## Bộ Tiền Xử Lý (Preprocessor) trong C

### Tổng Quan về Bộ Tiền Xử Lý

Bộ tiền xử lý trong C (C Preprocessor - CPP) là một công cụ xử lý mã nguồn **trước khi biên dịch**. Nó thực hiện các lệnh bắt đầu bằng dấu #, được gọi là **chỉ thị tiền xử lý (preprocessor directives)**.

**Chức năng chính của bộ tiền xử lý:**

* **Chèn file header** (#include)
* **Định nghĩa macro** (#define)
* **Điều kiện biên dịch** (#ifdef, #ifndef, #if, #else, #elif, #endif)
* **Khai báo inline cho macro** (#undef)
* **Tạo thông báo lỗi** (#error)
* **Chèn lệnh compiler** (#pragma)

### Các Chỉ Thị Tiền Xử Lý Quan Trọng

#### #include - Nhúng file tiêu đề

Chỉ thị này được dùng để nhúng các **header file** vào chương trình.

**a. Nhúng thư viện chuẩn**

Cú pháp:

|  |
| --- |
| #include <stdio.h> |

Ví dụ:

|  |
| --- |
| #include <stdio.h>  int main() {      printf("Hello, World!\n");      return 0;  } |

* #include <file.h> tìm file trong thư mục hệ thống.

**b. Nhúng file do người dùng định nghĩa**

Cú pháp:

|  |
| --- |
| #include "myheader.h" |

* #include "file.h" tìm file trong thư mục làm việc hiện tại trước.

#### #define - Định nghĩa Macro

Dùng để định nghĩa **hằng số hoặc macro**.

**a. Định nghĩa hằng số**

|  |
| --- |
| #define PI 3.14159  #define MAX 100 |

Ví dụ:

|  |
| --- |
| #include <stdio.h>  #define PI 3.14159  int main() {      printf("Chu vi hình tròn bán kính 5: %.2f\n", 2 \* PI \* 5);      return 0;  } |

**b. Định nghĩa macro có tham số**

|  |
| --- |
| #define SQUARE(x) ((x) \* (x)) |

Ví dụ:

|  |
| --- |
| #include <stdio.h>  #define SQUARE(x) ((x) \* (x))  int main() {      printf("Bình phương của 5 là: %d\n", SQUARE(5));      return 0;  } |

**c. Xóa bỏ macro với #undef**

|  |
| --- |
| #undef PI |

Ví dụ:

|  |
| --- |
| #include <stdio.h>  #define PI 3.14  #undef PI  #define PI 3.14159  // Thay đổi giá trị của PI  int main() {      printf("Giá trị PI: %f\n", PI);      return 0;  } |

#### #ifdef, #ifndef, #if, #else, #elif, #endif - Điều Kiện Biên Dịch

Chỉ thị tiền xử lý cho phép **biên dịch có điều kiện**.

**a. #ifdef và #ifndef**

* #ifdef MACRO: Nếu macro đã được định nghĩa thì biên dịch.
* #ifndef MACRO: Nếu macro **chưa** được định nghĩa thì biên dịch.

Ví dụ:

|  |
| --- |
| #include <stdio.h>  #define DEBUG  #ifdef DEBUG      #define LOG(x) printf("DEBUG: %s\n", x)  #else      #define LOG(x)  #endif  int main() {      LOG("Chương trình đang chạy");      return 0;  } |

Nếu #define DEBUG tồn tại → LOG sẽ được in.

**b. #if, #else, #elif, #endif**

Cho phép sử dụng điều kiện.

Ví dụ:

|  |
| --- |
| #include <stdio.h>  #define VERSION 2  #if VERSION == 1      #define FEATURE "Phiên bản 1"  #elif VERSION == 2      #define FEATURE "Phiên bản 2"  #else      #define FEATURE "Phiên bản không xác định"  #endif  int main() {      printf("Chương trình đang chạy ở: %s\n", FEATURE);      return 0;  } |

#### #error - Tạo lỗi tùy chỉnh

Dùng để ép trình biên dịch dừng lại nếu một điều kiện không thỏa mãn.

Ví dụ:

|  |
| --- |
| #if \_\_STDC\_\_ != 1      #error "Chương trình cần một trình biên dịch C tiêu chuẩn"  #endif |

Nếu trình biên dịch không hỗ trợ chuẩn C, nó sẽ dừng và báo lỗi.

#### #pragma - Các chỉ thị dành riêng cho compiler

#pragma là chỉ thị **phụ thuộc vào trình biên dịch**.

Ví dụ:

|  |
| --- |
| #pragma message("Chương trình đang biên dịch...") |

Một số chỉ thị phổ biến:

* #pragma once: Tránh lỗi include file trùng lặp.
* #pragma warning(disable: 4996): Tắt cảnh báo (Visual Studio).
* #pragma pack(1): Canh chỉnh bộ nhớ.

Ví dụ: #pragma once:

|  |
| --- |
| #pragma once  void sayHello(); |

### Lưu Đồ Hoạt Động của Bộ Tiền Xử Lý

1. **Xóa bỏ comment** (//, /\* \*/).
2. **Xử lý chỉ thị #include**: Nhúng code từ header file.
3. **Xử lý #define**: Thay thế macro vào mã nguồn.
4. **Xử lý #ifdef, #ifndef**: Xóa bỏ hoặc giữ lại phần code tương ứng.
5. **Gửi mã đã xử lý đến trình biên dịch**.