

1 Refer to : <https://backlog.com/git-tutorial/kr/>
2 1. Git History
3 1)Linux Kernel project를 위한 version 관리 system으로 개발
4 2)Birth year : 2005년
5 3)Linus Benedict Torvalds
6 4)A Short History of Git
7 -<https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>
8
9 2. Git과 SVN의 trend 비교
10 1)<https://trends.google.com/trends/explore?geo=US&q=git,svn>
11
12 3. Version관리 System이 필요한 이유
13 1)file의 version관리를 어떻게 하는가?
14 2)가장 간단한 방법은 파일을 미리 복사해두는 것이다.
15 3)예를 들면, file과 folder명 뒤에 편집한 날짜를 붙여주는 방식같은 것이다.
16 4)하지만 file을 편집할 때마다 매번 복사하는 일은 번거롭기도 하고 실수할 가능성도 많다.
17 5)또한 특별한 규칙 없이 마음대로 이름을 붙여놓는 경우 어느 file이 최신인지, 또 file의 어떤 부분이 변경된 것인지 파악하기 어렵다.
18 6)바로 이런 문제를 해결하기 위해 만들어진 것이 version 관리 system이다.
19
20 4. Git?
21 1)Source code를 보다 효과적으로 효율적으로 관리하기 위해 개발된 분산형 version 관리 system이다.
22 2)Git에서는 source code가 변경된 이력을 쉽게 확인할 수 있다.
23 3)특정 시점에 저장된 version과 비교하거나 특정 시점으로 되돌아갈 수도 있다.
24 4)또 내가 올리려는 file이 누군가 편집한 내용과 충돌한다면, server에 upload할 때 경고 메시지가 발생한다.
25
26 5. 초기 목표
27 1)속도(Network 및 File 처리)
28 2)동시 다발적인 개발
29 3)책임성
30 4)대형 project를 효율적으로 지원
31
32 6. 저장소
33 1)저장소(Git repository)란 file이나 folder를 저장해 두는 곳.
34 2)Git 저장소는 file이 변경 이력 별로 구분되어 저장된다.
35 3)비슷한 file이라도 실제 내용 일부 문구가 서로 다르다면 다른 file로 인식하기 때문에 file을 변경 사항 별로 구분해 저장할 수 있다.
36 4)Git은 원격 저장소와 로컬 저장소 두 종류의 저장소를 제공한다.
37 5)원격 저장소(Remote Repository)는 file이 원격 저장소 전용 server에서 관리되며 여러 사람이 함께 공유하기 위한 저장소를 말한다.
38 6)반면 로컬 저장소(Local Repository)는 내 PC에 file이 저장되는 개인 전용 저장소이다.
39 7)평소에는 내 PC의 local 저장소에서 작업하다가 작업한 내용을 공개하고 싶을 때에 원격 저장소에 upload를 하면 된다.
40 8)반대로 원격 저장소에서 다른 사람이 작업한 file을 local 저장소로 가져올 수도 있다.
41 9)저장소를 만드는 방법은
42 -저장소를 새로 생성한다.
43 -이미 만들어져 있는 원격 저장소를 local 저장소로 복사해서 사용한다.
44
45 7. 동작원리
46 1)Snapshot
47 -data를 가져오거나 project를 저장할 때마다 그 시점의 file에 대해 snapshot을 저장한다.
48 -변경되지 않은 file은 다시 file을 저장하지 않고 이전에 지정한 동일한 file을 link한다.
49 2)Checksum
50 -data를 저장하기 전에 checksum을 구하고 이 checksum을 통해 data를 관리한다.
51 -SHA-1 hash 사용
52 --16진수 문자 40개로 구성된 문자열
53 --file의 내용 또는 directory 구조를 기반으로 계산
54 -file 이름이 아닌 contents의 hash 값을 저장한다.
55 --file의 이름이 변경되더라도 내용이 동일하면 같은 hash를 갖는다.
56 3)Sections(Work Tree)
57 -Working Directory
58 -Repository의 project를 checkout하거나 수동으로 생성
59 -Stating Area(Index)
60 -Repository에 commit하기 위한 중간 저장소
61 -Repository
62 -Commit은 working directory에 있는 변경 내용을 저장소에 upload하는 과정이다.
63 -그런데, 바로 upload하는 것이 아니라 중간 단계인 staging area(index)에 file 상태를 기록해야 한다.
64 -따라서 저장소에 변경 사항을 기록하기 위해서는, 기록하고자 하는 모든 변경 사항들이 staging area(index)에

존재해야 한다.