



知识点总结



- 在C语言中，**编译预处理**是C语言编译过程中的一个重要阶段，它会在正式编译之前对源程序进行一些**文本替换和处理工作**。其中，**#include**是编译预处理指令的一种。

- **#include指令**：用于将指定文件的内容插入到当前源文件中该指令出现的位置。

- 它有两种使用形式：

- 1、**#include <头文件名>**：用于包含C标准库头文件、操作系统提供的系统头文件等。

例如，`#include <stdio.h>` 用于包含标准输入输出头文件，该文件包含了如printf、scanf等函数的声明。

在这个过程中，预处理器会在系统默认的头文件搜索路径中查找指定的头文件，不会检查当前目录。

- 2、**#include "头文件名"**：用于包含用户自定义的头文件。

例如，假设你自己编写了一个名为myheader.h的头文件，就可以使用 `#include "myheader.h"` 将其内容包含到当前源文件中。

在这个过程中，预处理器会**先在当前源文件所在的目录中查找指定的头文件**，如果找不到，**再到系统默认的头文件搜索路径中继续查找**。



- 编译预处理：

C语言提供的编译预处理功能主要有文件包含、宏定义和条件编译。

- 文件包含：

通过`#include`指令，将其他文件的内容包含进来，方便代码复用。

例如：当多个源文件都需要用到某些通用的函数声明时，可将其整合至一个头文件中。随后，在各个源文件内，使用`#include`指令包含该头文件，以此避免在每个源文件中重复编写相同代码。

- 宏定义：

使用`#define`指令定义宏，包括不带参数的宏和带参数的宏。宏在预处理阶段进行文本替换，不带参数的宏常用于定义常量，而带参数的宏可用于替换重复的代码片段。



- 条件编译：通过**#ifdef**、**#ifndef**、**#if**、**#else**、**#elif**和**#endif**等指令实现条件编译。
- 条件编译指令的作用范围是**单个编译单元**，即**单个.c文件以及这个.c文件包含的所有头文件**。
- 利用这些指令，可根据不同的条件，决定是否编译源程序的某些部分。

例：

```
#ifdef DEBUG
```

```
    printf("Debug information: variable value is %d\n", var);
```

```
#endif
```

在上述代码中，**#ifdef DEBUG**检查宏**DEBUG**是否已定义。

若已定义，则会编译printf语句，用于输出调试信息。

若未定义，**#ifdef**与**#endif**之间的这部分代码不会被编译进最终的可执行文件。

· 注意：1、头文件的包含顺序：一般来说，应优先包含**标准库**和**系统头文件**，最后才是**用户自定义头文件**。2、可通过条件编译指令（如**#ifndef**、**#define**和**#endif**组合），避免头文件的重复包含。

例：

```
// myheader.h

#ifndef MYHEADER_H
#define MYHEADER_H

// 头文件内容

#endif
```

在上述代码中，**#ifndef MYHEADER_H**检查宏**MYHEADER_H**是否未定义。
若未定义，执行**#define MYHEADER_H**定义该宏，并编译中间的头文件内容。
若已定义，**#ifndef**与**#endif**之间的头文件内容不会被编译进最终的可执行文件。



✓ 题目1、程序中使用putchar函数和getchar函数，需在程序的开头加上一行代码：#include <stdio.h>。



✓ 题目2、C语言提供的预处理功能主要有文件包含、宏定义、和条件编译。



- ! 预处理命令行可以出现在程序的任何一行的开始，只要符合语法规则就行
- 一行上只能有一条有效的预处理命令
- 预处理命令就是在程序正式编译之前被处理的

题目3、 以下关于编译预处理的叙述中错误的是 (A)

- A. 预处理命令行必须位于C源程序的起始位置
- B. 在C语言中，预处理命令行都以“#” 开头
- C. 一行上只能有一条有效的预处理命令
- D. 预处理命令是在程序正式编译之前被处理的