**THỰC HÀNH LAB3 – MÔN CÔNG CỤ VÀ MÔI TRƯỜNG PHÁT TRIỂN PHẦN MỀM**

**Thông tin sinh viên**

**Họ tên: Hồ Thanh Sang**

**MSSV: 1711060848**

**Lớp chiều T7 10/04/2021**

**PHẦN BÀI LÀM**

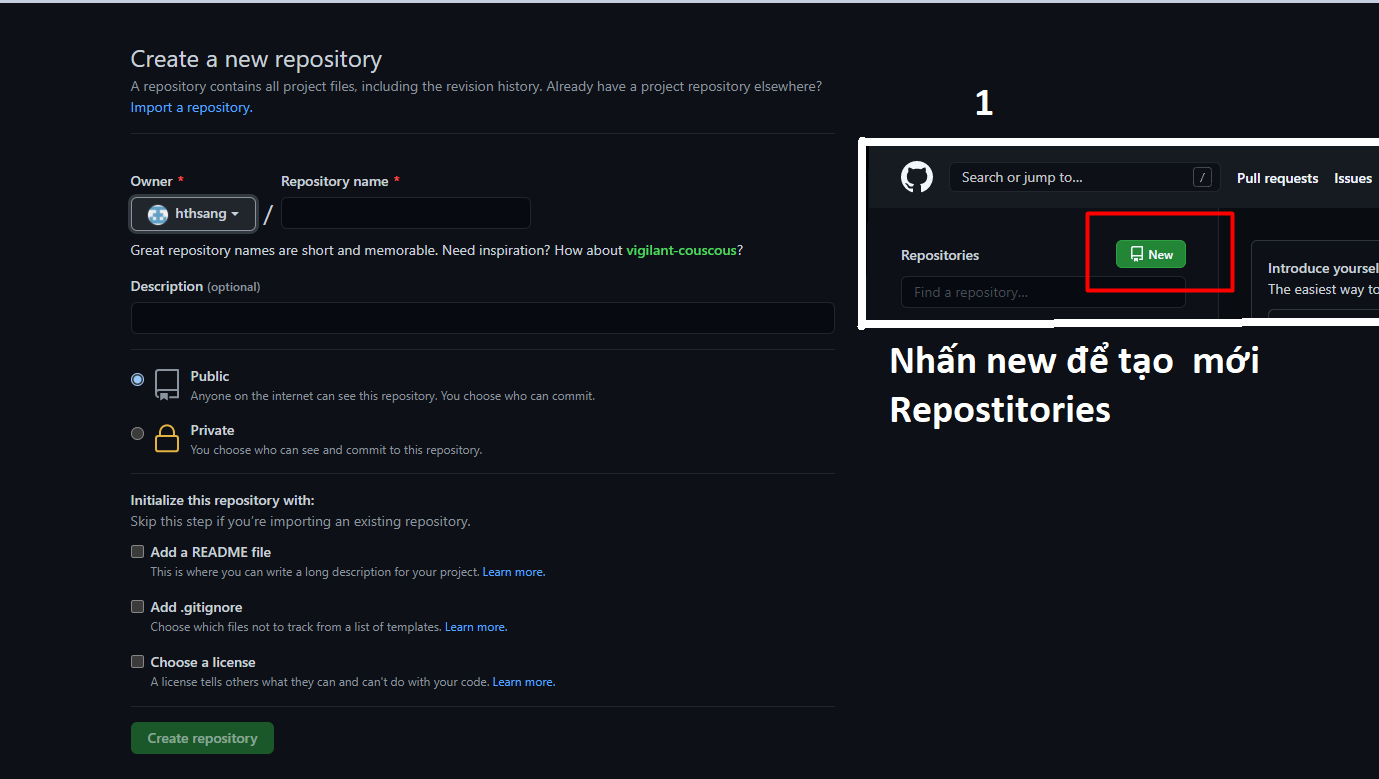
# Phần 1:

## Tạo mới reposive

### Giao diện tạo mới Repo

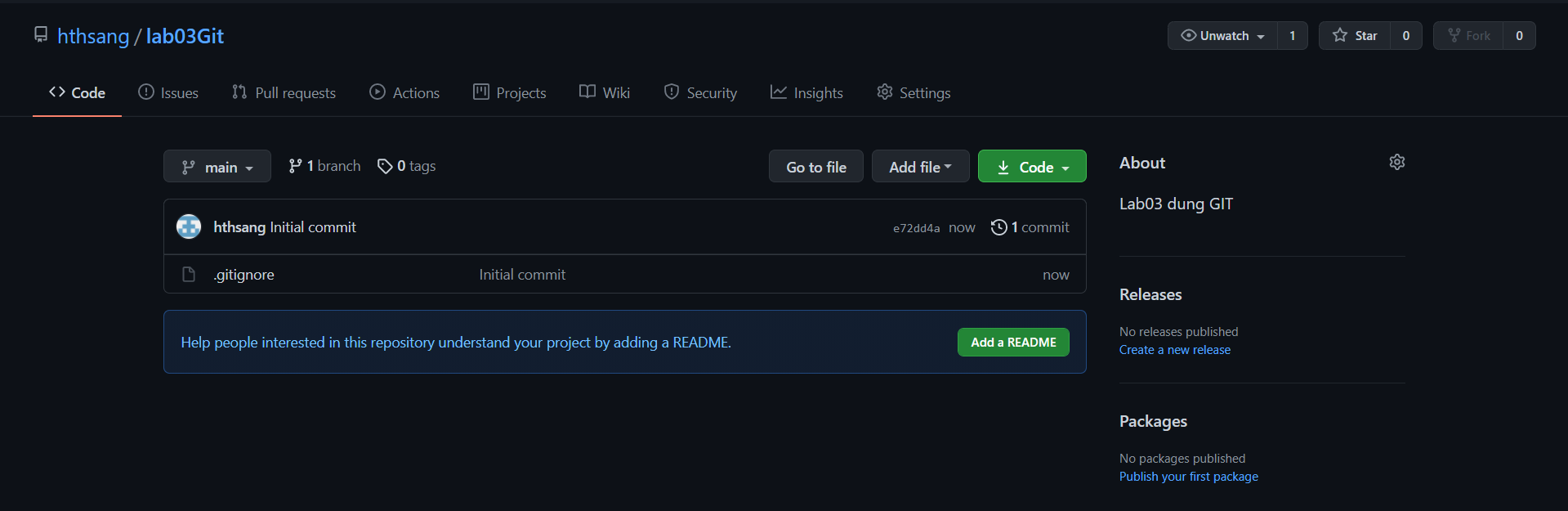
Truy cập vào trang web <https://github.com>

* Ta sẽ nhìn thấy cột bên trái phần trên cùng [1] có mục Repostitories và nút button “New” màu xanh. Ta thực hiện thao tác nhấn vào nút đó để tạo mới Repo
* Trong giao diện “Create a new repository ta lần lượt điền vào các thông tin trong form
* Repo name: Tên repo (bắt buộc)
* Desciption: Mô tả (không bắt buộc)
* Chọn chế độ public nếu muốn công khai và rivate để riêng tư
* Mục Intiallize this repository with ta chọn “Add gitignore” và chọn ngôn ngữ C++
* Sau khi điền xong các thông tin trong form ta nhấn Create Repo



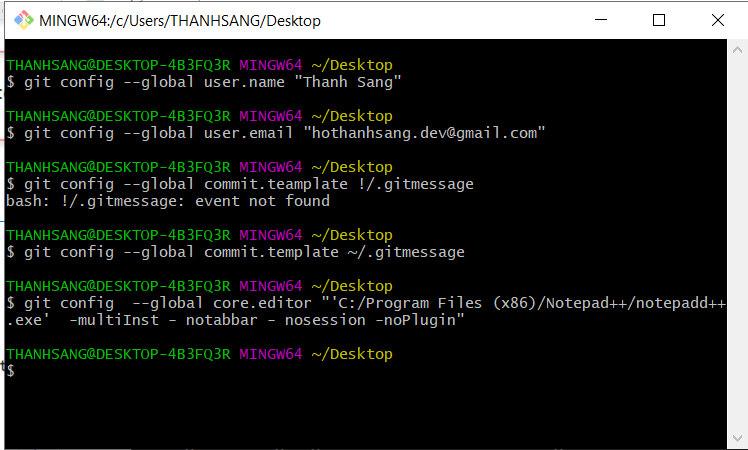
### Giao diện Repo

* Sau khi hoàn thành các thao tác ở phần 1.1 ta sẽ được Repo như hình bên dưới



## Cấu hình, thiết lập cơ bản

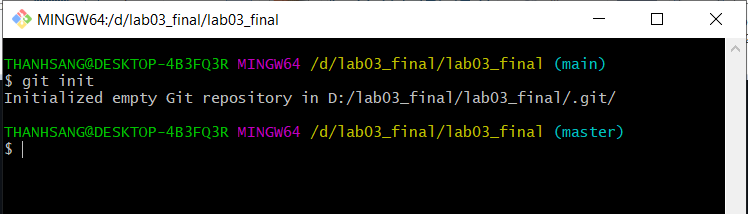
* Thiết lập username: git config –global user.name “…..”
* Thiết lập email: git config –global user.email “…..”
* Thiết lập công cụ Editor mặc định là Notepad ++:
* Thiết lập mẫu template cho commit:



## Thực hiện các lệnh

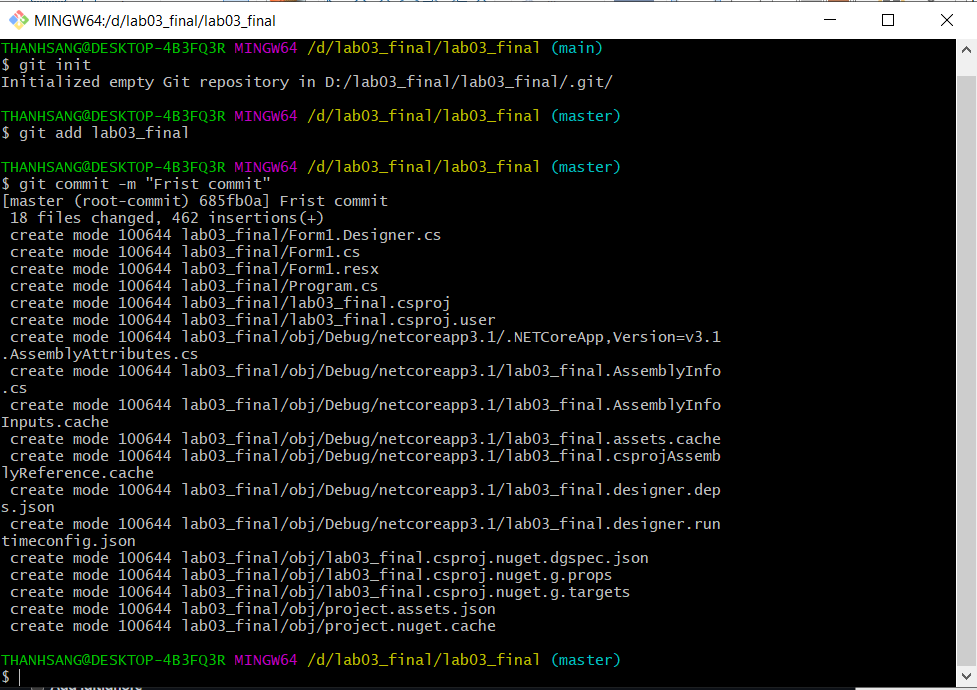
## Tạo repon mới

* Vào thư mục tạo ở phần 3 nháy chuột phải chọn Git Base Here. Sau đó nhập câu lệnh **git init**



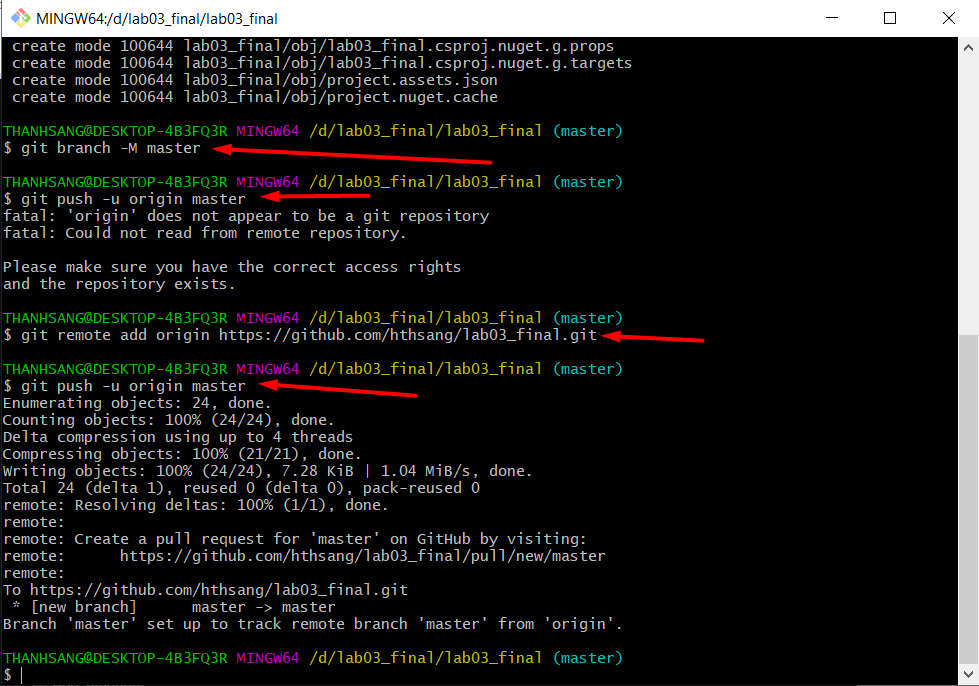
## Commit file

* Tiếp tục 4.1 thì ta nhập vào câu lệnh “git add lab03\_final”
* Lab03\_final là tên thư mục tạo
* Sau đó nhập tiếp câu lệnh “git comit -m “First commit”” để comit file
* First comit là ghi chú cho lệnh comit

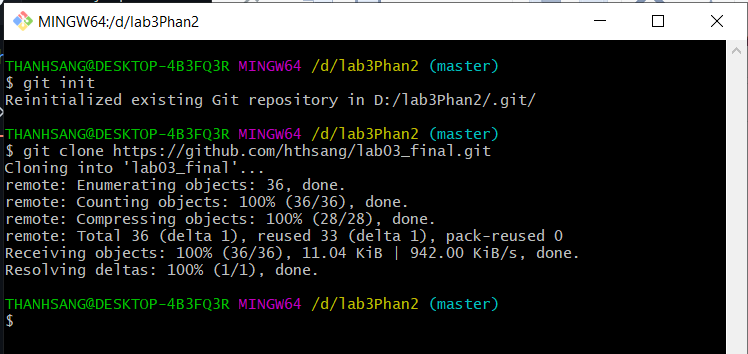


## Push lên remote repository

* Tiếp tục 4.2 thì ta lần lượt gõ các câu lệnh như trong hình

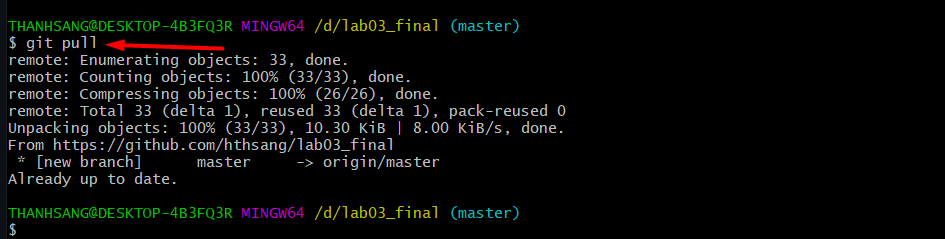


## Clone một remote repository

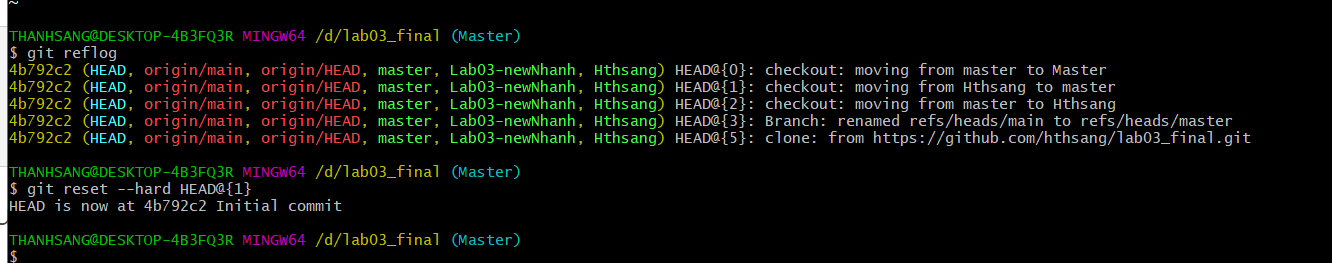


## Pull lên remote repository

* Gõ câu lệnh git pull để pull lên remote repo

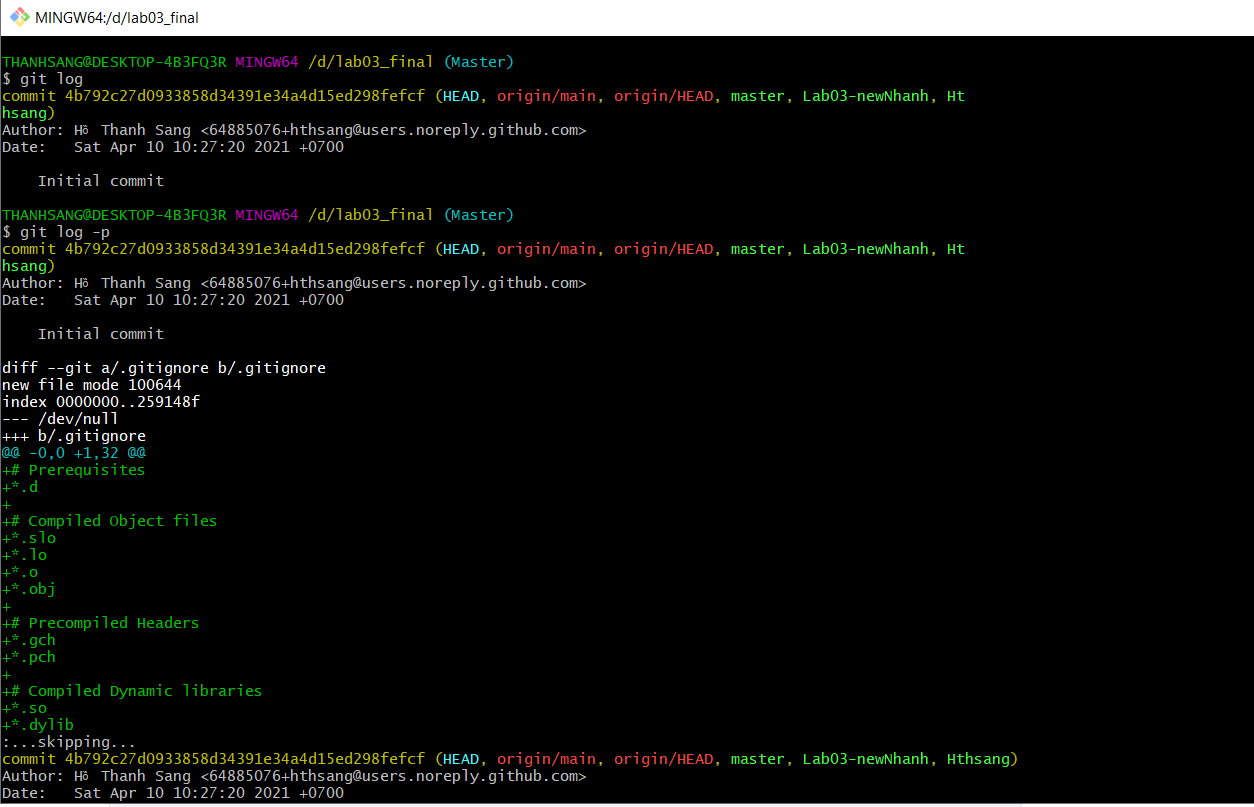


## Lệnh phục hồi về một phiên bản trước

* Gõ câu lệnh git relog để hiển thị tất cả các phiên bản và thao tác đã làm để xem thông tin trước khi phục hồi
* Gõ câu lệnh git reset –hard HEAD@{1}
* Trong đó {1} là phiên bản muốn phục hồi
* 

## Lệnh xem lịch sử các phiên bản

* Gõ câu lệnh git log để xem lịch sử
* Git log -p xem lịch sử một cách chi tiết hơn



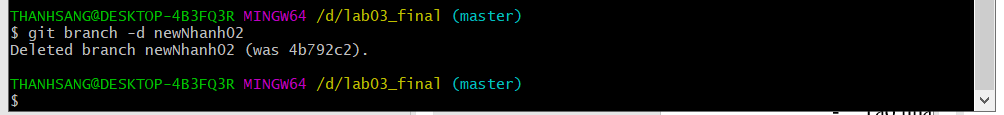
## Tạo nhánh

* Git branch newNhanh02
* Trong đó newNhanh02 là tên nhánh muốn tạo



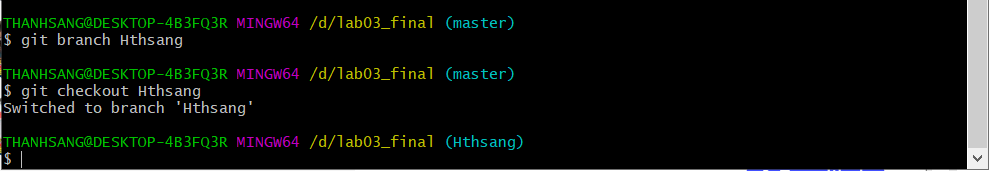
## Xoas nhanh

* Git branch -d newNhanh02
* Trong đó newNhanh là tên nhánh muốn xóa



## Lệnh chuyển qua làm việc tại các nhánh khác nhau

* Git branch Hthsang
* Trong đó Hthsang là tên nhánh muốn chuyển

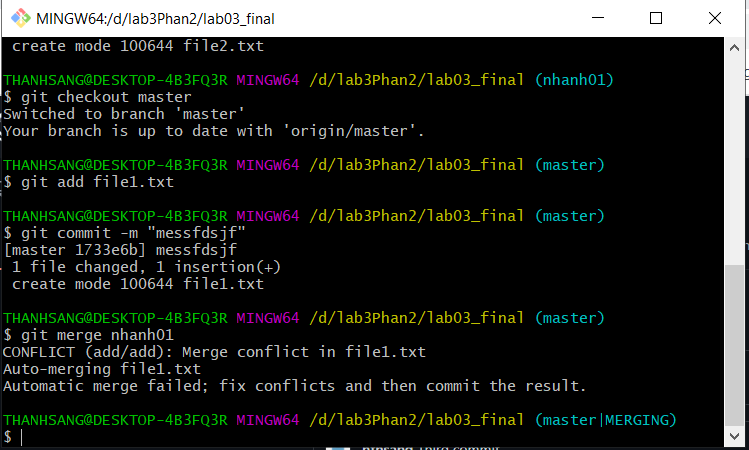


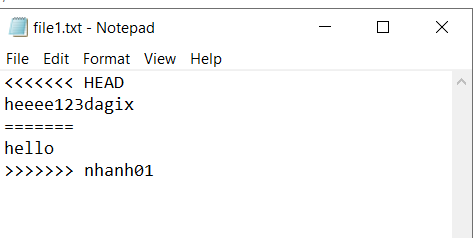
## Lệnh merge từ các nhánh

* Git merge Hthsang
* Trong đó Hthsang là tên nhánh muốn merge

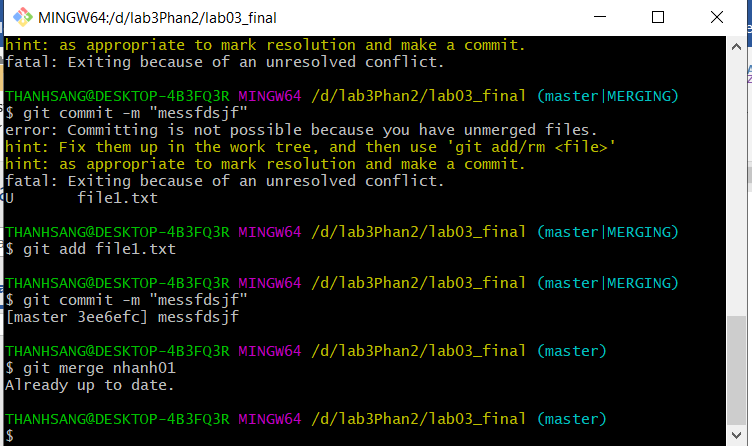
## Xung đột

* Tạo file1.txt với nội dung Hello và up lên bằng bracnh nhanh01 sau đó commit
* Sau đó tạo file cùng tên và với nội dung khác và commit thì xuất hiện xung đột

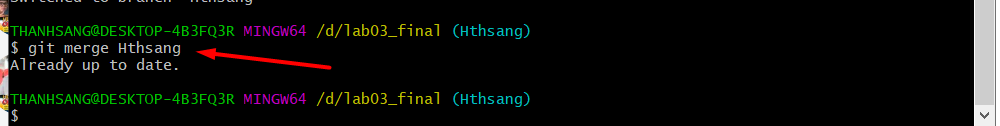




* Sửa file txx cho khác nội dung với file ban đầu



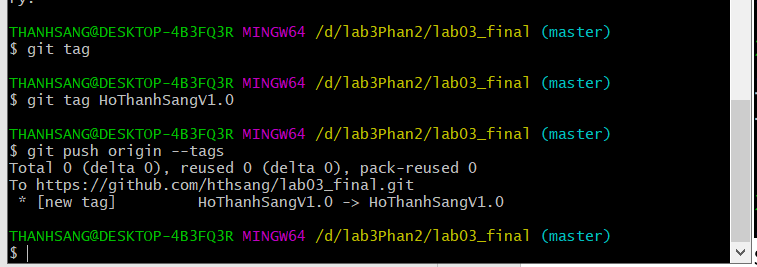
* Sau khi thay đổi nội dung và comit lên lại



# TÌM HIỂU THÊM

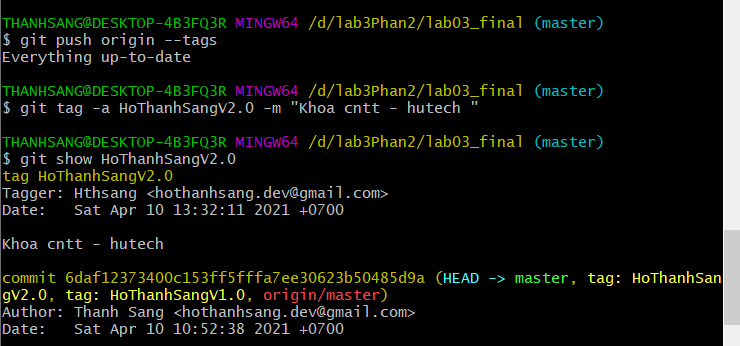
## Thêm 1 Tag một người khác và push lên

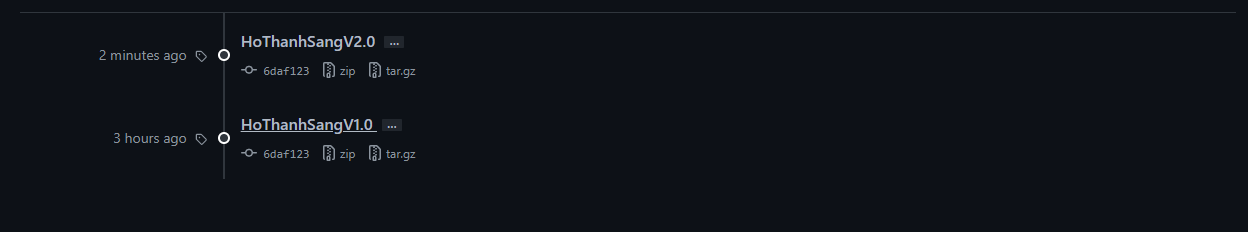
* Lưu ý: Tên viết liền không dấu + V1.0
* vd : HoThanhSangV1.0



## Thêm 1 Tag có tên như Tag trước

* Nhưng là V2.0 và thêm bình luận “khoa cntt - hutech” push lên
* Gõ câu lệnh git tag -a HoThanhSangV2.0 -m “Khoa cntt – hutech”

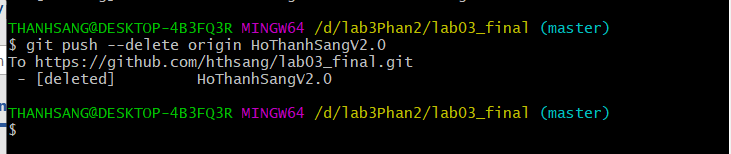




* Tag đã được update trên gibhub

## Xóa Tag thứ 2 đi và chụp lại kết quả

* Gõ lệnh git push –delete origin HoThanhSangV2.0
* Trong đó: chú ý bổ sung –delete / HoThanhSangV2.0 là tên tag muốn xóa



## Lịch sử commit

* Gõ lệnh “git log” để xem tất cả các commit

