2020년도 2학기 컴퓨터공학설계및실험I

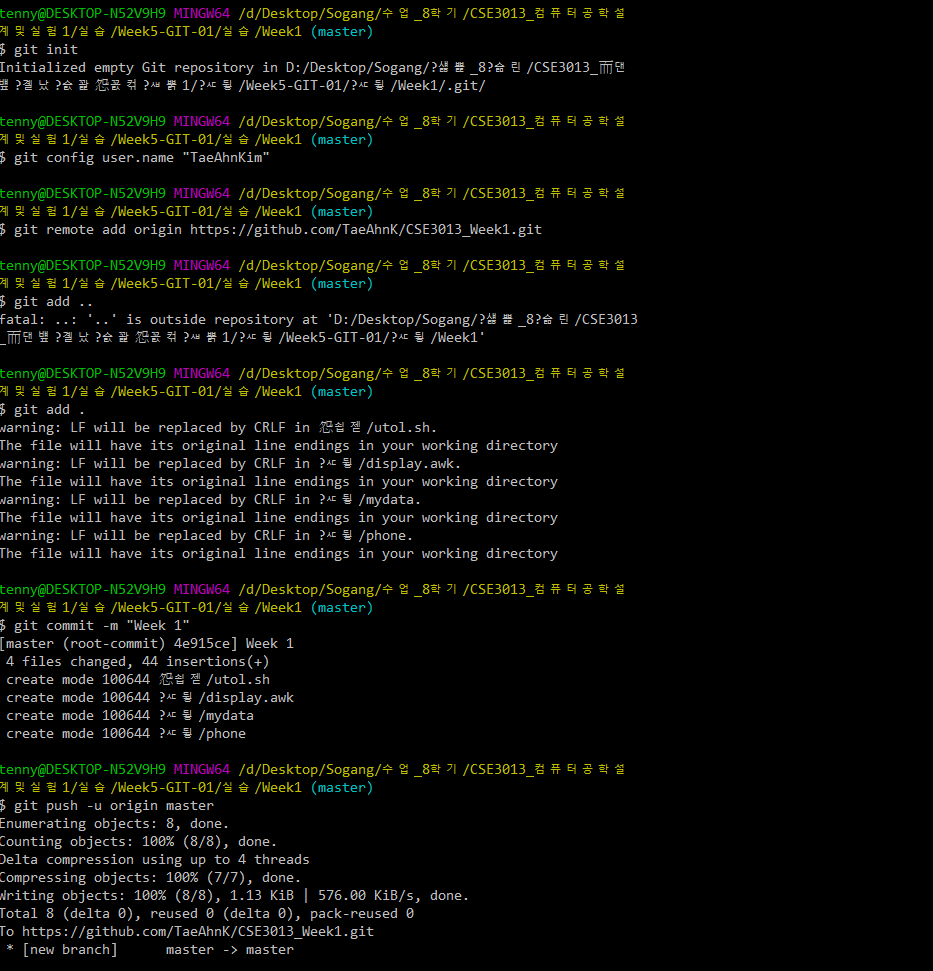
5주차 결과 보고서

20170175 김태안

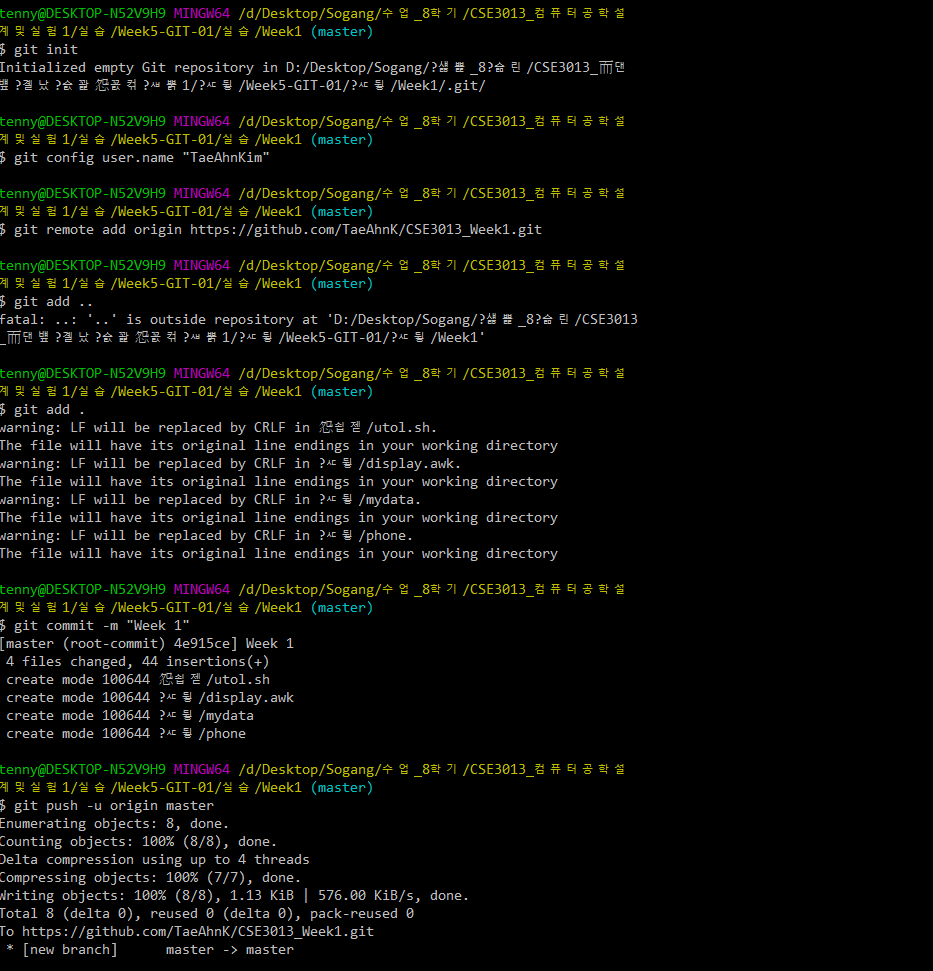
1. 실습 목적

버전 관리 시스템이 무엇인지 이해하고, Git과 GitHub의 사용법을 익힌다.

1. 실습 구현 내용
2. **push**

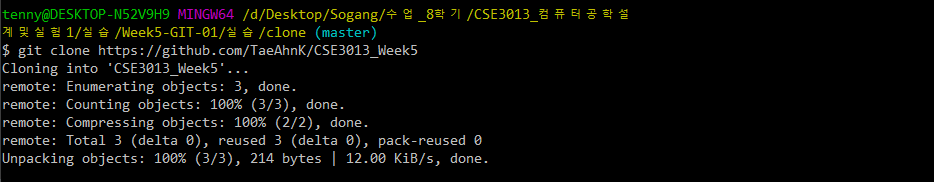


Git을 사용하기 위해 폴더에서 Git Bash를 실행한다. 그리고, git init 명령어를 통해 Local Repository를 만든다. config user.name 명령어를 통해 사용자의 이름을 등록할 수 있다. 생성한 Repository를 git remote add origin (주소) 명령어를 통해 GitHub의 Remote Repository와 연결한다. 연결한 이후 push 명령어를 사용하면 해당 Repository로 파일이 업로드된다.

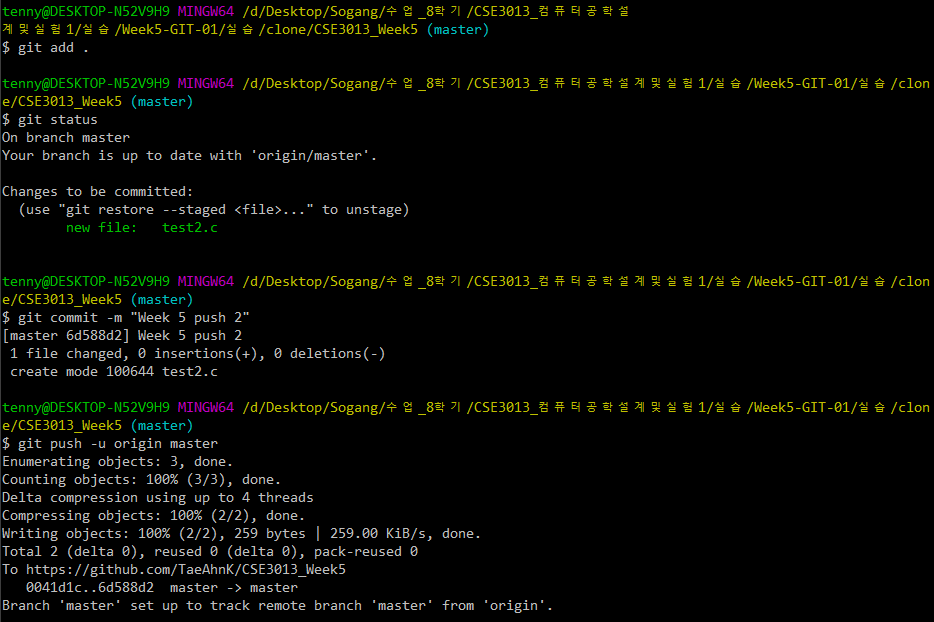


폴더에 업로드할 실습 파일을 모두 저장한 뒤, git add 명령어를 통해 git에 현재 폴더의 변경된 파일과 새 파일을 Staging Area에 추가한다. git add .을 사용하면 폴더 내의 모든 파일을 한번에 추가할 수 있다. git commit 명령어를 사용해 Staging Area에 있는 내용을 Local Repository에 저장한다. 이때, -m 인자를 통해 바로 커밋 메시지를 추가할 수 있다. Local Repository에 파일을 저장한 뒤, push (Remote Repository) (현재 브렌치) 명령어를 통해 Remote Repository로 파일을 업로드할 수 있다. 이때 사용한 Remote Repository는 origin, 브렌치는 master이다. 이와 같은 방법으로 1주차부터 4주차까지의 실습 결과를 GitHub의 Repository에 저장하였다.

1. **clone**



먼저, 복제하고 싶은 Remote Repository의 주소를 찾는다. 실습에 사용한 Repository는https://github.com/TaeAhnK/CSE3013\_Week5 로 test1.c, test1.h 파일을 1번과 같은 방식으로 push했으며, Week 5 push라는 커밋 메시지를 작성했다. git clone (주소) 명령어를 사용하면 Repository를 복제할 수 있다. 현재 git bash 주소 아래에 복제한 Repository 폴더가 생성된다. 새 폴더에서 Git Bash를 실행하거나 cd 명령어를 통해 현재 디렉토리를 복제한 Repository로 변경한다. 이 Repository는 이미 Git에서 사용하고 있는 Repository로 git init을 통해 초기화할 필요가 없다.



임의의 파일 test2.c를 추가해 Local Repository에 저장한다. 이를 push 명령어를 통해 Remote Repository에도 반영되도록 한다. 커밋 메시지는 Week 5 push 2로 설정했다.

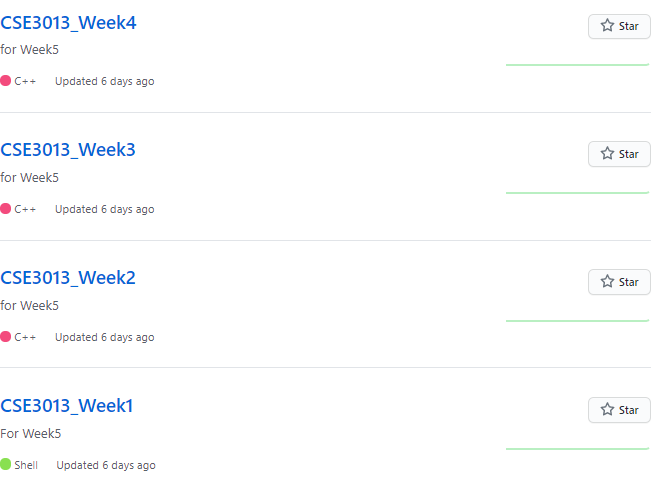
1. 실습 환경

OS: Microsoft Windows 10 Home 10.0.19041 N/A 빌드 19041

git: git version 2.27.0.windows.1

Visual Studio 2019: Microsoft Visual Studio Community 2019 버전 16.7.5

1. 실습 결과 및 분석
2. **push**



GitHub Repository에 4주간의 실습이 모두 push된 것을 확인할 수 있다.

GitHub Repository 주소

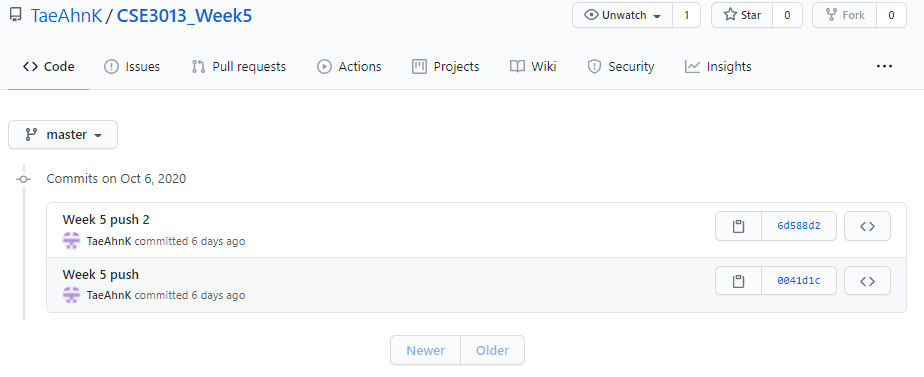
<https://github.com/TaeAhnK/CSE3013_Week1>

https://github.com/TaeAhnK/CSE3013\_Week2

https://github.com/TaeAhnK/CSE3013\_Week3

https://github.com/TaeAhnK/CSE3013\_Week4

1. **clone**

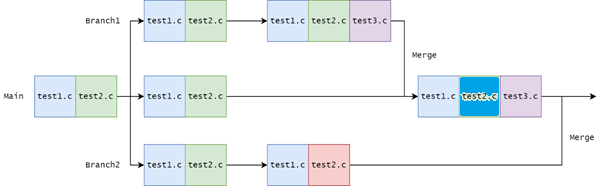


GitHub의 Commit History를 보면 커밋 내역을 볼 수 있다. 새로 push한 Week 5 push 2도 반영되어 있음을 확인할 수 있다.

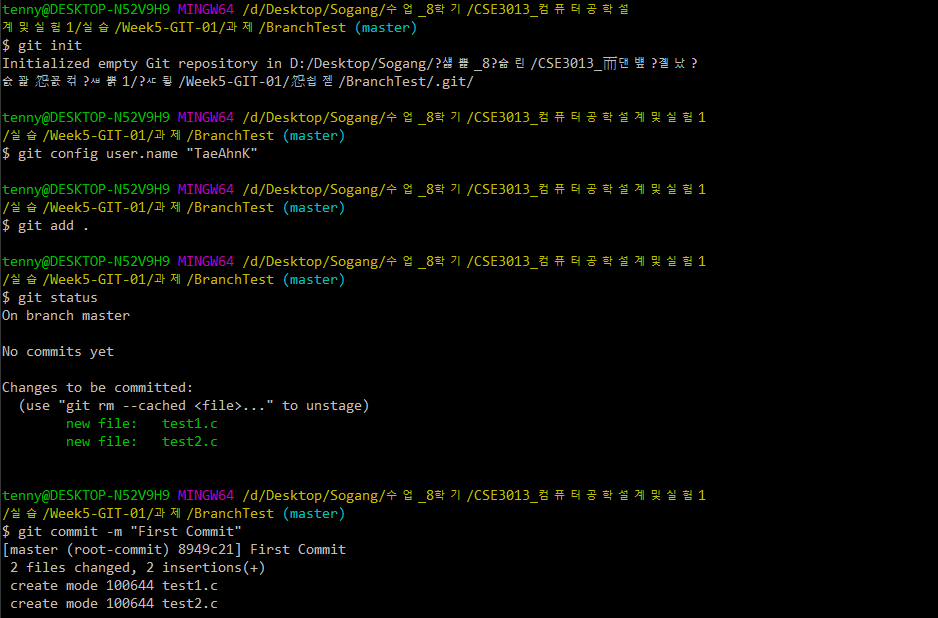
GitHub Repository 주소

https://github.com/TaeAhnK/CSE3013\_Week5

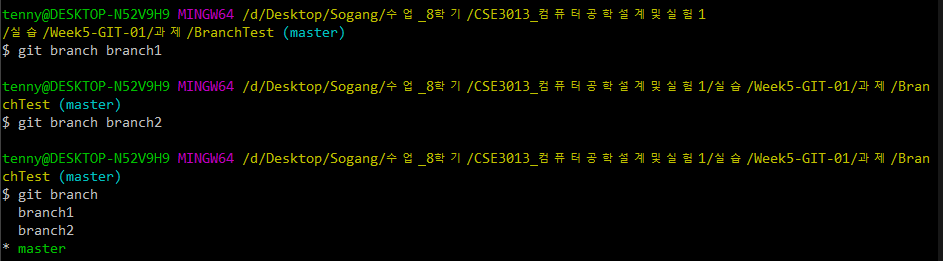
1. 과제
2. **Branch and Merge**



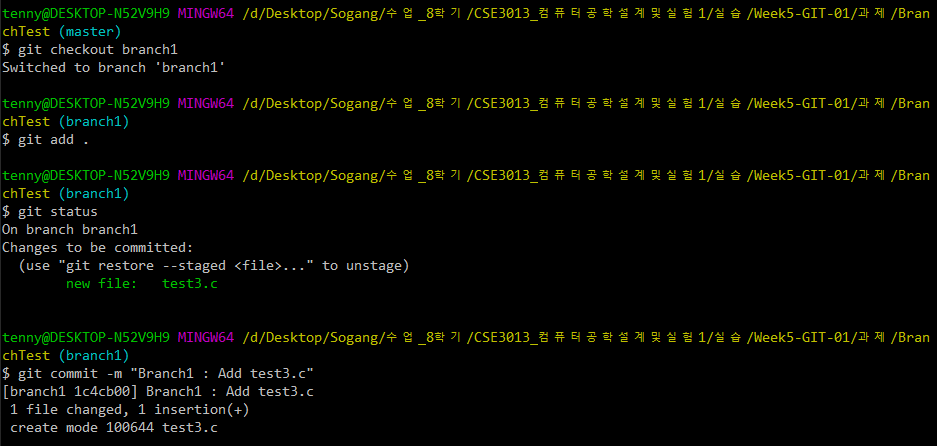
실습을 위해 다음과 같은 실험을 준비한다. 먼저 Master Branch에 test1.c와 test2.c를 commit 한다. test1.c는 내용이 1인 파일이고, test2.c는 2인 파일이다. 그리고 브렌치를 2개 생성해 총 3개의 브렌치를 만든다. 새로 생성한 두 Branch는 Branch1과 Branch2라고 한다. Branch1에서는 내용이 3인 test3.c를 commit 한다. Branch2는 test2.c의 파일을 내용을 -2로 수정한다. 그후, Master Branch와 Branch1을 Merge 한다. 그 다음, Master Branch와 Branch2를 Merge한다.



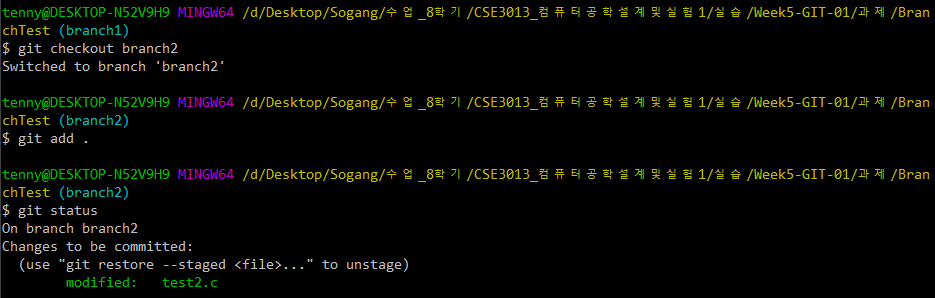
먼저 test1.c와 test2.c를 생성해 commit 하였다.



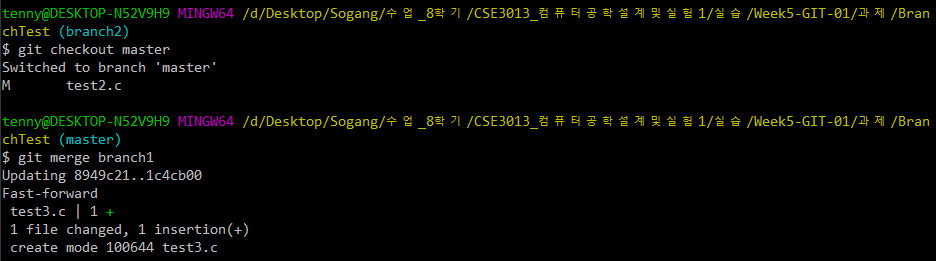
Branch1과 Branch2를 생성했다. 총 3개의 Branch가 존재한다.



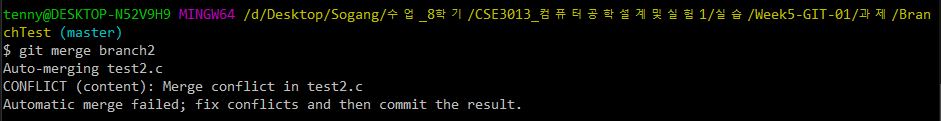
새 파일 test3.c를 생성하고, 현재 Branch를 Branch1으로 변경했다. 그리고, 변경된 파일을 커밋했다.



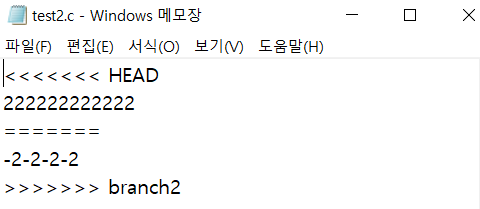
Branch2로 현재 Branch를 변경하고 test2.c의 내용을 2에서 -2-2-2로 변경했다.



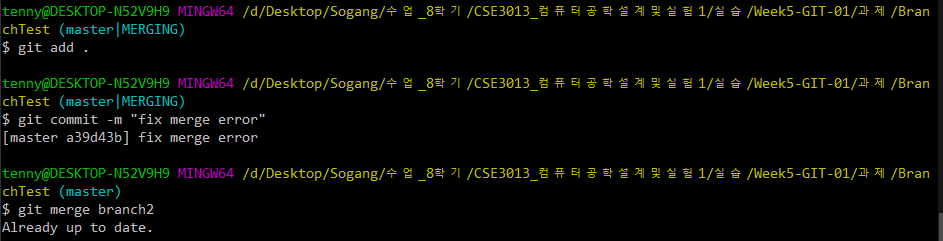
다시 Master Branch로 돌아와 Branch1과 Master Branch를 Merge했다. 그 결과, Master Branch는 test1.c, test2.c, test3.c를 가지게 되었다. 그리고, test2.c의 내용을 2222로 변경했다.



변경한 뒤, Branch2와 Master를 merge를 할 경우, Conflict가 발생하며 에러가 생긴다.

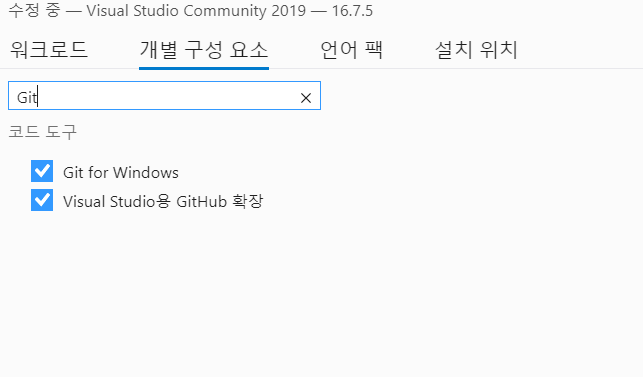


파일을 보면 위와 같이 두 파일에서 동시에 변경이 있어 충돌한다는 메시지를 볼 수 있다.



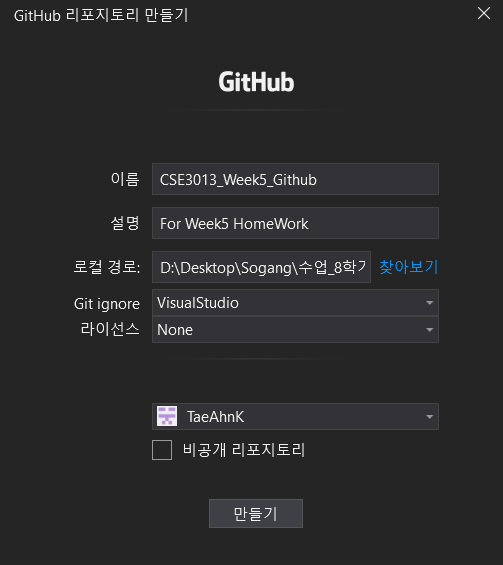
파일의 내용을 -2-2-2-2를 남기며 수정하고 커밋하면 에러가 해결된다.

1. **Visual Studio에서 GitHub 사용**

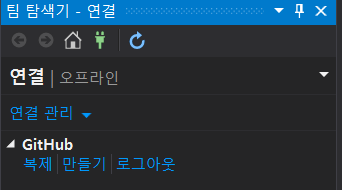


마이크로소프트의 Visual Studio도 Git과 GitHub 관련 애드온을 지원한다. 도구>도구 및 기능 가져오기>개별 구성 요소에서 Git과 GitHub 확장을 확인할 수 있다.

애드온을 설치한 후, 팀 탐색기 메뉴에서 GitHub와 VS를 연결할 수 있다.



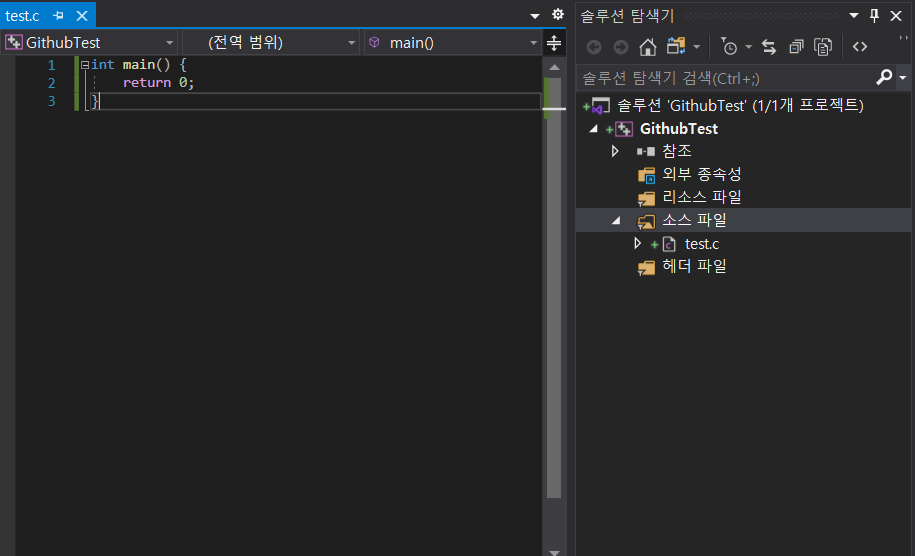
GitHub 계정을 등록해 연동한다.



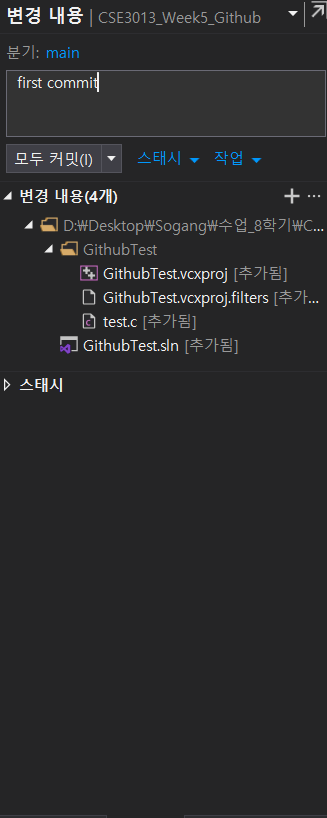
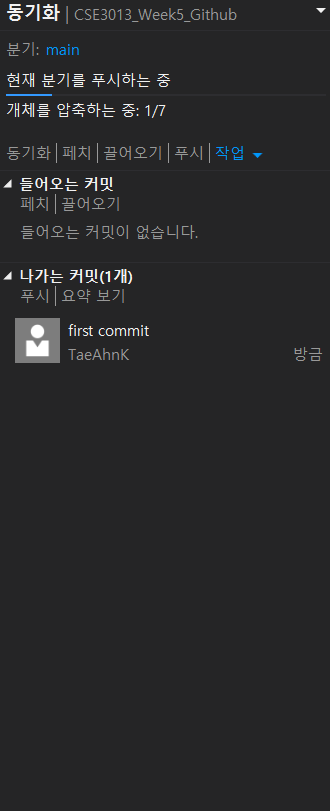
연동한 후, ‘만들기’를 통해 Remote Repository를 VS에서 생성할 수 있다.



생성하면 다음과 같은 메뉴를 볼 수 있다. 그 아래 있는 ‘솔루션>새로 만들기’를 통해 새 프로젝트를 생성한다.



위와 같이 프로젝트를 작성할 수 있다.

작성을 완료하고, 메뉴에서 ‘변경 내용’을 클릭하면 왼쪽 사진과 같은 커밋 창을 볼 수 있다. 커밋 메시지를 작성하고 커밋을 누르면 Local Repository에 커밋할 수 있다. 그리고, 동기화 메뉴에서 ‘푸시’를 클릭해 이전에 설정한 GitHub Repository로 파일을 push 할 수 있다.

GitHub Repository

<https://github.com/TaeAhnK/CSE3013_Week5_Github>

1. 결론

이번 실습에서는 버전 관리 시스템과 Git, 그리고 GitHub의 사용법을 실습했다. 버전 관리 시스템이란 작성 중인 파일을 시간대별로 저장하여 특정 시점에서의 버전을 쉽게 관리할 수 있게 하는 방법이다. 이전의 파일을 사용하고 싶거나, 변경된 사항을 확인하거나, 여러 명이 같이 협업할 경우 유용하게 사용할 수 있다.

버전 관리 시스템 중 가장 많이 쓰이는 프로그램인 Git이다. Git은 빠르고, 다른 버전 관리 시스템보다 저장공간을 효율적으로 사용한다는 장점이 있다. git init을 통해 초기 설정을 하고 git add를 통해 현재 파일을 Staging Area에 로드할 수 있다. git commit을 통해 현재 Staging Area의 상태를 Local Repository에 저장하고, git push를 통해 Local Repository를 Remote Repository에 저장할 수 있다.

GitHub는 Git을 업로드할 수 있는 온라인 Remote Repository로 협업을 할 때 매우 유리하다. 진행 중인 프로젝트를 pull 하여 여러 명이 다른 곳에서도 프로젝트를 진행할 수 있다. 또한, 코드를 push한 사람과 변경 사항을 파악할 수 있어 일일이 설명하지 않고도 변경사항을 찾아볼 수 있다. 또한, 온라인에서 파일이 저장돼 유실될 위험이 적다.