

Lec 05 Introduction to C

GCC编译器

GCC (GNU Compiler Collection) 的工作流程：

1. **预处理器**：处理宏替换等指令（如 `#include`、`#define`）。
2. **编译器**：将源代码转化为目标文件。
3. **链接器**：将多个目标文件和库文件组合成可执行文件。

C语言的基本结构

```
#include <stdio.h>
#define STOP 0
int global; // 全局变量
int main() {
    int counter, startPoint; // 变量声明
    printf("Enter a positive integer: "); // 输出提示
    scanf("%d", &startPoint); // 输入
    for (counter = startPoint; counter >= STOP; counter--) { // 控制结构
        printf("%d\n", counter);
    }
    return 0; // 返回值
}
```

包含了：预处理器指令、全局变量、main函数、变量声明、I/O操作、控制结构和返回值。

预处理器指令

1. `#include <stdio.h>` 在编译前将标准输入输出头文件内容复制到源代码中。
2. `#define STOP 0` 在编译前将所有的"STOP"替换为"0"。

main函数

- main函数是C程序的入口点。
- 常见形式：
 - `int main(void)` 无参数。
 - `int main(int argc, char *argv[])` 接受命令行参数。其中：
 - `int argc` (ARGument Count)是一个整数变量，它存储用户传递的命令行参数的数量，包括程序的名称。因此，如果我们向程序传递一个值，`argc`的值将是2（一个是参数，一个是程序名）。
 - `char *argv[]` (ARGument Vector)是列出了所有参数的一个字符指针数组。
 - 如果 `argc > 0`，那么从 `argv[0]` 到 `argv[argc-1]` 的数组元素将包含指向字符串的指针。`argv[0]` 是程序名，之后直到 `argv[argc-1]` 的所有元素都是命令行参数。

```
int main(int argc, char *argv[]) {
    printf("Argument count: %d\n", argc);
    printf("First argument: %s\n", argv[1]);
    return 0;
}
```

变量和数据类型

基本数据类型：

- int：整型。
- char：字符型。
- float/double：浮点型。

类型修饰符：signed、unsigned、long、short，用于调整数据类型的大小。

变量的声明和初始化。

全局变量和局部变量：

- 全局变量(Global Variables)
 - 定义：在所有函数外部声明的变量，作用域覆盖整个程序。
 - 示例：

```
int global; // 全局变量
int main() {
    // 可在此访问 global
}
```

- 特性：
 - 存储位置：静态存储区（Global data section），程序运行期间始终存在。
 - 生命周期：从程序开始到结束。
 - 初始化：未显式初始化时默认为0。
 - 访问范围：任何函数都可以访问和修改（除非使用static限制作用域到文件内）。
- 局部变量(Local Variables)
 - 定义：在函数或块内部生命的变量，作用于限于声明它的块。
 - 示例：

```
int main() {
    int counter;    // 局部变量
    int startPoint; // 局部变量
}
```

- 特性：
 - 存储位置：运行时栈（Run-time stack），由栈帧管理。

- **生命周期**：仅在函数调用期间存在，函数返回时销毁。
- **初始化**：未显式初始化时值为随机（未定义行为）。
- **访问范围**：仅在声明它的函数或块内有效。

输入输出

头文件<stdio.h>：

- #include <stdio.h>引入标准输入输出库，提供了printf和scanf函数。

printf函数：

- 格式：printf("格式字符串", 参数列表);
- "Enter a positive integer: "是静态字符串，直接输出。
- "%d\n"是格式化字符串，%d表示输出一个十进制整数，\n换行，counter作为参数替换%d。

scanf函数：

- 格式：scanf("格式字符串", &变量);
- "%d"指定读取十进制整数，&startPoint是变量的地址，&表示将输入值存储到该地址。

格式化字符串：

%d：十进制整数、%f：浮点数、%s：字符串、%c：字符

运算符

赋值运算符：=、算术运算符：*, /, %, +, -, //、位运算符：~, <<, >>, &, ^, |、逻辑运算符：!, &&, ||、增量/减量运算符：++, --

复合赋值运算符：+=, -=, *=, /=, %=