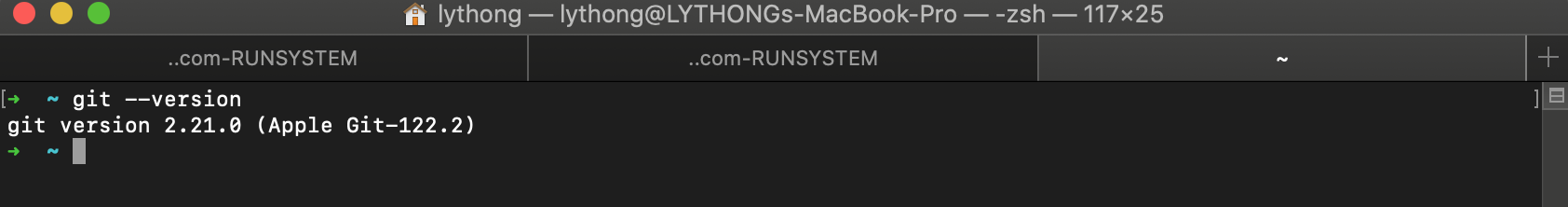
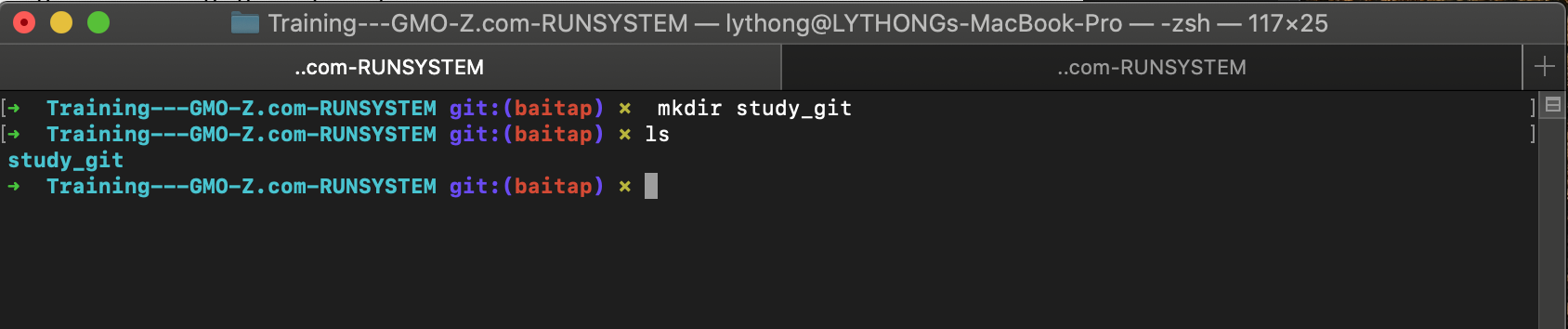
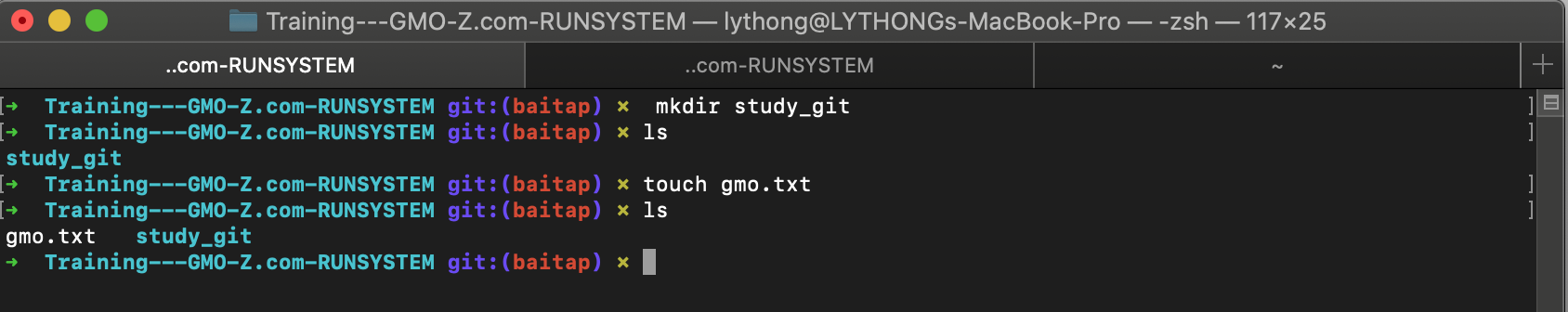
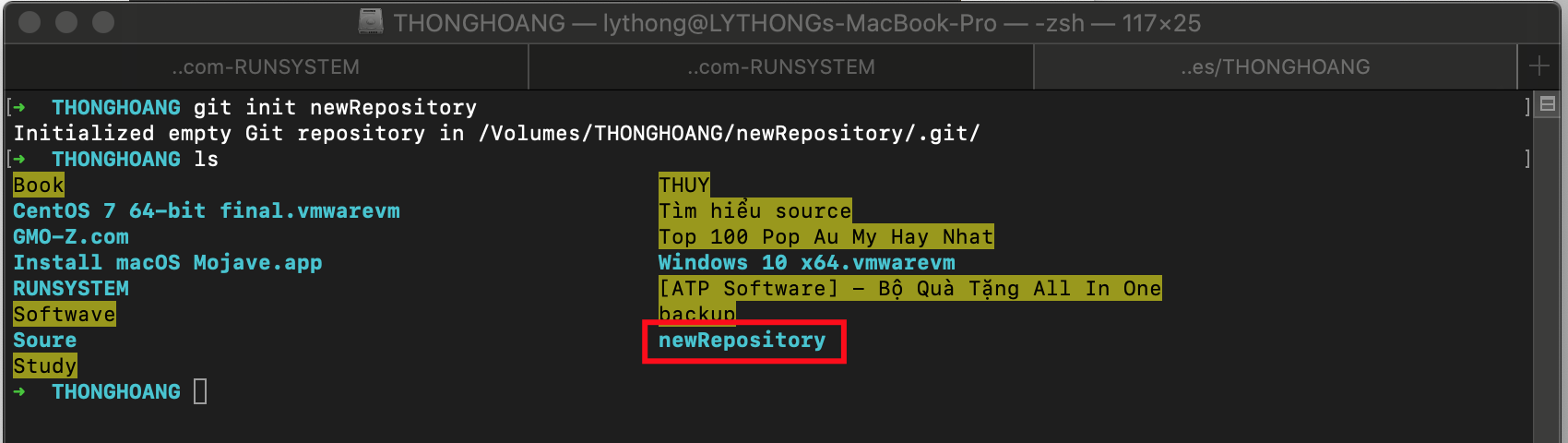
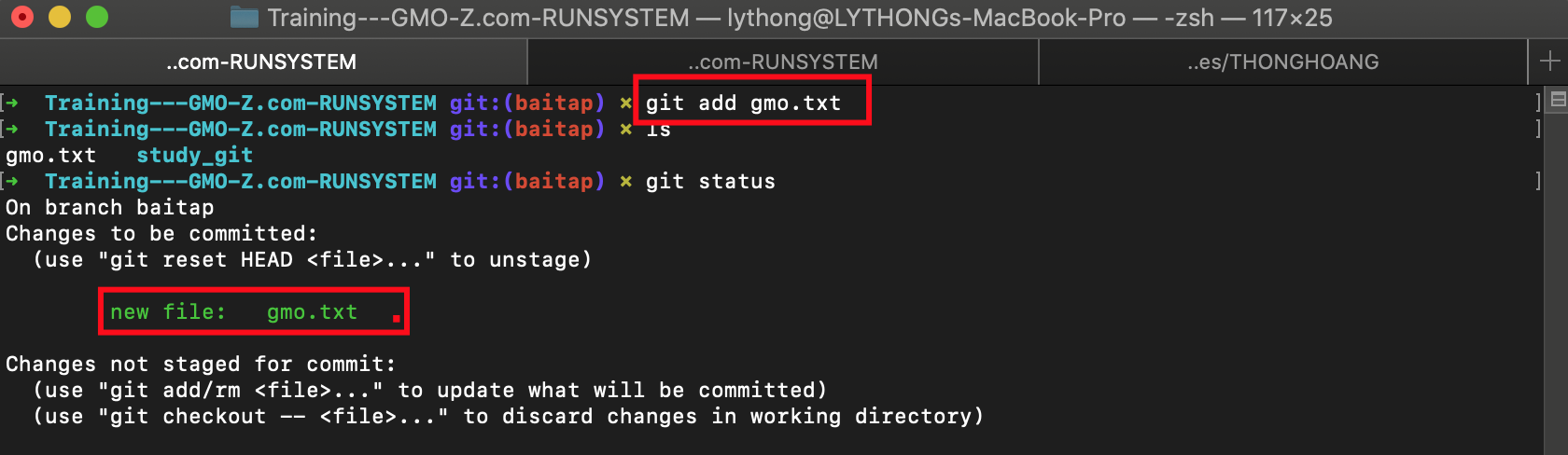
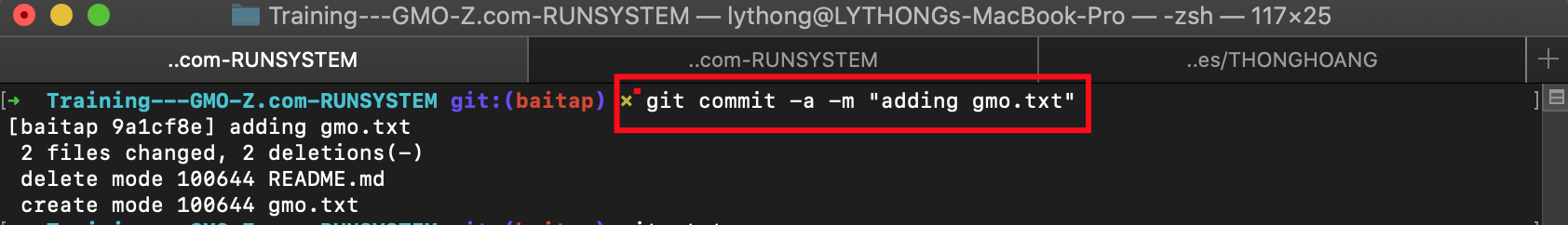
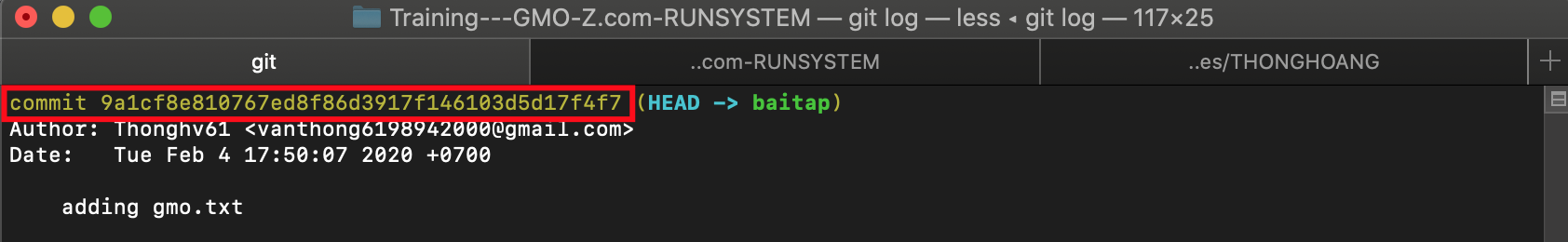
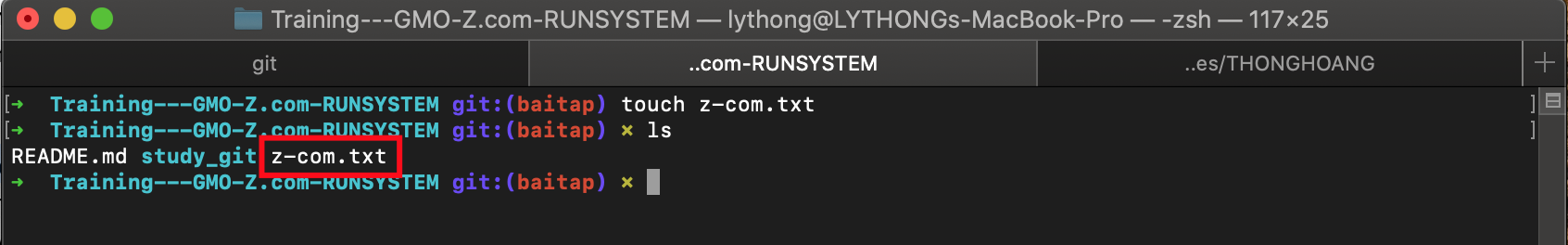
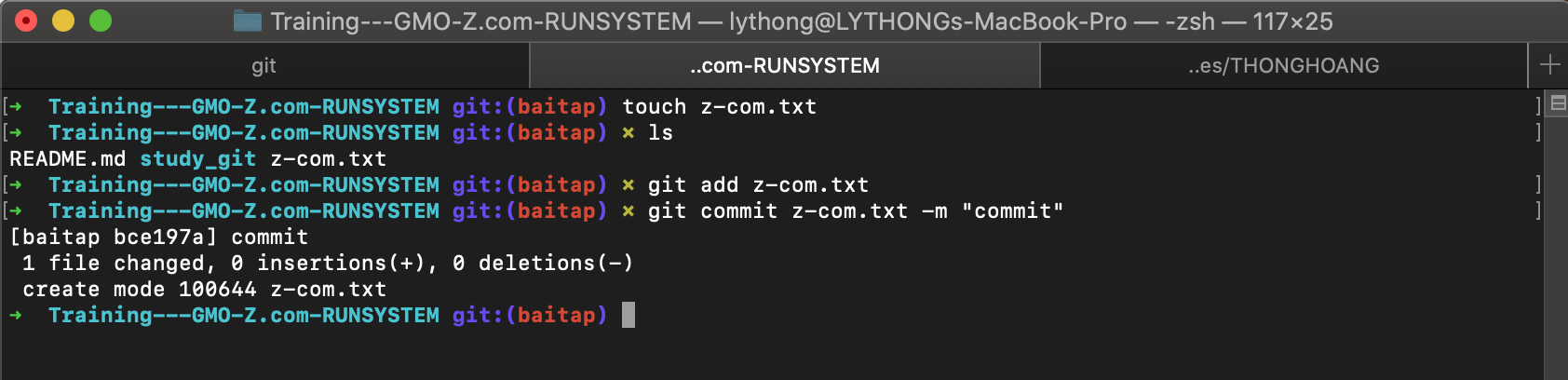
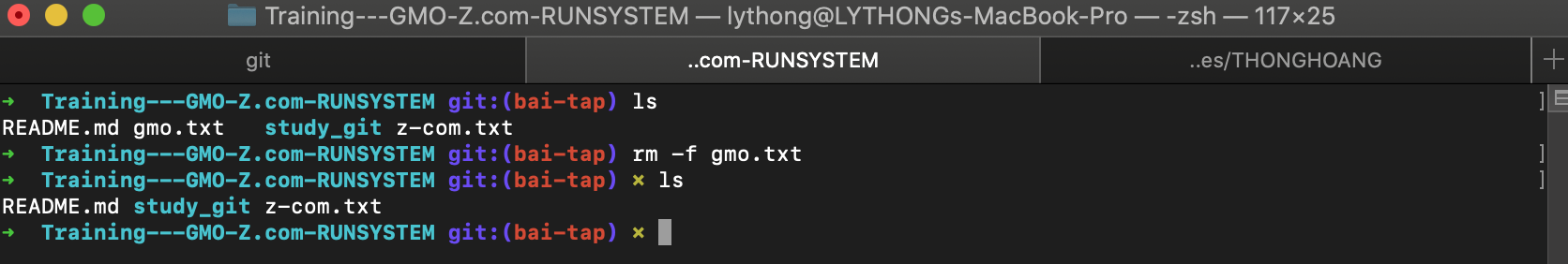
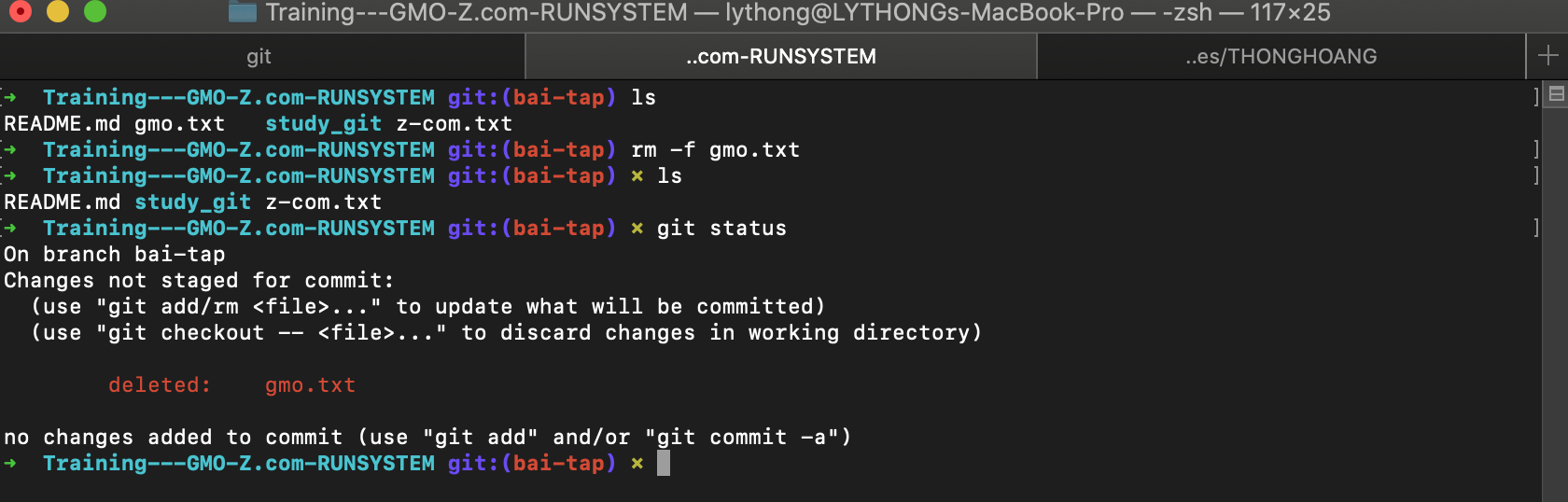
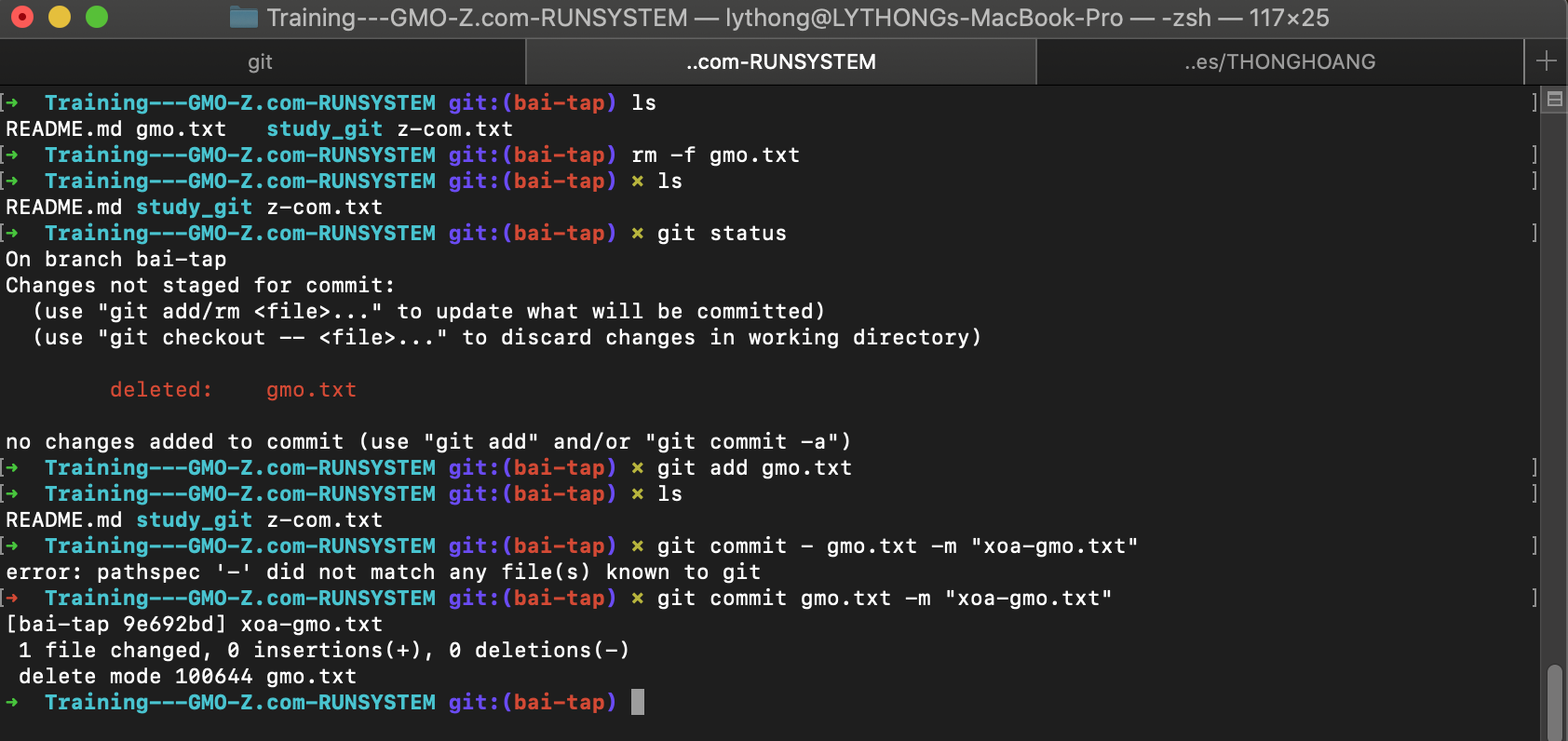
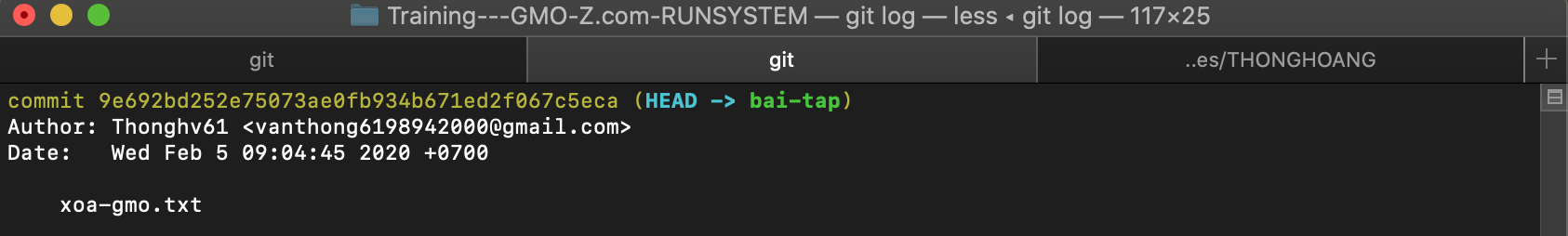
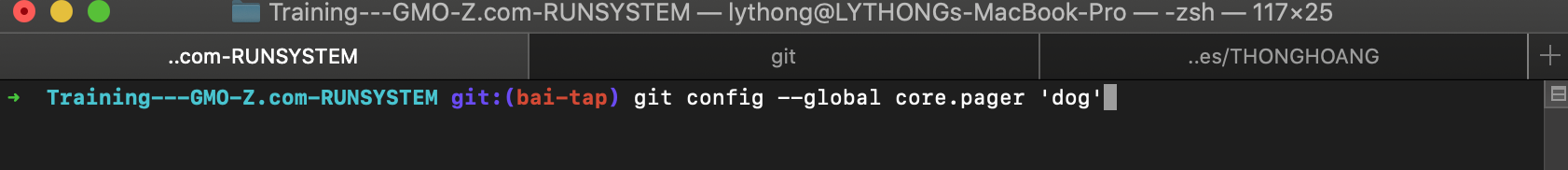
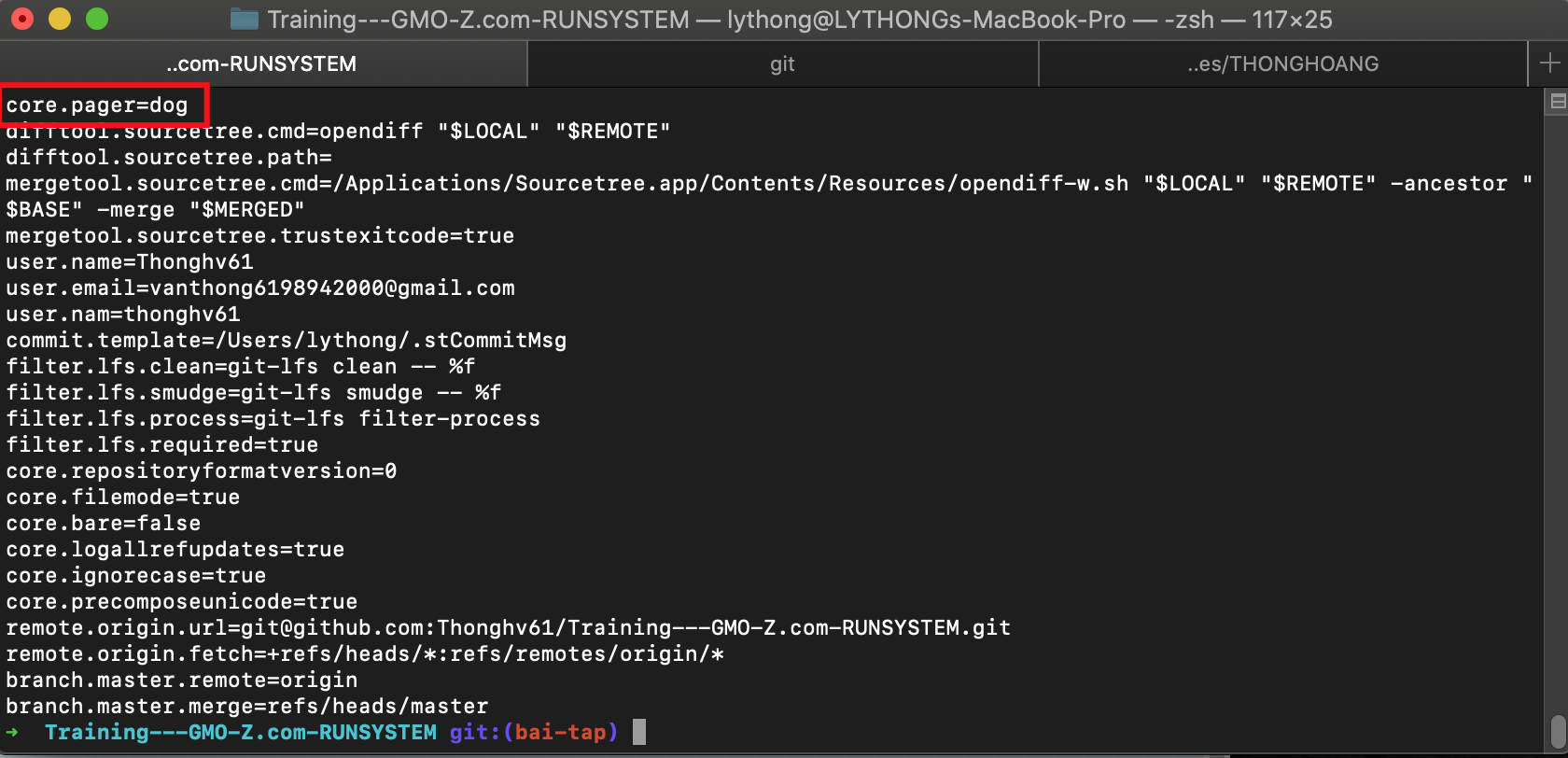
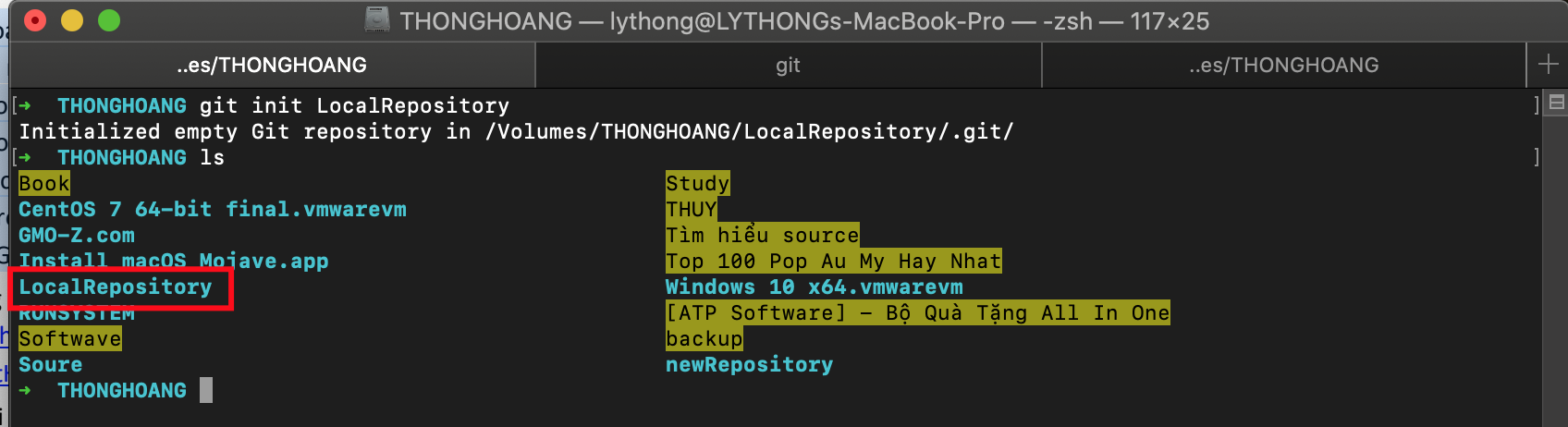
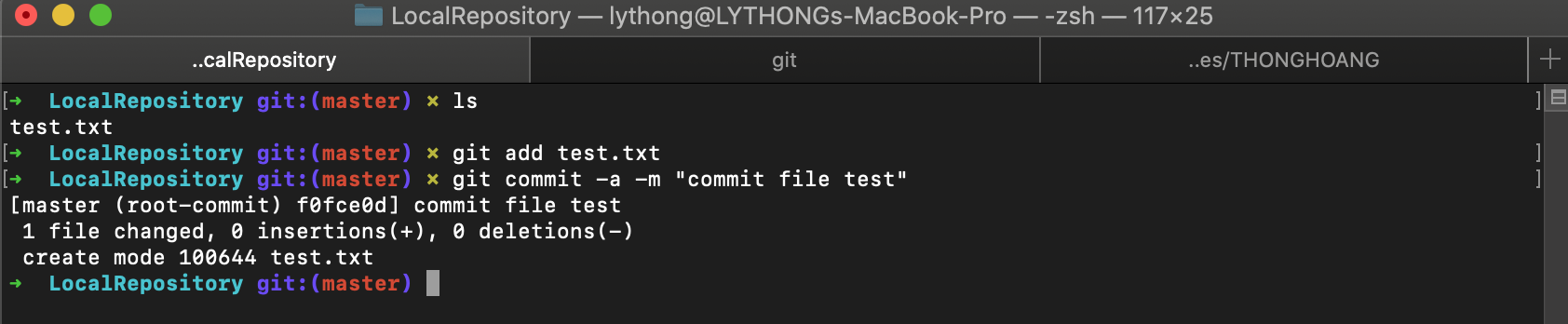
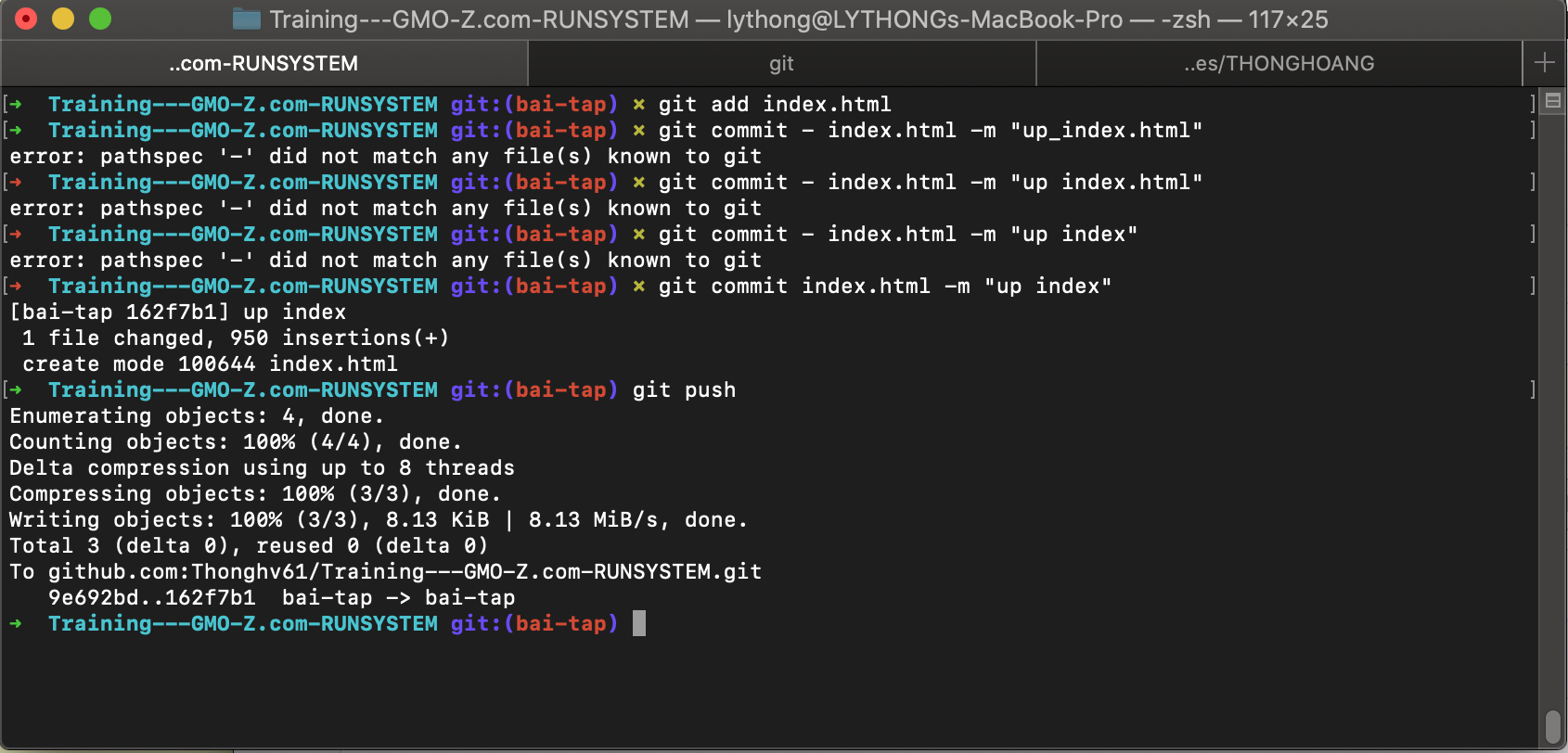
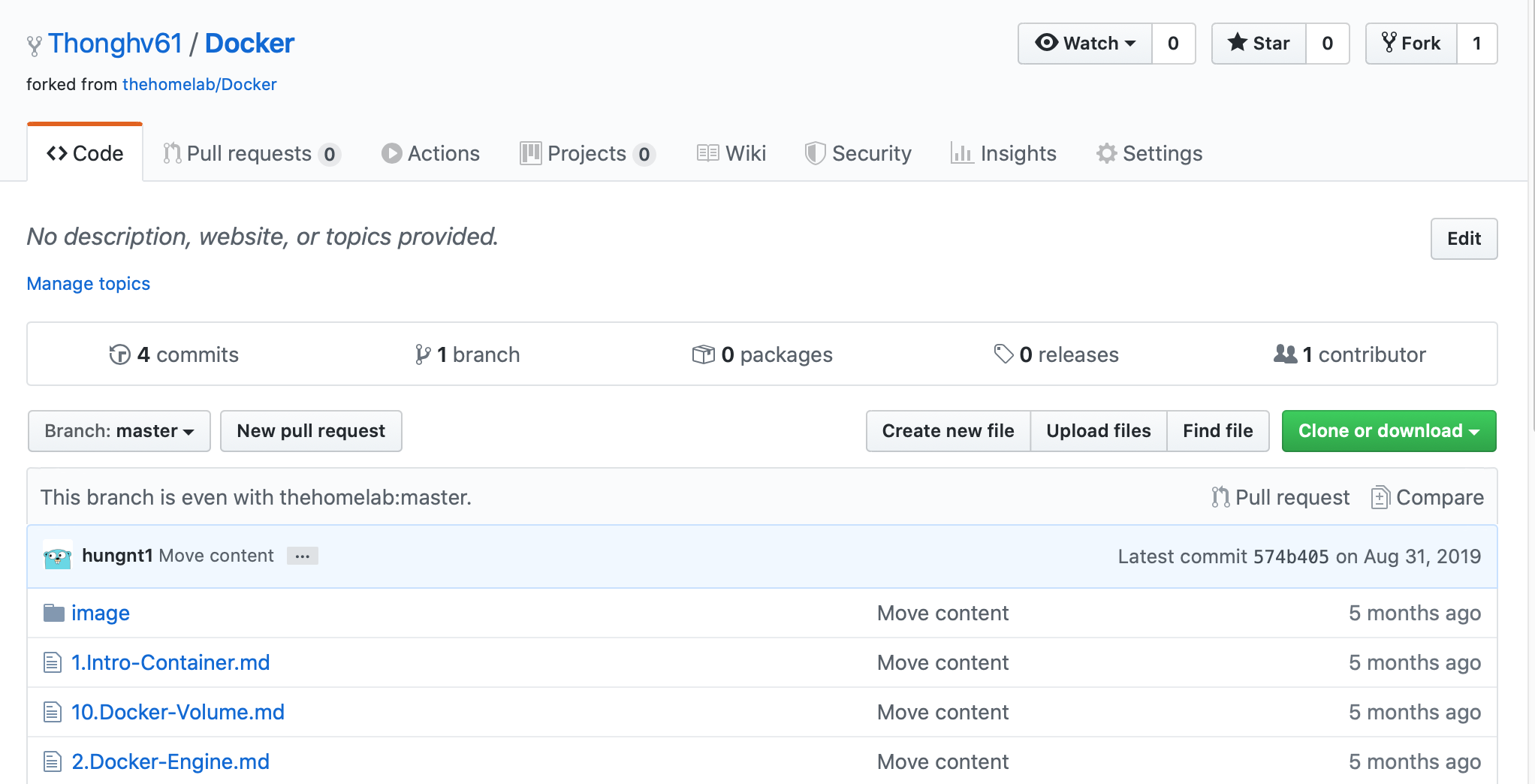
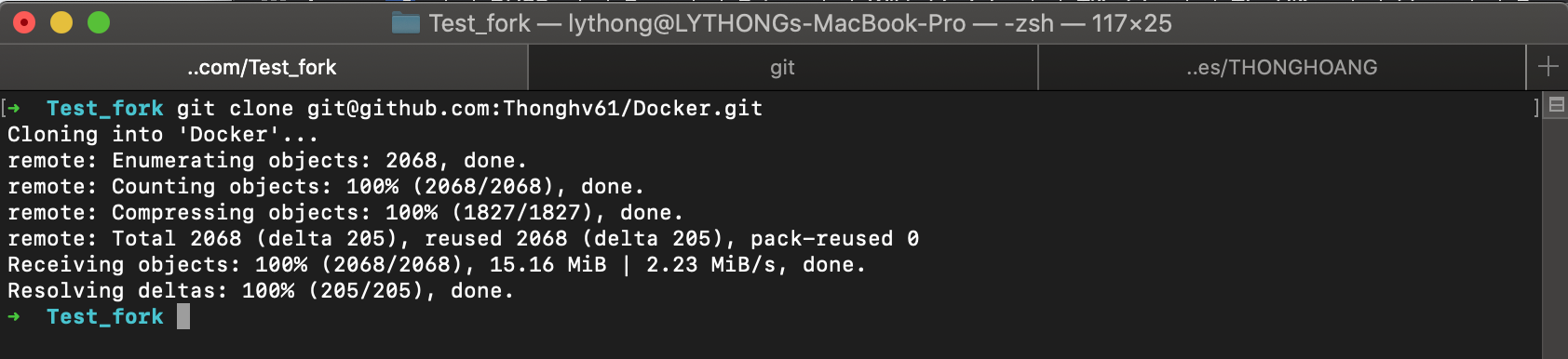
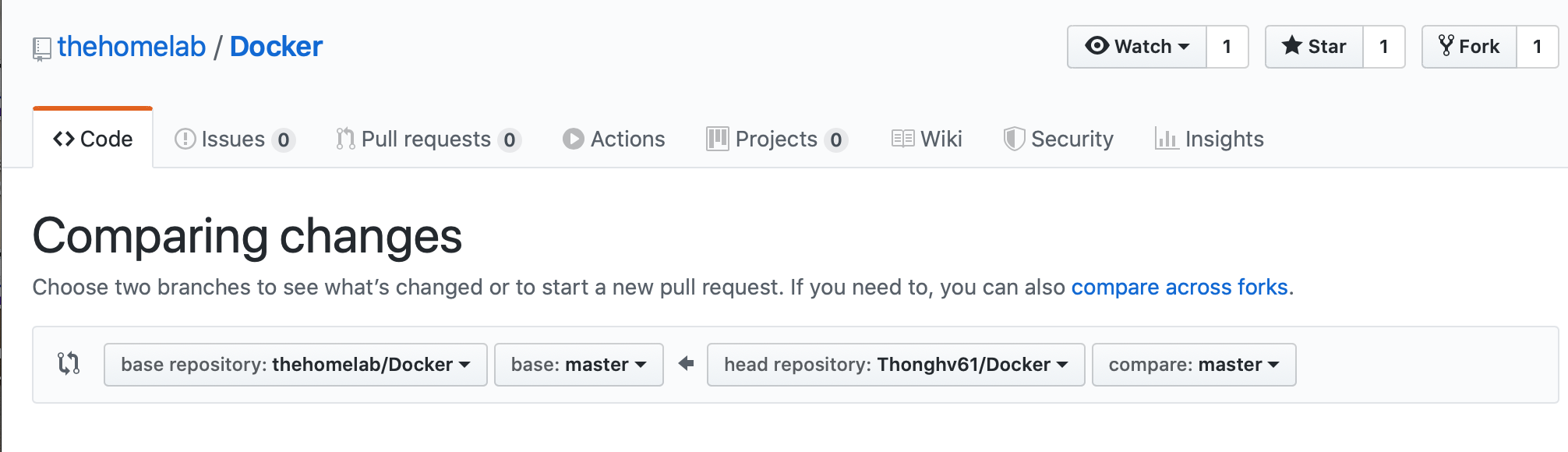
Fresher\_Hoàng Văn Thông  
Bài tập git  
  
1.2  
  
Bài 1: Cài đặt Git, xem phiên bản đang sử dụng  
  
  
  
  
Sau khi cài đặt thành công và kiểm tra version ta được như hình bên trên  
  
Bài 2: Tạo thư mục study\_git  
  
  
  
Bài 3: Tạo file gmo.txt  
  
  
  
Bài 4: Khởi tạo một repository trống  
  
  
  
  
  
Bài 5: Add file gmo.txt vào staging area (index)  
  
  
Bài 6: Commit file này với message “adding gmo.txt”  
  
  
  
Bài 7: Thực hiện check out commit này  
  
  
  
Bài 8: Tạo file z-com.txt, sau đó add vào staging area và commit  
  
  
  
  
  
  
  
  
Bài 9: Xóa file gmo.txt  
   


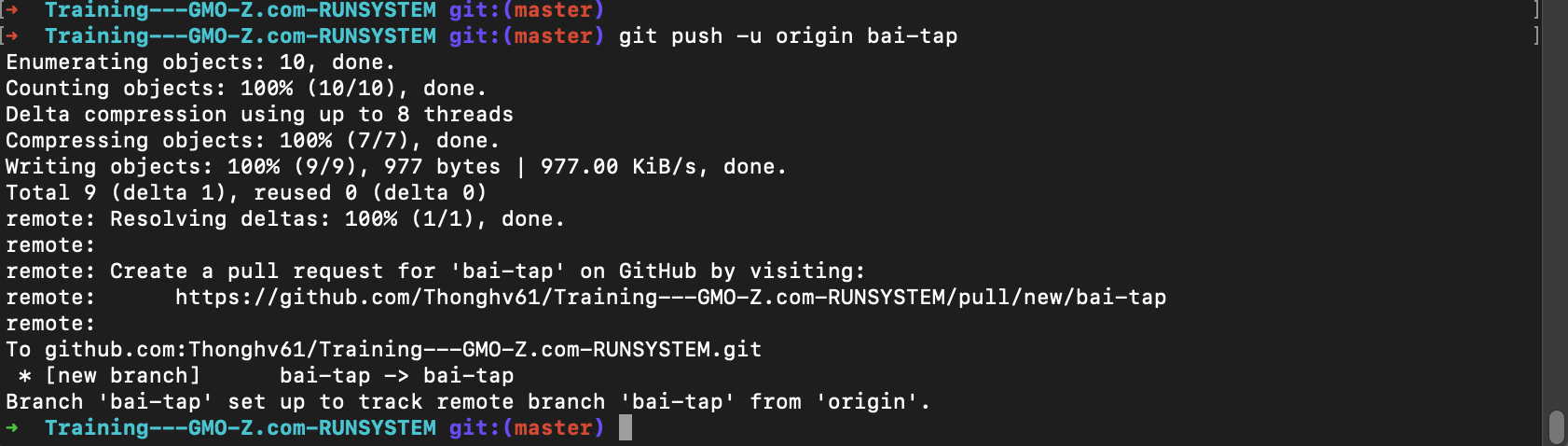
Bài 10: Add thay đổi này vào staging area, commit lên với message “xóa gmo.txt”  
  
  
  
  
  
  
  
Bài 11: Thay đổi cấu hình global core.pager=dog  
  
  
  
Bài 12: Viết lệnh để liệt kê tất cả các cấu hình global cho git.  
  
  
  
2.2. Bài tập  
  
Bài 1: Tạo tài khoản GitHub  
  


Bài 2: Tạo 1 local repository, add, commit 1 số files  
  




Bài 3: Tạo 1 remote repository và push code từ local lên remote  
  
  
  


Bài 4: Fork 1 repo bất kì nào đó – clone và thực hiện gửi pull request  
  
  
  
  
  


Bài 5: Tạo 1 branch mới trên local rồi push lên GitHub  


Bài 6: Gửi 1 pull request với nhánh mới đó trên repo vừa fork ở bài 4  
  
  
  
3.2  
Bài 1: Trả lời các câu hỏi sau vào file “Bai tap Git.docx”

- Lệnh Git clean làm gì?

- Flag -d và -f trong git clean để làm gì?

- Lệnh nào tạo branch?

- Sự khác nhau giữa 2 phương pháp merge fast forward và recursive?

- Lệnh git nào thay đổi sang nhánh khác?

- Làm thế nào để bỏ những file đã sửa hoặc xóa ở trong thư mục làm việc?

- Lệnh nào xóa branch?

- Lệnh git diff làm gì?

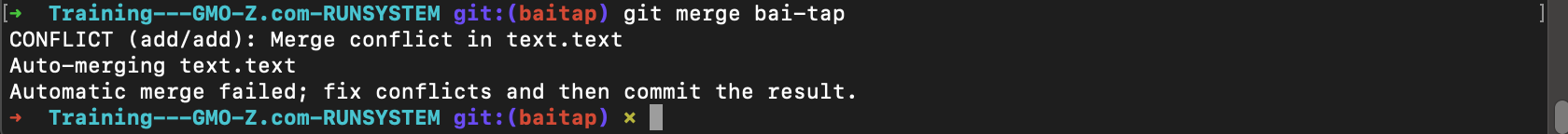
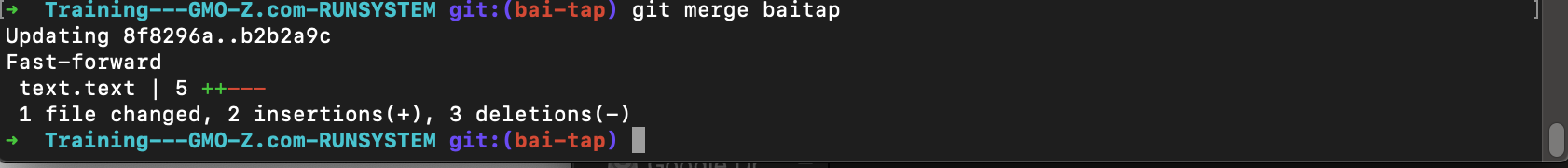
- Làm thế nào để xóa file từ staging area?

- Conflict xảy ra như thế nào? Làm thế nào khắc phục được?

Bài Làm

Bài 1  
- Lệnh Git clean để remove ở trạng thái untracked file  
- Flag -d và -f trong git clean để xoá file hoặc folder  
- Sự khác nhau   
 + Merge fast forward : khi chúng ta merge 2 nhánh với nhau thì nhánh được merge chỉ thêm được cập nhật thêm nhưng file mà nhánh kia không có…  
ví dụ nhánh A có các file a,b,c và nhánh B có các file a,b,c,d, khi chúng ta marge nhánh B vào nhánh A thì nhánh A bây giờ sẻ được cập nhật thêm file d, tức là nhánh A sẻ có file a,b,c,d….   
+ Recursive: Khi chúng ta merge 2 nhánh với nhau thì nhánh được merge sẻ cập nhật các file đã thay đổi trước đó và cập nhật thêm những file mà nhánh được merge không có  
- Lệnh tạo branch git branch <Tên Nhánh>  
- Lệnh thanh đổi nhánh git checkout <Tên nhánh muốn chuyển sang>  
- Xoá những file ở trong thư mục làm việc git rm –f <tên file>

- Lệnh Xoá Branch git branch –D <Tên nhánh>  
- Lệnh git diff dùng để hiện thị thay đổi mà chưa add và commit  
- Xoá file từ staging area git rm –cached <Tên-file>

- Comflict xảy ra khi 2 người cùng làm trên một nhánh và cùng chỉnh sửa một file, cách khắc phục chỉnh sửa file push lên cho phù hợp  
  
Bài 2: Tự tạo conflict bằng cách làm việc với 1 file trên 2 nhánh riêng biệt, sau đó merge 2 nhánh đó lại. Tự sửa lỗi conflict đó.  
  
  
  
  
  
4.2  
Bài 1: Trả lời các câu hỏi sau vào file “Bai tap Git.docx”

- So sánh sự khác biệt giữa git reset và git revert. Khi nào sử dụng?

- So sánh sự khác biệt giữa git merge và git rebase. Khi nào sử dụng?

- So sánh sự khác biệt giữa git stash pop và git stash apply. Khi nào sử dụng?

- Khi rebase ở interactive mode, có thể làm những gì?

Bài 2: Thực hành các kiến thức đã học

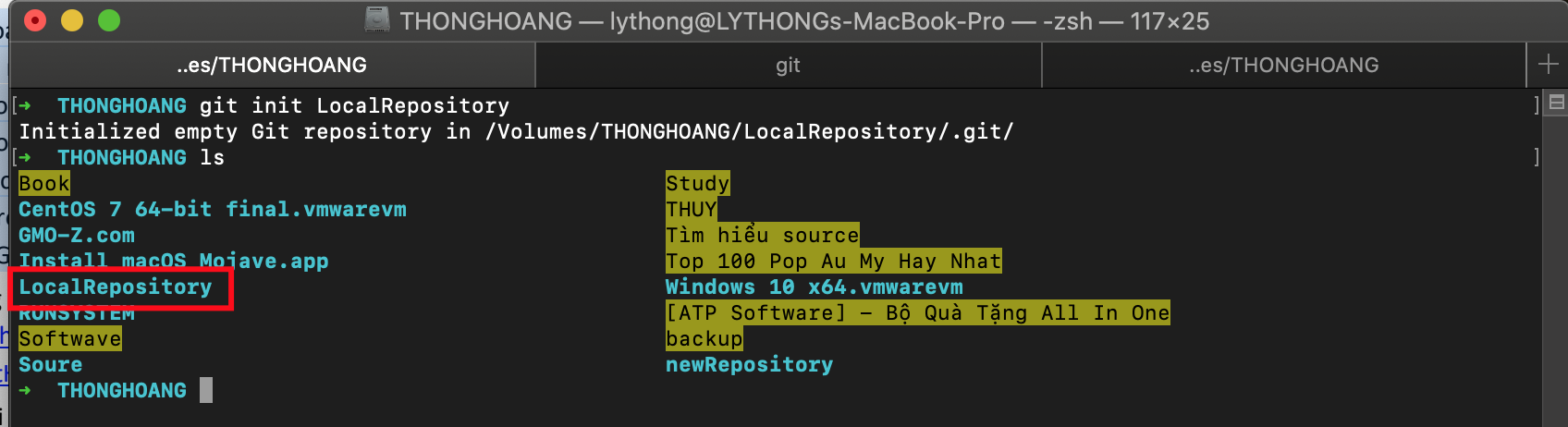
- Tạo local repository từ CLI, add và commit 1 số file, push lên GitHub

- Fork và clone 1 repo bất kì nào đó

- Add và commit 1 số thay đổi, rồi thực hiện pull request  
- Thay đổi sang nhánh khác rồi thêm 1 số file, sau đó merge vào master branch

- Tập sử dụng revert để undo commit và rebase với interactive mode

- Thực hiện pull request mới, với squash commit  
  
Bài làm  
  
Bài 1;  
- Git reset dùng để khôi phục sử dụng khi muốn khổi phục toàn bộ index, hoặc khôi phục về commit 3,2,1. Git revert undo lại commit trước đó dùng khi muốn huỷ commit cũ.

- Khác biệt giữa git merge và git rebase  
 + Git merge khi gộp nhánh chúng ta có thể gộp nhiều nhánh cùng lúc nhưng, không mang theo commit lên nhánh master được  
+ Git rebase cũng tương tự như merge nhưng khi gộp nhánh đồng thời sẻ gộp luôn commit của nhánh con, nên rất dễ theo dõi khi phát hiện ra lỗi  
- Khác biệt stash pop và git stash apply  
+ git stash pop: dùng trở về stash cũ trước đó và đồng thời xoá stash trở về  
+ git stash apply: dùng để trở về stash id bất khì trong list stash  
- Khi rebase ở interactive mode dùng có thể thay đổi commit  
  
Bài 2  
  
- Tạo local repository từ CLI, add và commit 1 số file, push lên GitHub  
  
  
- Fork và clone 1 repo bất kì nào đó  
- Add và commit 1 số thay đổi, rồi thực hiện pull request  
