1. VCS 이란 무엇인가?

-버전 관리 시스템이란 파일 변화를 시간에 따라 기록했다가 나중에 특정 시점의 버전을  
 다시 불러올 수 있는 시스템을 의미합니다.

1. GIT 이 SVN 과 같은 중앙집중식 VCS와 가장 큰 차이는 무엇인가?

-SVN은 중앙 서버에 저장소(repository)를 중심으로 하며, 사용자들은 중앙 서버에서 파일을 가져오고 변경사항을 반영합니다. 이는 네트워크 연결이 필요하며, 중앙 서버에 장애가 발생하면 협업이 중단될 수 있습니다. 반면에 Git은 분산 시스템으로, 모든 개발자들은 로컬에 전체 저장소의 사본을 가지고 작업합니다. 이는 네트워크 연결이 없어도 작업이 가능하며, 중앙 서버에 장애가 발생해도 로컬 작업을 계속할 수 있습니다.

1. 파일의 변경 사항을 저장소에 반영할 때 git add와 git add -p의 차이는 무엇인가?

-git add 명령어는 모든 변경된 파일을 스테이징 영역에 추가하는 반면

git add -p 명령어는 변경된 파일을 한번에 스테이징하지 않고, 변경 사항을 패치 단위

로 스테이징 할 수 있게 해주는데 이는 파일의 일부 변경 사항만을 스테이징하고

다른 변경 사항은 스테이징하지 않을 수 있도록 합니다.

1. 다음 페이지의 Git lifecycle을 참고해서 다음 케이스들에서 file1이 어떤 state 인지 기술하라.
2. 저장소에 없는 file1 이라는 파일을 메모장으로 만든 직후

-Untracked 상태로 파일이 저장소에 아직 추가되지 않았으므로

Git은 이 파일을 추적하지 않는다.

1. git add와 git commit 으로 file1을 저장소에 추가한 직후

-Committed 상태로 파일이 커밋되었으므로 저장소에 안전하게 기록되어있다.

1. file1의 내용을 수정한 직후

-Modified 상태로 파일이 수정되었지만 아직 스테이징 영역에 추가되지 않았으므로 수정된 상태이다.

1. 수정한 file1을 git add 또는 git add -p한 직후

-Staged 상태로 파일이 스테이징 영역에 추가되었으므로, 다음 커밋에 포함될

준비가 된 상태이다.

1. git commit 명령을 한 직후

-Committed 상태로 파일이 커밋되었으므로 저장소에 안전하게

기록되어있습니다. 이제 파일의 상태는 저장소의 최신 커밋과 일치합니다.

1. Step#6에서 메모장으로 확인한 file1의 내용은 무엇인가?

-세 번째 줄이였던 “Great weather” 을 제외한 “Hello world” 와

“What a wonderful world” 내용만 첫번째 둘째줄에 남아있다.

1. Step #7에서 file2는 존재하는가? 그 내용은 무엇인가?

만일 존재하지 않는다면 왜 존재하지 않는지 기술하라.

-file2는 존재하지 않았습니다.

file2가 이전에 “main” 브랜치에서 생성되고, 해당 브랜치로 병합되지 않았을 경우,

branch2로 전환하더라도 file2는 존재하지 않을 것이기 때문이다.

1. git show-branch 명령은 변경 사항에 대한 commit graph 를 보여준다

step#1 ~ #7 까지 작업한 뒤의 commit graph 는 어떻게 되는가?

스크린샷, 텍스트, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

1. 다른 branch의 내용을 반영하는 방식에는 git merge 와 git rebase 가 있다.

두 방식의 차이에 대해서 조사해서 설명해라.

-Merge의 경우 다른 branch에서 여태까지 commit한 내용을 하나의 merge commit으로 합치는 것이다. 브랜치가 너무 많아지면 히스토리를 보기 어렵다

-Rebase의 경우 base의 위치를 변경해서 다른 branch에서 commit한 내역을 최신으로 보고 그대로 끌어오는식으로 합치는 것이다. 중복 로그를 남기지 않고, merge log를 줄여

히스토리를 깔끔하게 정리할 수 있어 Rebase는 주로 히스토리를 깔끔하게 유지하기 위해 사용한다.

1. git cherry-pick 이 어떤 일을 하는 명령인지 설명하라.

: cherry-pick은 한 곳에서 특정 변경 사항을 선택하고 다른 모든 것을 가져오지 않고

다른 곳에 추가할 수 있는 마법의 도구와 같다.

ex) Git A Branch 에서 작업중인데 B Branch에서 필요한 작업이 필요할 경우 사용하면 된다. 체리 픽이 없었다면 A 브랜치와 B브랜치를 통합하고 작업해야하지만 ,체리픽을 이용하여 필요한 Commit만을 사용하고 나머지 B 브랜치의 모든 것을 가져오지 않아도 되는 것이다.

1. git rebase -i가 어떤 일을 하는 명령인지 설명하라. 직전 두 커밋의 순서를 바꾸고자

할 때 git rebase -I 를 써서 해결할 수 있다. 그 방법을 설명하라.

-interactive 옵션은 대화형으로 실행하는 옵션인데 시각적으로 커밋 로그들과

옵션을 나열해주기 때문에 실수하지 않고 수정할 수 있도록 도와준다.

커밋의 순서변경, 커밋의 편집, 커밋의 삭제 등의 기능들이 있다.

1. git 저장소를 zip 으로 묶어서 제출할 예정이다. 담당 교수가 이를 다운로드해서 압축을 푼 뒤 git log 를 했을 때의 출력 결과는 학생의 원본 저장소에서 git log 한 것과 어떻게 다른가? (다르지 않다면 다르지 않다고 기술할 것) 이런 결과가 나오는 이유는 무엇인가?

-제출한 zip파일은 원본 git 저장소의 스냅샷이므로 git log 결과는 동일합니다.

Git은 변경 이력을 커밋으로 기록하며, 이러한 커밋들은 스냅샷을 통해 저장되고

따라서 같은 스냅샷을 가진 저장소라면 git log 결과가 같아야한다고 생각합니다.