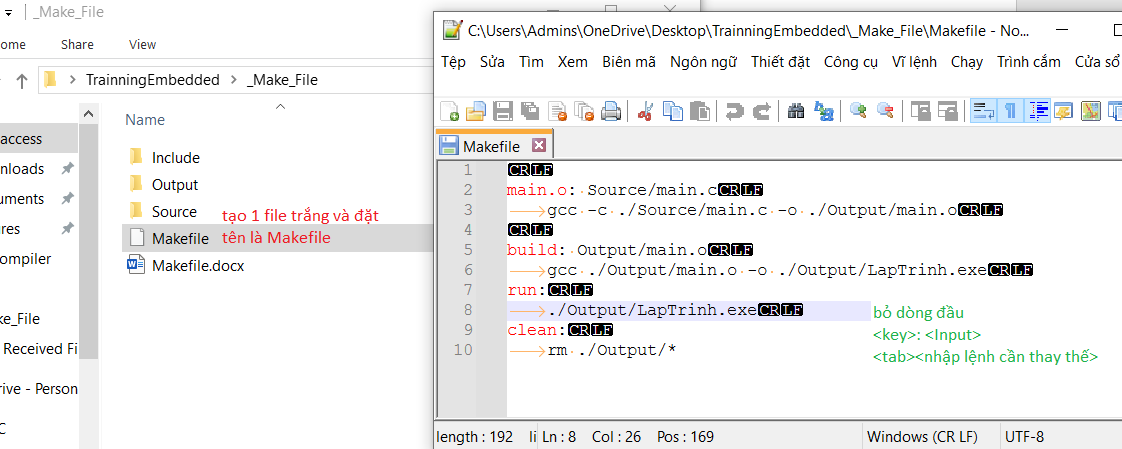
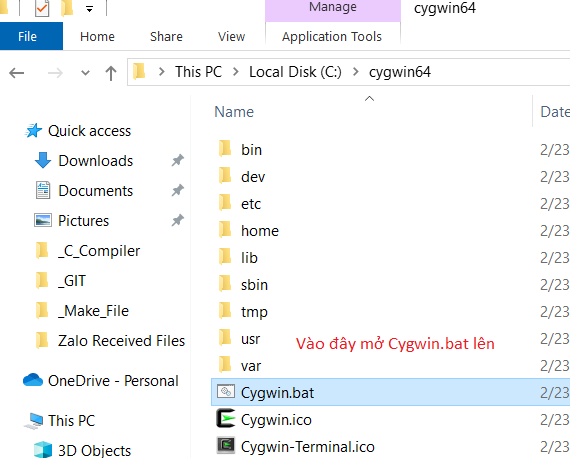
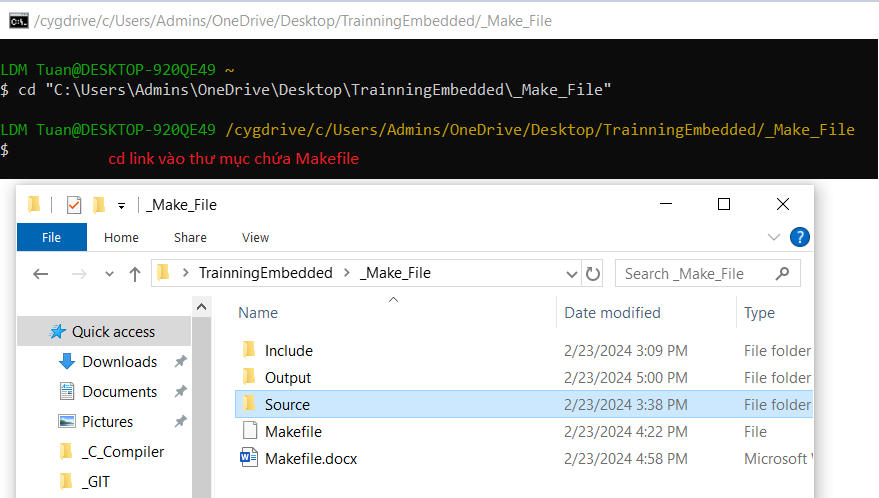
Makefile là gì?

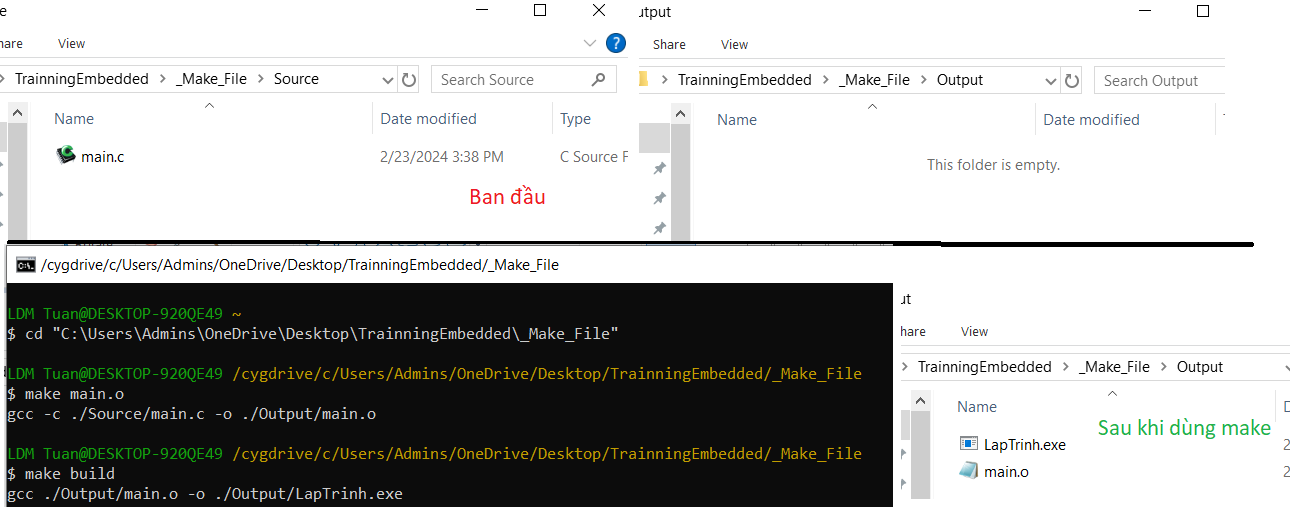
- Bình thường để thực hiện C compiler (hoặc bất kì ứng dụng nào khác) ta phải gõ từng dòng như “gcc -c ….” -> mất thời gian

- Makefile là sẽ viết ra 1 hàm (với guidline riêng) và sẽ thực hiện các chức năng tương ứng

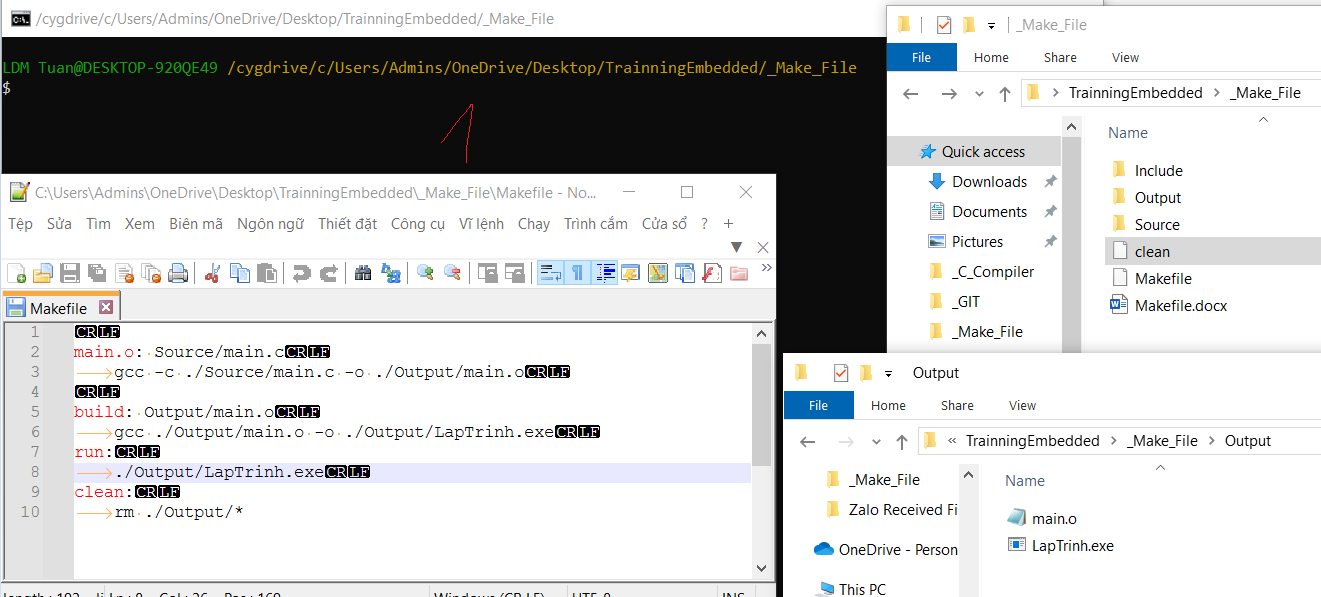


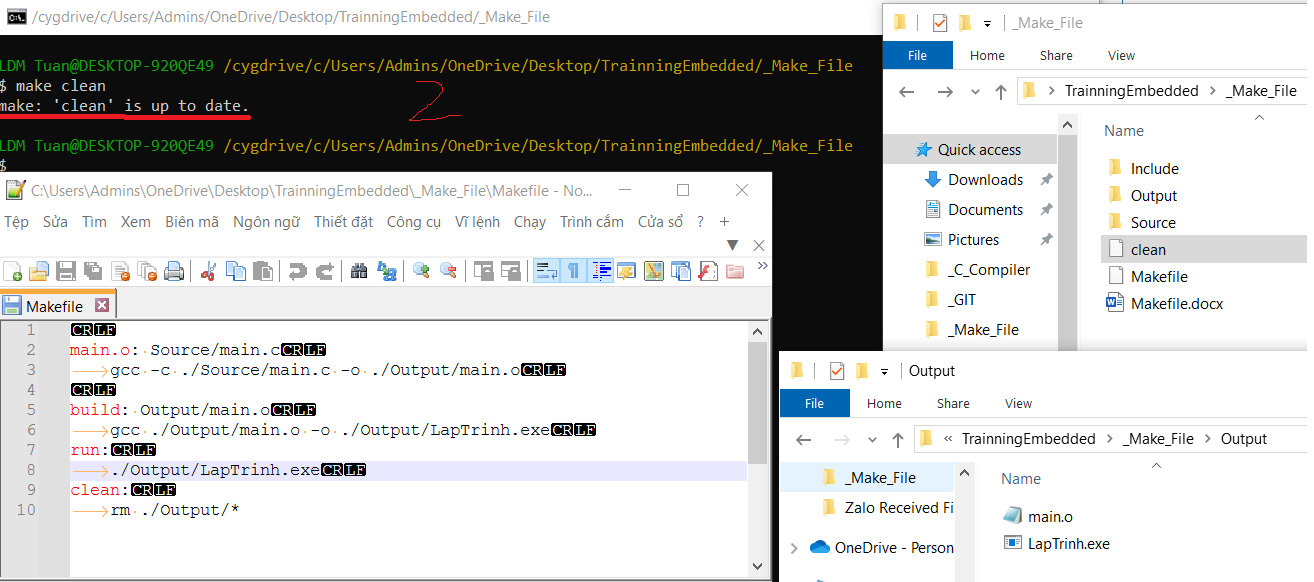




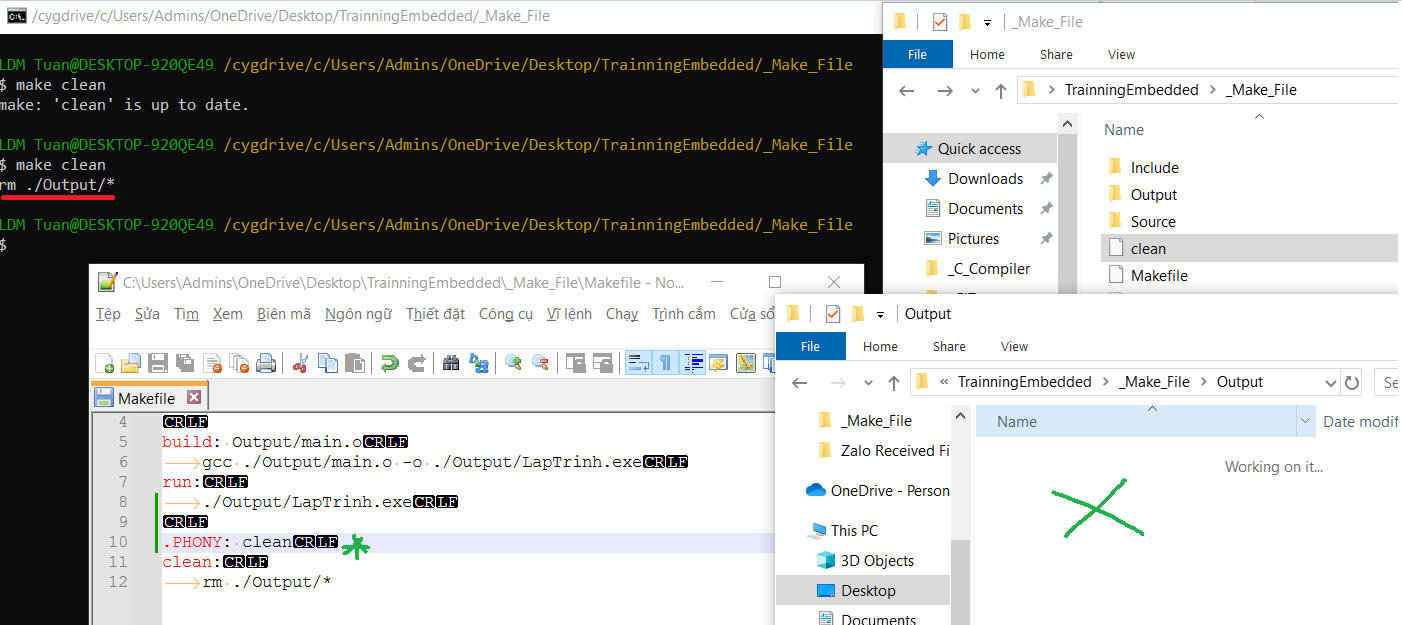


Đặt trường hợp nếu có 1 file có cùng tên với key trong Makefile thì sao? -> câu lệnh trong Makefile sẽ không được thực hiện





Cách khắc phục -> thêm **.PHONY:** <key> ở trên các key

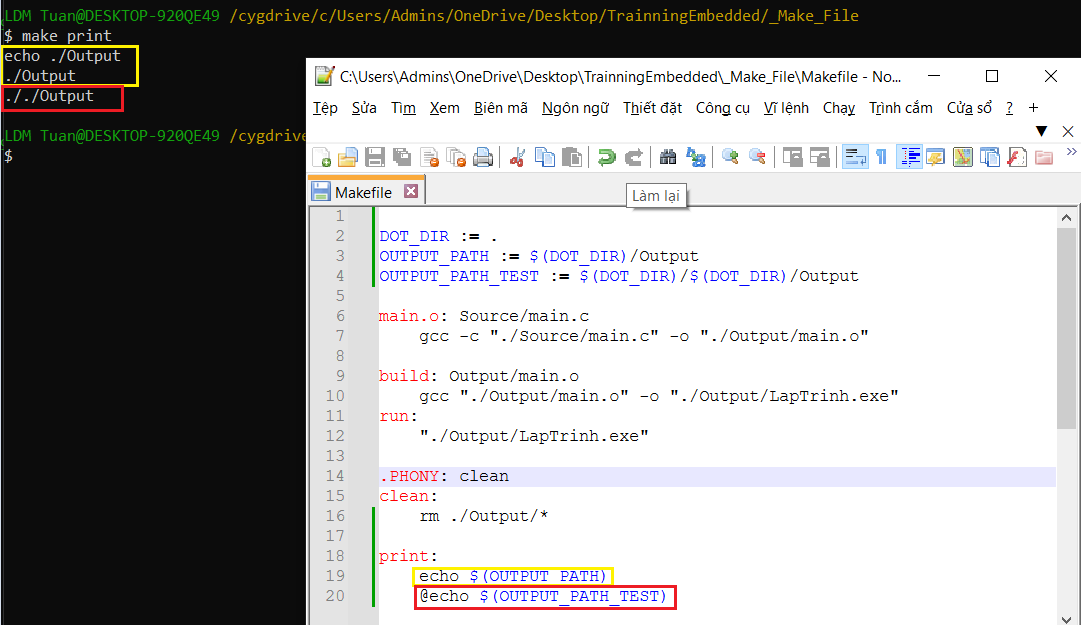


Mỗi lần làm Makefile cứ phải ghi nhiều lần đường link như vậy nhìn sẽ rất rối -> ta sẽ define lại để sử dụng cho thuận tiện và rõ ràng (giống như khi ta làm code C).

-> ta dùng thực hiện: <key define> **:=** <đối tượng define>

-> để truy cập ta dùng **$(**<key define>**)**

-> để **@** trước câu lệnh để ẩn câu lệnh đó đi (ẩn nhưng vẫn thực hiện)



Đối với cách trên thì chỉ áp dụng cho làm thủ công từng file một. Nhưng nếu trong folder Source có nhiều file .c thì lại phải làm lại hoặc bổ sung thêm cho Makefile. Do đó ta phải chuẩn hóa lại làm sao mà input key là tất cả thư mục trong folder chứ không phải là từng file riêng lẻ. Ta thực hiện gồm 4 lệnh:

+ **+=** dùng để gộp nhiều nội dung lại trong 1 <key define>

+ **wildcard** đối với define ta có thể định nghĩa link, thế nhưng ta không thể truy cập hết toàn bộ file có trong link đó -> ta dùng wildcard để truy cập các file trong link đó

+ **foreach** <key define>**, $**<key define> **,** <thực hiện lệnh> có thể xem như for

+ **notdir** <đường dẫn thư file> xóa đường dẫn cho dễ nhìn

+ **subst** <kí tự muốn thay thế>**,** <thay thế thành>**,**<văn bản gốc> thay thế kí tự trong văn bản

