pszData);`

자동 변수와 달리 동적 할당된 메모리는 반드시 `free()` 함수를 이용해 해제(반환) 해주어야 함.

할당 해제를 안 할 경우 해당 메모리가 계속 점유되어 메모리 누수 (Memory Leakage) 발생

실습 과제



실습 환경 설정

1. 이미 C 컴파일러 환경이 구축되어 있다면 그대로 사용해도 무방 (e.g., vscode)
2. 없는 경우 <https://ideone.com/> 에 접속하여 사용 언어를 "C" 로 설정 후 사용
 - 소스코드 작성 후 "Run" 버튼을 눌러 실행 및 결과 확인
3. <https://sourceforge.net/projects/orwelldvcpp/>
 - Dev-C++ 설치



실습 내용

1. 포인터를 사용하여 연결 리스트 (Linked-List)로 스택 구현
 - 실습 파일의 "practice2_stack.c" 템플릿 사용 가능
2. 함수 포인터를 사용하여 버블 정렬 구현 (오름차순/내림차순 무관)
 - 실습 파일의 "practice2_bubble.c" 템플릿 사용 가능

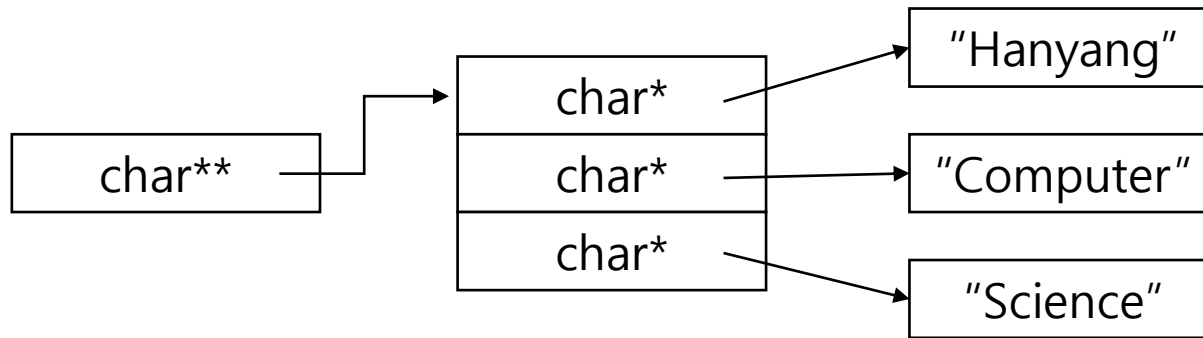


실습 내용

3.

- 1) 정수를 입력 받고 그 개수만큼 char*(문자 포인터)형을 여러 개 저장할 수 있는 메모리를 동적으로 할당
- 2) 입력할 문자의 최대 길이를 입력 받고, 해당 크기 만큼의 메모리를 동적으로 각각 할당
- 3) 입력 받은 문자열을 각 메모리에 저장한 후, 출력하는 프로그램을 작성

- 실습 파일의 "practice2_malloc.c" 템플릿 사용 가능



(메모리 구조)



실습 내용

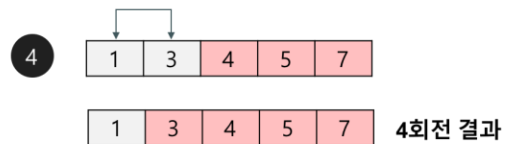
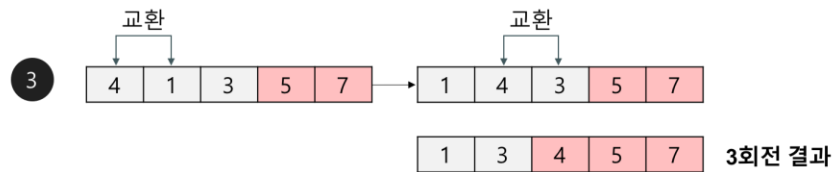
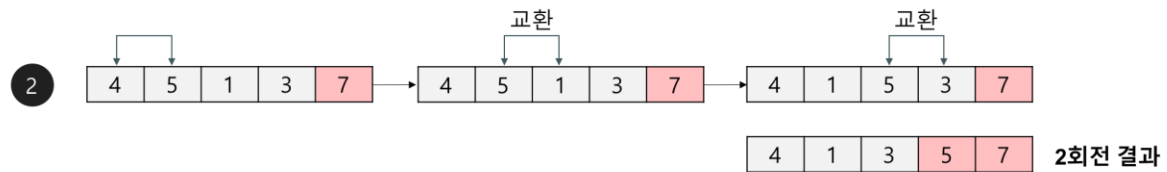
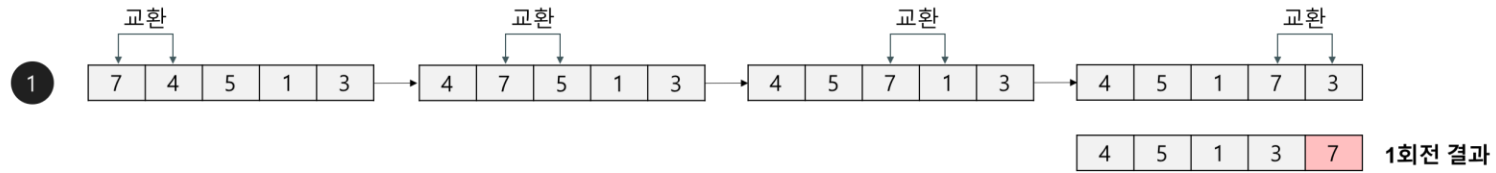
1. 본인 GitLab 저장소 안에 "Week2" 폴더 생성
2. Week2 폴더 안에 3개의 소스코드 커밋



버블 정렬

초기상태

7	4	5	1	3
---	---	---	---	---



오름차순
완성상태

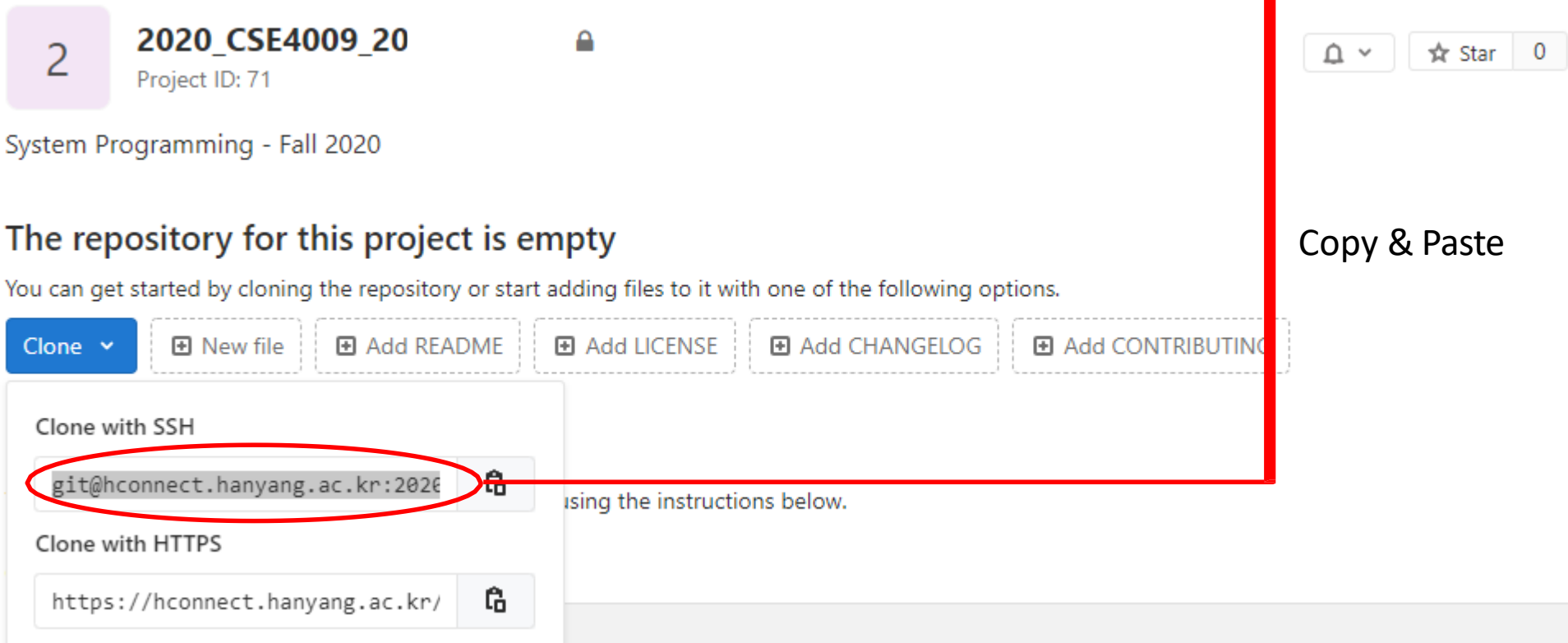
1	3	4	5	7
---	---	---	---	---



Git Basic Usage

2. Clone your Git repository

```
$ git clone https://hconnect.hanyang.ac.kr/2020_CSE4009_11781/2020_cse4009_20xxxxxxxxx.git
```



The screenshot shows a GitHub repository page for '2020_CSE4009_20' (Project ID: 71). The repository is empty. Below the repository name, there are buttons for 'Clone', 'New file', 'Add README', 'Add LICENSE', 'Add CHANGELOG', and 'Add CONTRIBUTING'. The 'Clone' button is selected, and a dropdown menu is open. In the dropdown, the 'Clone with SSH' option is selected, and the URL 'git@hconnect.hanyang.ac.kr:2020_...' is highlighted with a red oval. A red arrow points from this URL to the command line in the top block of the slide. Another red arrow points from the text 'Copy & Paste' to the same URL. The text 'The repository for this project is empty' is visible, along with the instruction 'You can get started by cloning the repository or start adding files to it with one of the following options.'

2020_CSE4009_20
Project ID: 71

System Programming - Fall 2020

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.

Clone ▾ New file Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING

Clone with SSH

git@hconnect.hanyang.ac.kr:2020_...

Clone with HTTPS

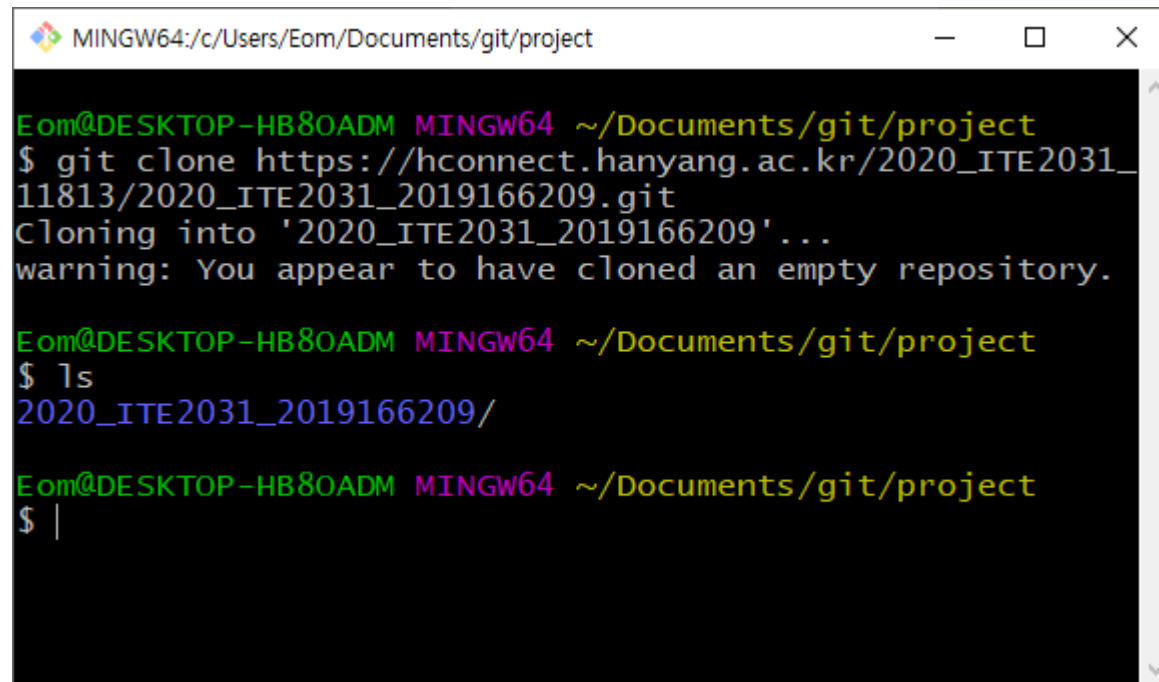
https://hconnect.hanyang.ac.kr/...

Copy & Paste



Git Basic Usage

3. Enter username(student id) and password(set in GitLab)

A screenshot of a Windows terminal window titled 'MINGW64:/c:/Users/Eom/Documents/git/project'. The terminal shows the execution of the 'git clone' command to clone a repository from Hanyang University. The output indicates the repository was cloned successfully, although it is empty. The user then runs 'ls' to list the contents of the newly created directory.

```
MINGW64:/c:/Users/Eom/Documents/git/project

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ git clone https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_2019166209.git
Cloning into '2020_ITE2031_2019166209'...
warning: You appear to have cloned an empty repository.

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ ls
2020_ITE2031_2019166209/

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ |
```



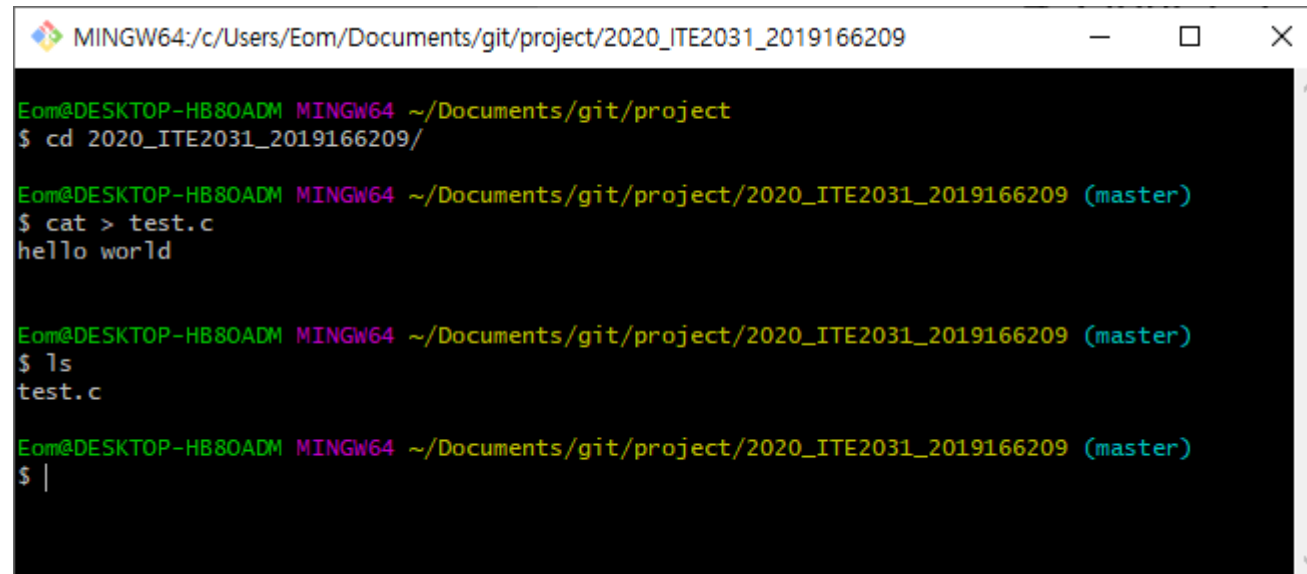
Git Basic Usage

4. Move to cloned directory

```
$ cd 2020_CSE4009_XXXXXXXXXX
```

5. Create file

```
$ vi test.c
```



```
MINGW64:/c:/Users/Eom/Documents/git/project/2020_ITE2031_2019166209
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project
$ cd 2020_ITE2031_2019166209/

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ cat > test.c
hello world

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ ls
test.c

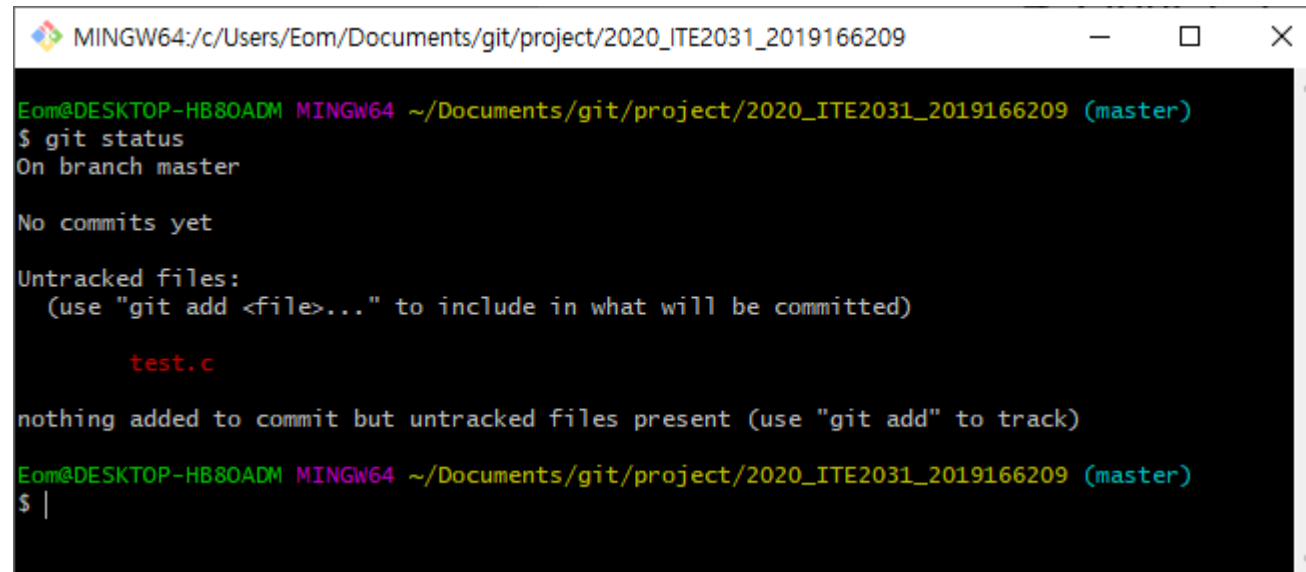
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ |
```



Git Basic Usage

6. If you check current status, created file is categorized as untracked.

```
$ git status
```



```
MINGW64:/c:/Users/Eom/Documents/git/project/2020 ITE2031_2019166209
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020 ITE2031_2019166209 (master)
$ git status
On branch master

No commits yet

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

        test.c

nothing added to commit but untracked files present (use "git add" to track)
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020 ITE2031_2019166209 (master)
$ |
```



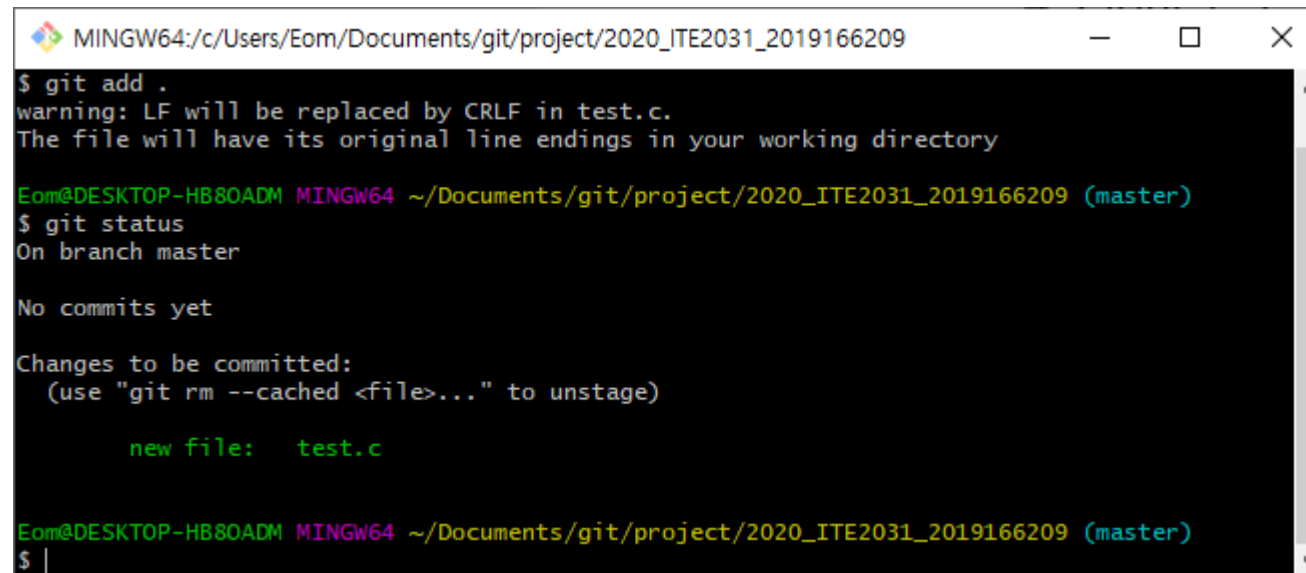
Git Basic Usage

7. Move all created or modified files in directory to staged area

```
$ git add .
```

8. Check status again

```
$ git status
```



```
MINGW64:/c:/Users/Eom/Documents/git/project/2020_ITE2031_2019166209
$ git add .
warning: LF will be replaced by CRLF in test.c.
The file will have its original line endings in your working directory

Eom@DESKTOP-HB80ADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ git status
On branch master

No commits yet

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

        new file:   test.c

Eom@DESKTOP-HB80ADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ |
```



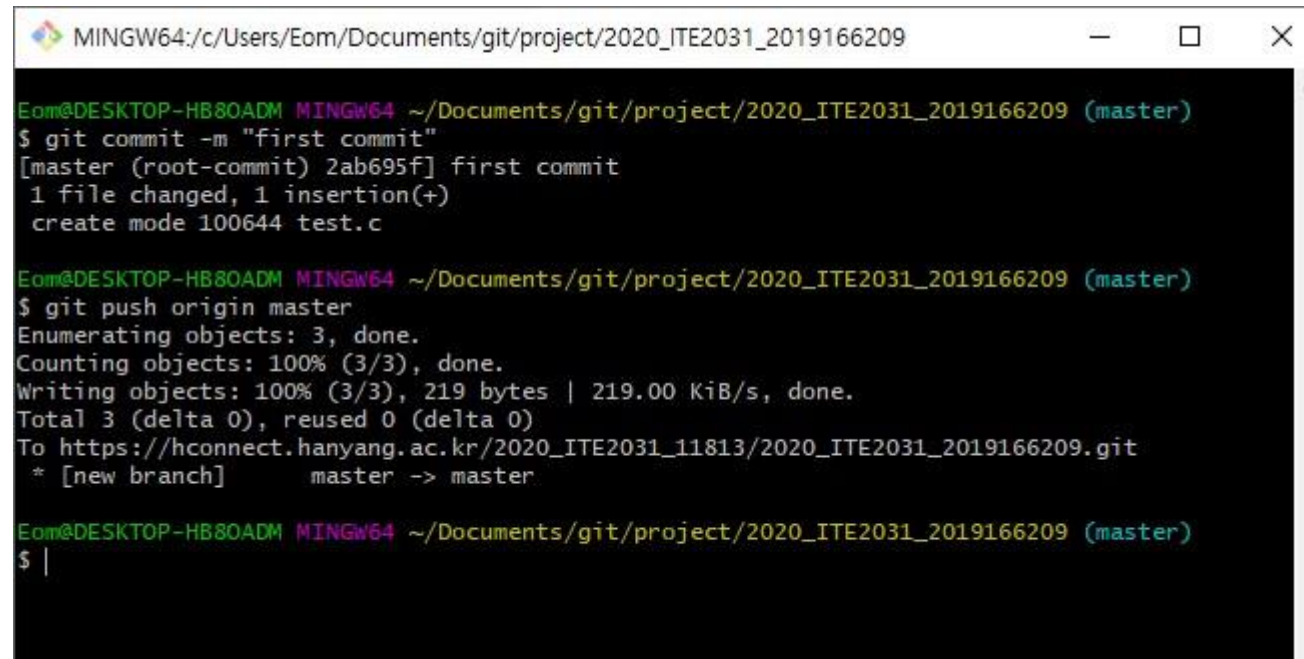
Git Basic Usage

9. Commit added or modified files(affect only local repository)

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

10. Push commits to remote repository

```
$ git push origin master
```

A screenshot of a Windows Command Prompt window titled "MINGW64: c:/Users/Eom/Documents/git/project/2020_ITE2031_2019166209". The window shows the execution of two Git commands. The first command is "git commit -m 'first commit'", which results in a successful commit with hash 2ab695f, creating a new file "test.c". The second command is "git push origin master", which successfully pushes the commit to the remote repository at "https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_2019166209.git". The output shows the progress of enumerating, counting, and writing objects, and confirms that a new branch "master" has been created on the remote.

```
MINGW64: c:/Users/Eom/Documents/git/project/2020_ITE2031_2019166209

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ git commit -m "first commit"
[master (root-commit) 2ab695f] first commit
1 file changed, 1 insertion(+)
create mode 100644 test.c

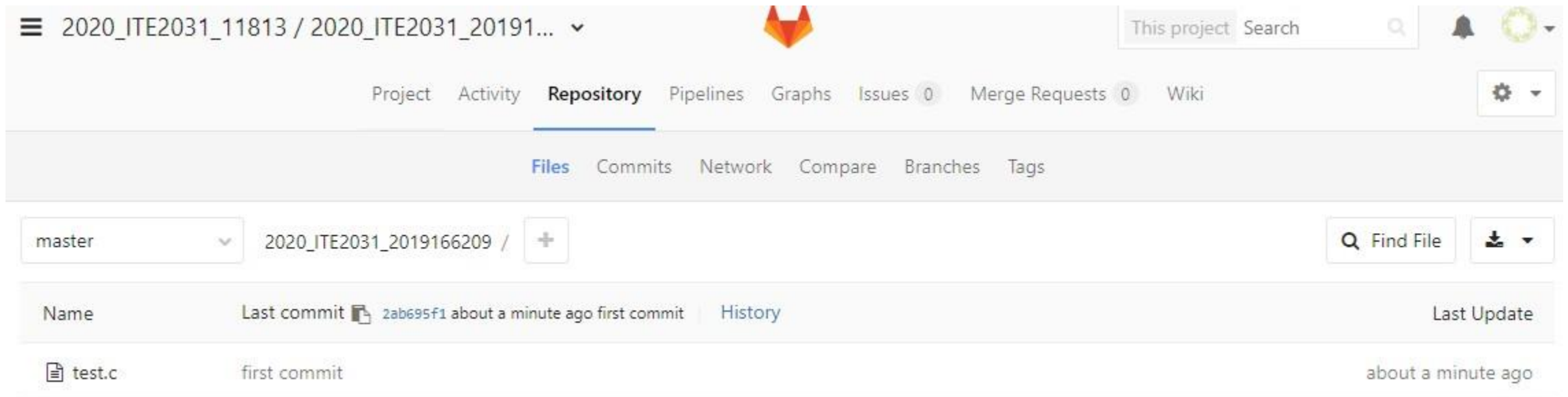
Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://hconnect.hanyang.ac.kr/2020_ITE2031_11813/2020_ITE2031_2019166209.git
 * [new branch]      master -> master

Eom@DESKTOP-HB8OADM MINGW64 ~/Documents/git/project/2020_ITE2031_2019166209 (master)
$ |
```



Git Basic Usage

11. Commits sent to the remote repository via git push are visible in the GitLab webpage



The screenshot shows the GitLab web interface for a repository. The top navigation bar includes a hamburger menu, the repository name "2020_ITE2031_11813 / 2020_ITE2031_20191...", the GitLab logo, a search bar, and notification icons. Below this is a secondary navigation bar with tabs for Project, Activity, Repository (selected), Pipelines, Graphs, Issues (0), Merge Requests (0), and Wiki. A third navigation bar contains links for Files (selected), Commits, Network, Compare, Branches, and Tags. The main content area shows the "master" branch selected, with a path "2020_ITE2031_2019166209 /" and a "+" button to add new files. A "Find File" search bar and a download icon are also present. Below this is a table of files with columns for Name, Last commit, and Last Update. The table shows a single file "test.c" with a commit hash "2ab695f1" and a message "first commit" made "about a minute ago".

Name	Last commit	Last Update
test.c	first commit	about a minute ago

