2020년도 2학기 컴퓨터공학설계및실험I

5주차 예비 보고서

20170175 김태안

1. 실습 목적

버전 관리 시스템이 무엇인지 이해하고, Git과 GitHub의 사용법을 익힌다.

1. 관련 이론

**버전 관리 시스템**

버전 관리 시스템이란 파일의 변화를 시간에 따라 기록한 뒤, 나중에 특정 시점의 버전을 다시 꺼내 올 수 있는 시스템이다. 소프트웨어의 코드나 그림의 도안을 만들다 보면 일부 파일을 이전 상태로 되돌리거나 프로젝트를 통째로 특정 시점으로 되돌리고 싶은 때가 있다. 또는, 프로젝트를 수정한 이후에 갑자기 오류가 발생하기도 한다. 버전 관리 시스템을 사용하면 파일의 변화가 기록되어 파일을 특정 시점으로 되돌리거나, 변경 내용을 추적하기에 용이하다. 또한, 다수의 사람들과 협업을 통해 프로젝트를 진행할 때도 파일 변경의 흐름과 변경한 사람을 쉽게 파악할 수 있다.

**Git**

Git은 버전 관리 시스템의 한 종류이다. 리눅스 운영체제를 만든 리누스 토발드가 오픈소스 프로그램의 버전을 쉽게 관리하기 위해 제작했다. 이름 자체는 특별한 뜻은 없으며 리누스 토발드는 "global information tracker"부터 "goddamn idiotic truckload of sh\*t"까지 기분에 따라 뜻을 붙이라고 설명한다. 빠른 속도와 단순한 구조, 비선형적 개발, 완벽한 분산, 대형 프로젝트에서도 사용 가능이라는 원칙을 바탕으로 만들었으며, 많은 커맨드와 고급 명령을 수행할 수 있다. Git 자체도 오픈소스로 이루어져 있으며, 마음대로 수정과 배포가 가능한 GNU General Public License 프로그램이다.[1], [2]

**Git의 장단점**

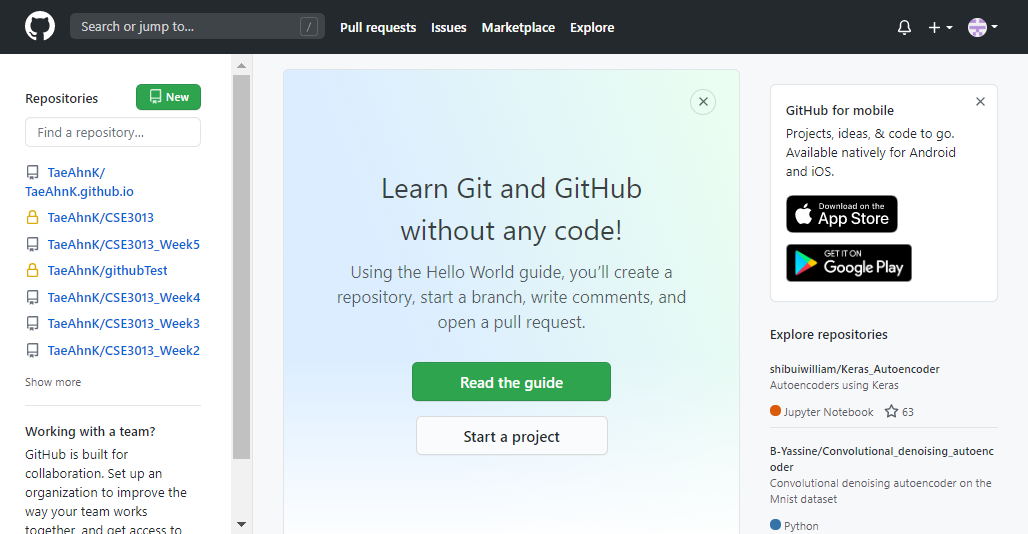
Git 이외에도 CVS, Subversion, Perforce 등의 버전 관리 시스템이 있다. 그러나 전체 버전 관리 시스템의 70%를 Git이 차지할 정도로 Git에 대한 선호가 높다. [3]이는 Git만이 가지는 독특한 특징들 때문이다.

CVS, Subversion, Perforce 등의 버전 관리 시스템은 델타 기반 버전 관리 시스템 방식을 사용한다. 특정 시점을 저장할 때, 모든 파일을 다시 저장한다. 이 경우, 시간이 지날수록 버전 관리 시스템의 용량이 매우 커진다. 그에 반해 Git은 데이터를 스냅샷으로 저장해 스냅샷의 스트림으로 취급한다. 파일이 바뀌지 않았으면 새로 파일을 저장하지 않고, 이전 상태의 파일의 링크만 저장한다. 또한, Git은 대부분의 명령을 로컬에서 처리한다. 네트워크에 연결하는 다른 프로그램과 달리 로컬 데이터베이스에서 히스토리를 읽기 때문에 처리가 매우 빠르며, 오프라인 상태에서 매우 편리하다. 그러나 앞에서 소개했듯, 많은 명령어를 사용하는 만큼 익히기 어렵다는 단점이 있다.[2]

1. 실습 방법

**GitHub 가입**

GitHub는 Git을 온라인에서도 사용해 다른 개발자와 같이 프로젝트를 진행하도록 도와주는 웹사이트이다. GitHub 홈페이지 (<https://github.com/>)로 접속해 사용자명, 이메일, 패스워드를 입력한 뒤, 이메일 인증을 하면 간단하게 회원가입을 할 수 있다.

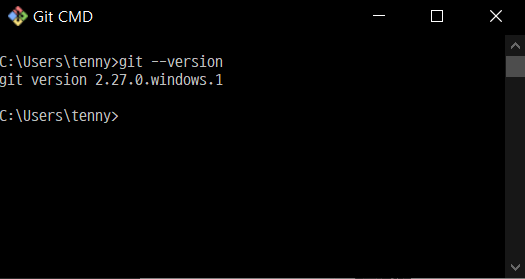


<https://github.com/TaeAhnK?tab=repositories>

**Git 설치**

Git은 Git 홈페이지 (<https://git-scm.com/downloads>)에서 다운받을 수 있다. 운영체제에 맞는 설치파일을 다운받아 실행한다. 설치 옵션과 관련해 주의할 점은 Line Ending 설정이다. Line Ending은 텍스트를 입력하고 줄바꿈을 하는 방법을 의미한다. 윈도우와 유닉스는 서로 다른 Line Ending을 가지고 있기 때문에 주의하지 않으면 환경이 바뀌었을 때 오류가 발생한다. 줄바꿈에는 Carriage Return (CR) 방식과 Line Feed (LF) 방식이 있다. 이는 타자기에서 유래되었는데, CR은 커서를 맨 앞으로 옮기는 것을 말하고, LF는 종이를 한 라인 위로 옮기는 것을 말한다. 윈도우는 CRLF 방식을, 유닉스는 LF 방식을 사용한다. 사용자가 텍스트 에디터에서 텍스트를 입력할 때는, 줄을 바꾸면 커서가 당연히 제일 앞에 있기 때문에 차이를 구별하지는 못하지만, 실제로 컴퓨터에 입력될 때는 윈도우는 \r\n이, 유닉스는 \n이 입력된다. 이러한 호환 문제로 Git을 설치할 때 Line Ending 방식을 지정하는 옵션이 존재한다.[4]

설치 완료 후 –-version 명령어를 통해 버전을 확인하면 다음과 같은 문구가 출력된다.

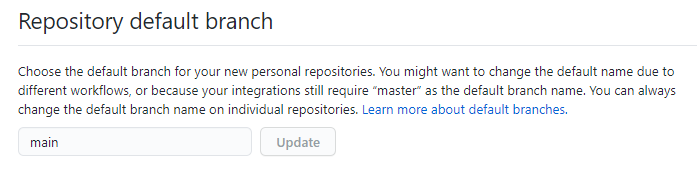


1. 기타

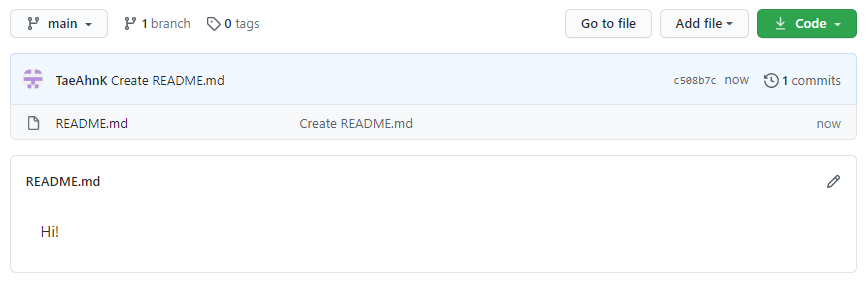
**‘master’에서 ‘main’으로**

2020년 5월 26일, 미국에서 조지 플로이드라는 흑인 남성이 경찰 체포 도중 숨지는 사건이 발생하였다. 이에 미국 전역에서 #Black Lives Matter라는 구호화 함께 사회 운동이 다시 전개되었다. 그 과정에서 여러 IT 영역에서 사용되는 Master-Slave, Blacklist-Whitelist와 같은 용어에 대한 논의가 진행되었다. 이에 GitHub의 CEO 냇 프리드만을 포함한 많은 IT기업들은 긍정적으로 반응하며 GitHub는 10월부터 새로 생성되는 레포지토리에 대해 master 대신 main을 사용하도록 선택권을 주는 업데이트를 진행했다. 더 나아가, 연말까지 기존의 레포지토리에서도 master를 main으로 변경할 수 있도록 업데이트를 진행할 것이라고 밝혔다. [5]

Github의 레포지토리 설정 페이지에 가면 다음과 같이 default branch의 이름을 설정할 수 있는 옵션이 추가되었다.



버튼의 업데이트를 누르고 새 레포지토리를 생성하면 아래와 같이 main을 사용한 branch가 적용되는 것을 확인할 수 있다.



1. 참고 문헌
2. Hamano, Junio C. “Git/Git.” *GitHub*, 2020, github.com/git/git.
3. Chacon, Scott, and Ben Straub. “Pro Git.” *git*, Apress, 2014, git-scm.com/book/ko/v2.
4. Stack Overflow. “Stack Overflow Developer Survey 2017.” *Stack Overflow*, Stack Overflow, 2017, insights.stackoverflow.com/survey/2017.
5. Krueger, Richard. “DOS vs. Unix Line Endings.” *Windows vs. Unix Line Endings*, Computer Science University of Toronto, 2006, www.cs.toronto.edu/~krueger/csc209h/tut/line-endings.html.
6. BBC, “GitHub Abandons 'Master' Term to Avoid Slavery Row.” *BBC*, 15 June 2020, www.bbc.com/news/technology-53050955.
7. Mullans, Allex. “Renaming the Default Branch from Master.” *GitHub*, 2020, github.com/github/renaming.