

Make là công cụ dùng để tự động hóa quá trình biên dịch và xây dựng phần mềm. Nó sử dụng một tệp tin gọi là Makefile để xác định các quy tắc và phụ thuộc giữa các tệp nguồn và tệp đích

Trong 1 dự án, Makefile là tệp tin văn bản chứa các quy tắc để biên dịch và liên kết các tệp nguồn thành tệp thực thi hoặc thư viện. Makefile định nghĩa các mục tiêu (targets), phụ thuộc (dependencies) và các lệnh (commands) cần thiết để xây dựng mục tiêu đó.

Cú pháp cơ bản của Makefile bao gồm:

```
target: dependencies  
        command
```

- `target` : Tên của tệp đích mà bạn muốn tạo ra (ví dụ: tệp thực thi hoặc thư viện).
- `dependencies` : Các tệp nguồn hoặc tệp khác mà target phụ thuộc vào.
- `command` : Lệnh để xây dựng target từ dependencies. Lưu ý rằng lệnh phải bắt đầu bằng một tab.

Ví dụ đơn giản về Makefile:

```
main: main.c  
      gcc -o main main.c
```

Trong ví dụ trên, khi bạn chạy lệnh `make main`, Make sẽ kiểm tra xem tệp `main.c` có thay đổi không. Nếu có, nó sẽ thực thi lệnh `gcc -o main main.c` để biên dịch tệp nguồn thành tệp thực thi `main`.

Câu lệnh thực thi trong terminal là

```
make main
```

Tuy nhiên, ở giai đoạn 2026, Make đã trở nên ít phổ biến hơn so với các công cụ xây dựng hiện đại như CMake, Meson, hoặc Bazel, đặc biệt trong các dự án phức tạp hoặc đa nền tảng. Các công cụ này cung cấp nhiều tính năng hơn và dễ sử dụng hơn so với Make truyền thống. Đặc biệt là CMake được ưa chuộng rộng rãi trong cộng đồng phát triển phần mềm hiện đại do khả năng tạo Makefile tự động và hỗ trợ nhiều nền tảng khác nhau.