소프트웨어공학 Homework #2

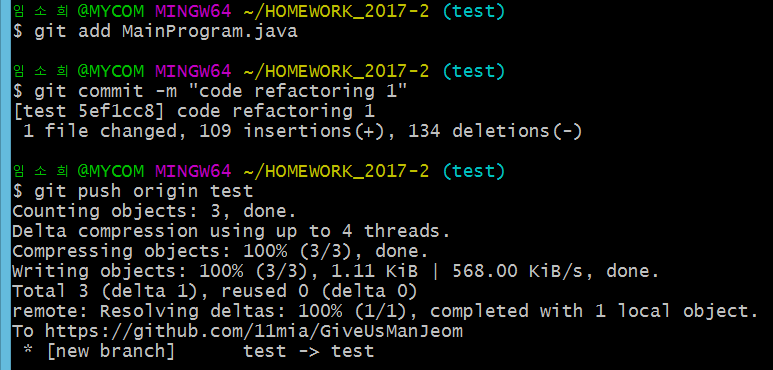
1. Version Control System

<team’s GitHub Remote URL>

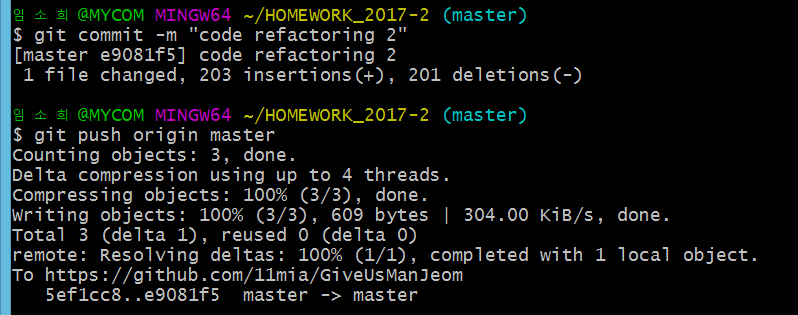
-<https://github.com/11mia/GiveUsManJeom.git>

<4> 임소희 (1515655)

① git commit

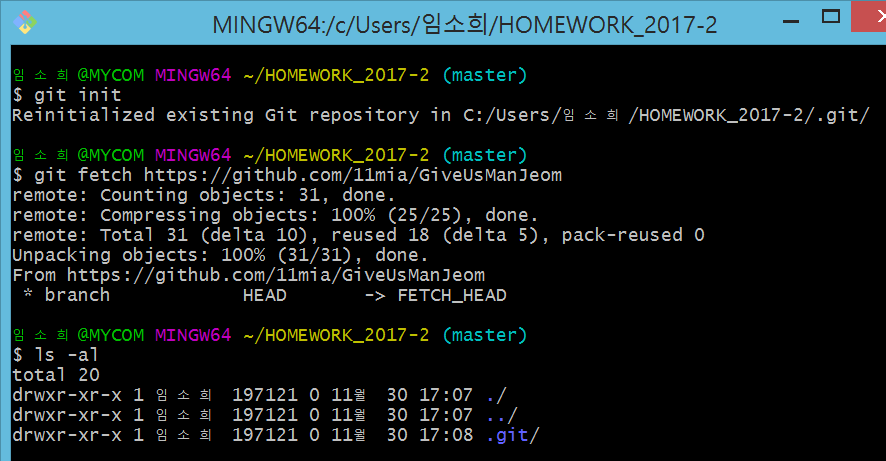


제가 로컬에서 한 작업을 서버에 올리려면 add🡪commit🡪push 과정을 거쳐야 합니다. 수정한 MainProgram.java를 add로 staging area에 올리고, commit 명령을 통해 수정한 사항을 추가한 후 code refactoring 메시지를 남깁니다. 마지막으로 push를 하면 서버에 올라가게 됩니다. commit 시 남긴 메시지는 서버에 표시됩니다. 여기서 add와 commit의 차이는 다음과 같습니다. Add는 staging area에 수정된 파일을 올리는 것이고, commit은 staging area에 올라온 파일 전체가 아닌, 수정된 부분만 기록합니다.

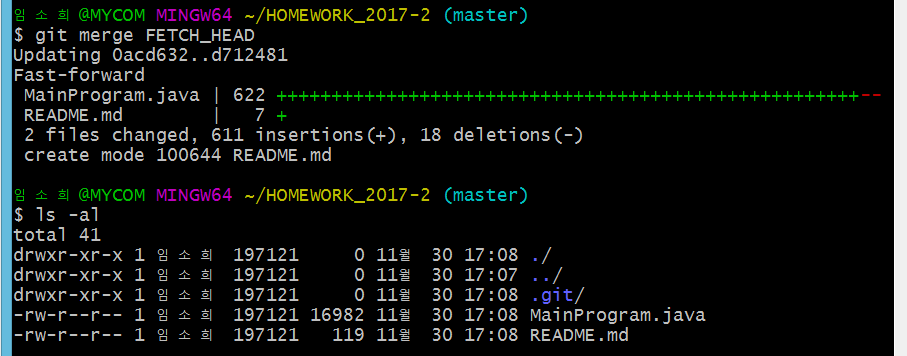


commit 시, 메시지를 남기면 위의 code refactoring과 구분이 됩니다.

② git fetch

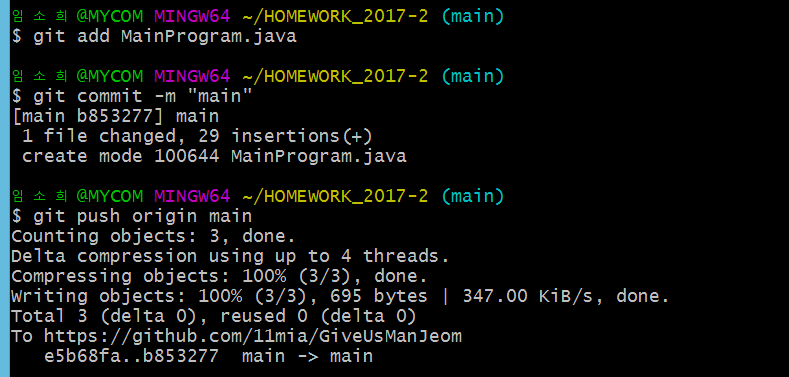


서버로부터 작업을 받을 때 fetch를 사용합니다. 이경우는 처음으로 나의 로컬 디렉터리에 작업을 받는 것입니다. fetch만 해서는 로컬에 서버작업들을 확인할 수 없습니다. ls명령어로 디렉터리의 파일을 확인해보면 서버작업이 오지 않았음을 확인할 수 있습니다.

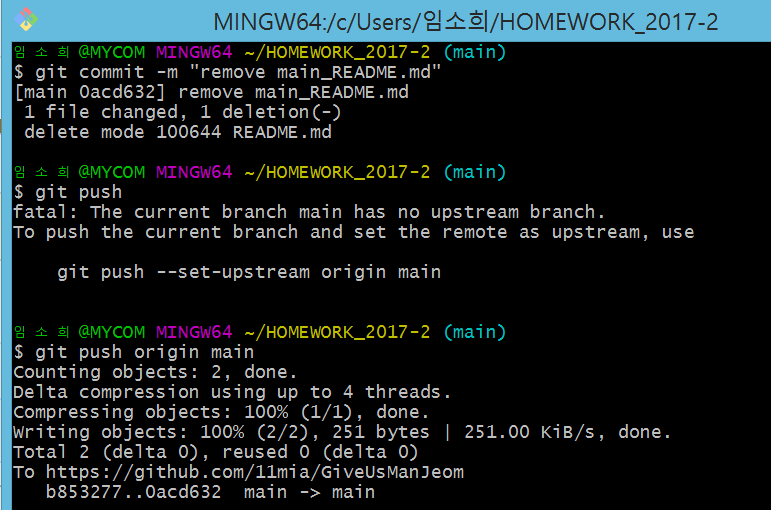


merge를 해야만 로컬 디렉터리에 서버작업이 들어옵니다. ls 명령어로 확인해보면 서버작업이 들어왔음을 확인 할 수 있습니다.

③ git push

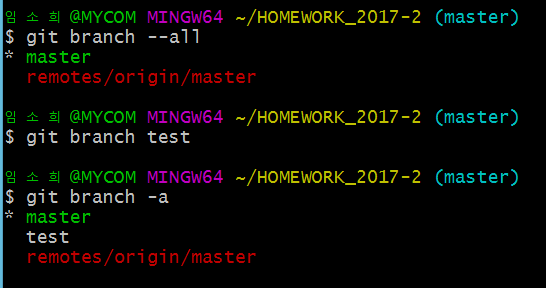


add->commit->push 명령을 수행하면 서버에 작업한 파일이 올라갑니다. push를 하지 않으면 파일은 staging area에만 올라가만 있고, 실제 서버에 반영되지 않습니다.



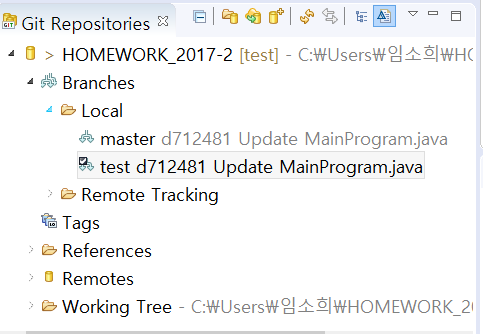
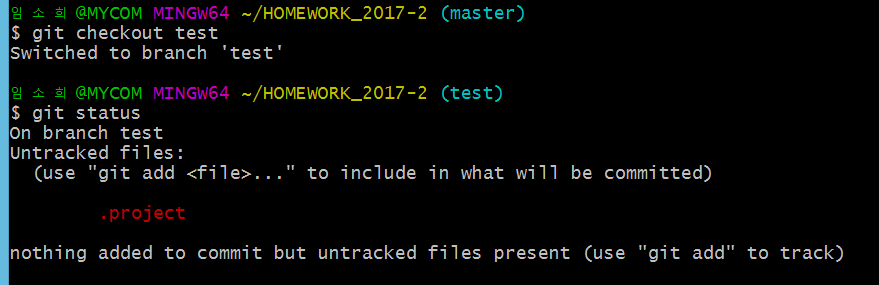
서버에 파일을 올리는 것 외에 삭제도 가능합니다. git rm README.md 명령어 수행 후 이를 커밋 후 push하면 서버에 삭제된 것이 반영됩니다. 단, 이전 push 때 ‘git push -u origin master’을 사용하지 않는 이상(u옵션 사용) git push로만 적으면 오류가 뜹니다. Push 명령어 사용시 올릴 브랜치 명을 써서 그 브랜치에 수정사항이 업데이트 되도록 하였습니다.

④ git branch



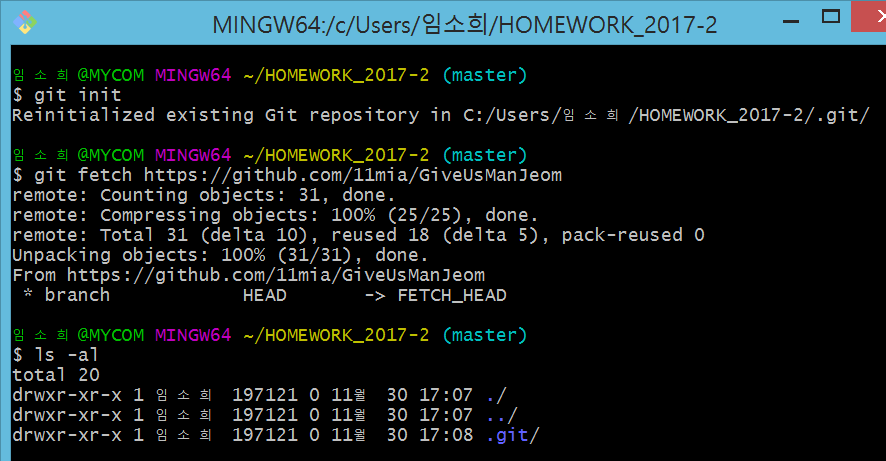
branch 명령어 옵션(--all, -a)을 사용하면 모든 브랜치 목록을 확인할 수 있습니다. ‘git branch 브랜치명’을 수행하면 새로운 브랜치를 생성하게 됩니다.

⑤ git checkout

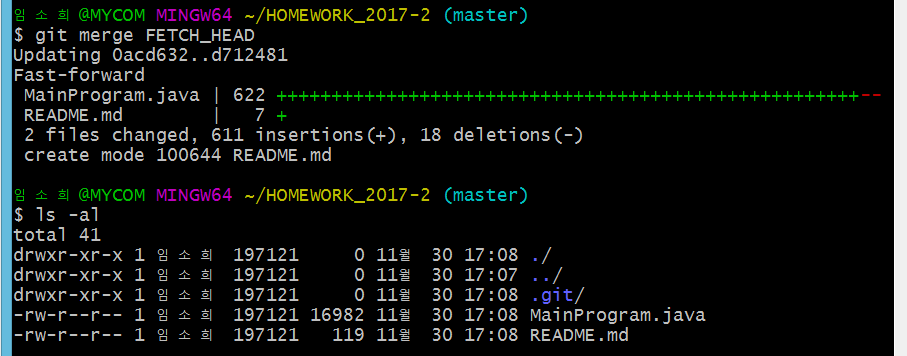


git checkout으로 브랜치를 이동할 수 있습니다. master->test로 브랜치가 이동되었음을 확인할 수 있습니다.

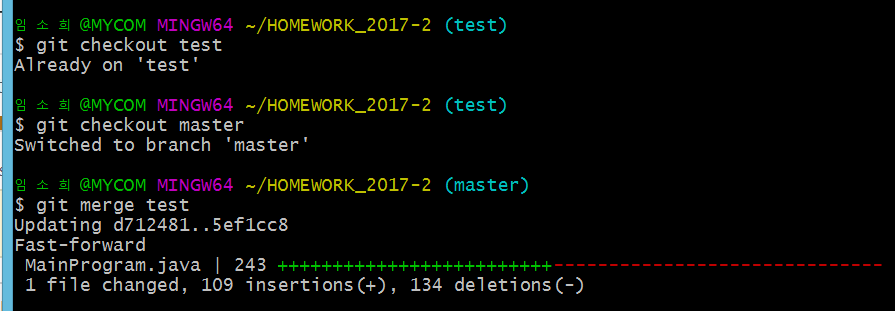
⑥ git merge



merge 전 로컬 디렉터리 내 파일 목록은 다음과 같습니다.



merge를 이용하여 로컬 디렉터리에 서버에서 받은 내용을 업데이트 할 수 있습니다. Fetch 후 FETCH-HEAD를 mater쪽으로 merge하면 서버의 최신내용으로 로컬에 업데이트됩니다. ls명령어로 보면 파일 2개가 들어왔음을 확인할 수 있습니다.



merge를 통해 브랜치를 하나로 합칠 수 있습니다. 예를 들어 a브랜치를 b로 합치고 싶을 때, 현재 작업하는 브랜치가 b인 상태에서 ‘git merge a’를 합니다. 캡쳐에서 test 브랜치를 master로 합치려고 할 때, 작업하는 브랜치를 옮기기 위해 checkout을 합니다. 현재 작업 브랜치가 master가 되고 ‘git merge test’를 수행하면 test브랜치가 master쪽으로 합병됩니다. 이 때 test브랜치에서 작업한 내용이 master쪽으로 업데이트 됩니다.