

基本数据类型

参考 C Primer Plus 第五版 中文 ch03-p51 基本数据类型与类型大小

预处理

编译程序前，先由预处理器检查程序。
根据程序中使用的预处理器指令，用符号缩略语所代表的内容替换程序中的缩略语。

存储类型

C 语言存储类型				
存储类	存储时期	作用域	链接	声明方式
自动	自动	代码块	空	代码块内
寄存器	自动	代码块	空	代码块内，使用关键字 register
具有外部链接的静态	静态	文件	外部	所有函数之外
具有内部链接的静态	静态	文件	内部	所有函数之外，使用关键字 static
空链接的静态	静态	代码块	空	代码块内，使用关键字 static

说明：

- ① 存储时期
- 静态存储时期的变量，在程序执行期间将一直存在。
自动存储时期的变量，在程序进入定义这些变量的代码块时，将为这些变量分配内存；
当退出这个代码块时，分配的内存将被释放。
- ② 作用域
- 指定变量能被访问的区域，包含代码块，函数原型和文件作用域。
函数原型作用域：仅在函数声明中使用、形参。
`void use_VLA(int n, int m, arr[n][m]);`
- ③ 链接
- 说明在不同文件中出现的相同标识符应该如何处理。链接属性分为 3 类：**external**（外部）、**internal**（内部）和 **none**（空）。

C/C++混合编程

```
extern "C" {}
```

为了让 C++ 能够与 C 接口而采用的一种语法形式。之所以采用这种方式，是因为两种语言之间的一些差异所导致的。

C++ 支持多态性，也就是具有相同函数名的函数可以完成不同的功能，C++ 通常是通过参数区分具体调用的是哪一个函数。在编译的时候，C++ 编译器会将参数类型和函数名连接在一起，于是在程序编译成为目标文件以后，C++ 编译器可以直接根据目标文件中的符号名将多个目标文件连接成一个目标文件或者可执行文件。但是在 C 语言中，由于完全没有多态性的概念，C 编译器在编译时除了会在函数名前面添加一个下划线之外，什么也不会做（至少很多编译器都是这样干的）。由于这样的原因，当采用 C++ 与 C 混合编程的时候，就可能会出问题。

结构体、指针与内存分配

参见代码示例。

参考资料：

C Primer Plus 第五版 中文 ch13-存储类、链接和内存管理

C 和指针 第二版

<https://blog.csdn.net/rcj183419/article/details/45459969>