

高级语言程序设计

张伯雷

bolei.zhang@njupt.edu.cn

bolei-zhang.github.io

软件教学中心

助教：陈然

高级语言程序设计

第02章 初识C源程序及其数据类型

1. 初识计算机、程序与C语言

- 什么是计算机？
- 存储程序思想？
- 程序语言，为什么学习C语言？
- 进制转化

- C语言源程序的组成结构
- 基本数据类型**常量**
- 基本数据类型**变量**

预测未来的最好办法，就是把它创造出来。

The best way to predict the future is to invent it.

——艾伦·凯 (Alan Kay) ,图灵奖得主

2.1 C源程序及其符号

```
#include <stdio.h>
```

库 (library)包含区

```
/*函数功能：计算两个整数的乘积
```

```
入口参数：整型数a和b
```

```
返回值： 整型数a和b之积 */
```

注释区

```
int multiply( int a, int b)
```

```
{
```

```
    return (a*b);
```

```
}
```

程序区

```
/*主函数*/
```

```
int main( ){
```

```
    int x, y, product;
```

```
    printf("Please input two integers:");
```

```
    scanf("%d%d", &x, &y);    /*输入两个整型数x和y*/
```

```
    product = multiply(x, y);    /*调用函数multiply计算x和y的乘积*/
```

```
    printf("The product is %d\n", product); /*输出x 和y的乘积*/
```

```
    return 0;
```

```
}
```

2.1 C源程序及其符号

```
#include <stdio.h>
```

```
/*函数功能：计算两个整数的乘积
```

```
入口参数：整型数a和b
```

```
返回值：整型数a和b之积 */
```

```
int multiply( int a, int b)
```

```
{
```

```
    return (a*b);
```

```
}
```

```
/*主函数*/
```

```
int main( )
```

```
{ int x, y, product;
```

```
    printf("Please input two integers:");
```

```
    scanf("%d%d", &x, &y); /*输入两个整型数x和y*/
```

```
    product=multiply(x, y); /*调用函数multiply计算x和y的乘积*/
```

```
    printf("The product is %d\n", product); /*输出x 和y的乘积*/
```

```
    return 0;
```

```
}
```

函数是C程序的基本单位

用户自定义函数

函数的定义：
函数首部、函数体{语句；}

主函数：程序的入口

库函数

函数的调用：
函数名(参数...)

2.1 C源程序及其符号

```
#include <stdio.h>
```

```
/*函数功能：计算两个整数的乘积
```

```
入口参数：整型数a和b
```

```
返回值： 整型数a和b之积 */
```

```
int multiply( int a, int b)
```

```
{
```

```
    return (a*b);
```

```
}
```

```
/*主函数*/
```

```
int main( )
```

```
{ int x, y, product;
```

```
    printf("Please input two
```

```
    scanf("%d%d", &x, &y);
```

```
    product = multiply(x, y);
```

```
    printf("The product is %d\n", product);
```

```
    return 0;
```

```
}
```

定义3个整型变量

输入提示

输入2个数

获得计算结果

输出结果

输入两个整型数x和y*/

/*调用函数multiply计算

/*输出x和y的乘积*/

输入(Input)

处理(Process)

输出(Output)

2.1 C源程序及其符号

C程序的6种符号

1 关键字：附录B

2 标识符：系统预定义、用户自定义

```
#include <stdio.h>
```

```
/*函数功能：计算两个整数的乘积
```

```
入口参数：整型数a和b
```

```
返回值： 整型数a和b之积 */
```

```
int multiply( int a, int b)
```

```
{
```

```
    return (a*b);
```

```
}
```

```
/*主函数*/
```

```
int main( )
```

```
{ int x, y, product;
```

```
    printf("Please input two integers:");
```

```
    scanf("%d%d", &x, &y);    /*输入两个整型数x和y*/
```

```
    product = multiply(x, y);    /*调用函数multiply计算x和y的乘积*/
```

```
    printf("The product is %d\n", product); /*输出x 和y的乘积*/
```

```
    return 0;
```

```
}
```


自定义标识符的命名

- 自定义标识符主要包括变量名、函数名
- 由英文字母、数字和下划线组成，区分大小写
- 不可以是数字开头
- 不允许使用关键字

- 不合法

2a s# C\$ char π

- 合法的，但不推荐

xyz xuehao fabs sqrt

- 符合语法与习惯

ID sum average box_length

2.1 C源程序及其符号



```
#include <stdio.h>
```

```
/*函数功能：计算两个整数的乘积
```

```
入口参数：整型数a和b
```

```
返回值：整型数a和b之积 */
```

```
int multiply( int a, int b)
```

```
{
```

```
    return (a*b);
```

```
}
```

```
/*主函数*/
```

```
int main( )
```

```
{ int x, y, product;
```

```
    printf("Please input two integers:");
```

```
    scanf("%d%d", &x, &y);    /*输入两个整型数x和y*/
```

```
    product=multiply(x, y);    /*调用函数multiply计算x和y的乘积*/
```

```
    printf("The product is %d\n", product); /*输出x 和y的乘积*/
```

```
    return 0;
```

```
}
```

C程序的6种符号

1 关键字：附录B

2 标识符：系统预定义、用户自定义

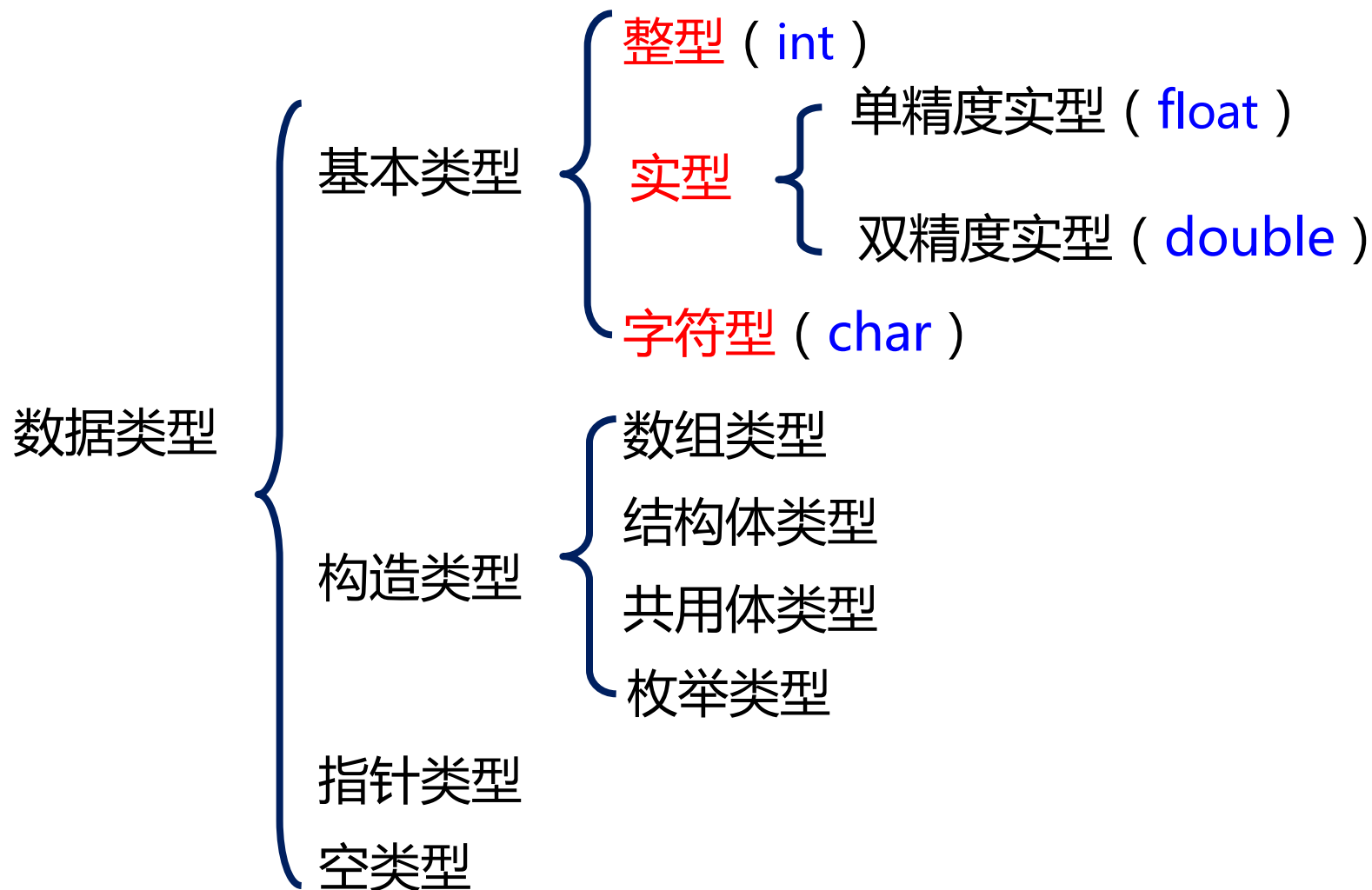
3 运算符：附录D

4 分隔符：空格、回车、逗号、分号

5 其它符号：{ }、/* */、//

6 数据：常量

2.2 C语言中的数据类型



- 整型 (int)
 - 在VS2010环境下占4个字节，范围是多少？
- 实型或浮点型
 - 单精度浮点数(float)，在VS2010下占4个字节，范围？
 - 双精度浮点数(double)，在VS2010下占8个字节
- 字符型 (char)
 - 在VS2010环境下占1个字节
- sizeof(数据类型)

数据类型的修饰符

- short/long (占用字节的多少)
 - short int , 短整数 , 一般2个字节长。通常简写为short
 - long int , 长整数 , 一般4个字节长。通常简写为long
 - long double , 长双精度 (高精度) 浮点数。
- signed/unsigned (数据的正负)
 - signed用来修饰char、int , 说明他们是有符号的整数 (正整数、0和负整数) 。
通常缺省
 - unsigned用来修饰char、int , 说明他们是无符号的整数 (正整数和0)
- 表2.1

2.3 常量



- 常量 (Constant) 是计算机程序运行过程中其值不能发生改变的量。
 - 整型常量
 - 实型常量
 - 字符常量
 - 字符串常量
 - 符号常量

- 十进制：18，-31
- 八进制在前面加上数字0：023，057
- 十六进制在前面加上0x或者0X：0x1a，0XA
- 整型常数默认是int型
- 后缀l、L、u、U、lu、LU等：123456l，123456Lu

- 十进制小数形式：123.45、456.78
- 指数形式：1e-2、4.5e3
- 实型常量缺省为double型
- 可加后缀l、L、f、F：3.14f、3.14L
- float型数据提供7位有效数字，为什么？
- double型数据提供16位有效数字

实型都是不精确的

- 用一对单引号 ' ' 括起来
 - 可打印字符：'a'、'A'、'5'、'%'、' ' 等。
 - 控制字符：用 '\ ' 开头的转义字符，如 '\n' 等。

表2.4

- ASCII码 (附录A)
 - 每个字符都对应一个整数值，称为ASCII码。
(American Standard Code for Information Interchange)
 - 计算机存储、读取字符，以ASCII值进行，如：'A'对应 65、'a'对应97等。

字符相关的一些技巧

- 应熟记

- 'A' ----65 'a' ----97
- '0' ----48 '' ----32

- 技巧

- 大小写字母的转换：

- 小写字母-32=大写字母
如：' b' -32=' B'

- 数字字符与整数数字的转换

- 数字字符-48=整数数字
如：8+48=' 8'

- 用一对**双引号**将零个或多个字符序列括起来
 - 如：“hello” ， “This is a program” ， “A” 等
- 每一个用双引号括起来的字符串常量的末尾都**添加一个空字符 ‘\0’ 作为结束标记**

| | | | | | | |
|----------|----|-----|----|----|----|-----|
| "hello"↵ | h↵ | e↵ | l↵ | l↵ | o↵ | \0↵ |
| ↵ | ↵ | ↵ | ↵ | ↵ | ↵ | ↵ |
| "A"↵ | A↵ | \0↵ | ↵ | ↵ | ↵ | ↵ |

- 字符串常量的实际字符数总是比其双引号中的**字符数多1**

- 5 5.0 '5' "5"
 - 5是一个整型常量，是int型。
 - 5.0是一个实数常量，是float型。
 - '5'是一个字符常量，char型。
 - "5" 是一个字符串常量。

- 定义形式：`#define` <标识符> <字符串>
 - 如：`#define PI 3.14159`
 - 举例
 - 注意：`#define`不是C语言语句，后面没有分号

(进一步说明见第9章)

- **写程序就和书法一样**
- **不仅要体现正确的逻辑和功能，同时要有美观**
 - **在完成大型项目的时候，往往是多人合作的，让别人看懂自己的代码很重要**
 - **函数设计**
 - **命名**
 - **层次感**
 - **注释**
 - **逻辑**
 - **...**

10844 lines (9220 sloc) | 273 KB

```

1 // SPDX-License-Identifier: GPL-2.0-only
2 /*
3  * kernel/sched/core.c
4  *
5  * Core kernel scheduler code and related syscalls
6  *
7  * Copyright (C) 1991-2002 Linus Torvalds
8  */
9 #define CREATE_TRACE_POINTS
10 #include <trace/events/sched.h>
11 #undef CREATE_TRACE_POINTS
12
13 #include "sched.h"
14
15 #include <linux/nospec.h>
16
17 #include <linux/kcov.h>
18 #include <linux/scs.h>
19
20 #include <asm/switch_to.h>
21 #include <asm/tlb.h>
22
23 #include "../workqueue_internal.h"
24 #include "../fs/io-wq.h"
25 #include "../smpboot.h"
26
27 #include "pelt.h"
28 #include "smp.h"
29
30 /*
31  * Export tracepoints that act as a bare tracehook (ie: have no trace event
32  * associated with them) to allow external modules to probe them.
33  */

```

```

492 bool raw_spin_rq_trylock(struct rq *rq)
493 {
494     raw_spinlock_t *lock;
495     bool ret;
496
497     /* Matches synchronize_rcu() in __sched_core_enable() */
498     preempt_disable();
499     if (sched_core_disabled()) {
500         ret = raw_spin_trylock(&rq->__lock);
501         preempt_enable();
502         return ret;
503     }
504
505     for (;;) {
506         lock = __rq_lockp(rq);
507         ret = raw_spin_trylock(lock);
508         if (!ret || (likely(lock == __rq_lockp(rq)))) {
509             preempt_enable();
510             return ret;
511         }
512         raw_spin_unlock(lock);
513     }
514 }
515

```


- **编写一个程序：**
- **从键盘输入一个圆柱体的底面半径和高，计算其表面积和体积，最后在屏幕输出结果。**

- C语言的结构
- 常量



2.4 变量



- 变量 (Variable) 是计算机程序运行过程中其值可以发生改变的
量
 - 常量 (Constant) 是计算机程序运行过程中其值不能发生改变的量
- 变量要求：先定义后使用
- 变量的定义：
 类型名 变量名 ;
 short i;
 long sum;
 double score;
 char c;

变量的初始化



```
int i=5;
```

```
int j, k=6, m;
```

```
double y = float x = 3.1f; Example: 1.print.c
```

```
char c = 'T' + 32;
```

未初始化的变量的值是随机值！

用图示法理解！

比较：

```
int i;
```

```
i = 5;
```

变量赋值，不是初始化

变量的输入

- **scanf** 输入函数

- 数据输入：数据从输入设备送到内存中的变量

- 格式：

- scanf(格式控制字符串, 变量地址表列);

int a; scanf("%d", &a);

float x; scanf("%f", &x);

double y; scanf("%lf", &y);

char c; scanf("%c", &c);

- 表2.5 函数scanf的格式转换说明符



不要加\n

变量的输入和输出——scanf

- `scanf("%d%d",&a,&b);`

若从键盘输入：1<空格>2<回车>
则输入结果：a=1，b=2

- `scanf("%d,%d",&a,&b);`

若从键盘输入：1,2<回车>
则输入结果：a=1，b=2

- `scanf("%2d%3d%4d",&a,&b,&c);`

若从键盘输入：1234567890<回车>
则输入结果：a=12，b=345，c=6789

- 表2.6 函数scanf的格式修饰符

变量的输出

- **printf** 输出函数
 - 数据输出：数据从内存中的变量送到输出设备
- 格式：
 - printf(格式控制字符串, 输出值参数表);
`printf("Hello world!\n");`
`printf("The product is %d\n", product);`
`printf("size = %d \n", sizeof(product));`
 - 表2.7 函数printf的格式转换说明符
 - 表2.8 函数printf的格式修饰符

• 例如：

- `printf("%f, %f", 3.14, 3.14159265);`

输出结果为：3.140000, 3.141593

- `int a=127;`

- `printf("%d, %o, %x", a, a, a);`

输出结果为：127, 177, 7f

- `char ch='A';`

- `printf("%c, %d", ch, ch);`

输出结果为：A, 65

- 例如：

- `printf("%.10f, %.10f", 3.141592653589793, 3.141592653589793f);`

输出结果为： 3.1415926536, 3.1415927410

Example: 1.print.c

变量的输入和输出

• 例2.2 日期格式转换

| 日期格式分类 | 日期格式 | 示 例 |
|--------|-------------|------------|
| 标准 | YYYY-MM-DD | 2018-9-10 |
| 中国 | YYYY年MM月DD日 | 2018年9月10日 |
| 美国 | MM/DD/YYYY | 9/10/2018 |
| 英国 | DD/MM/YYYY | 10/9/2018 |

◉ 运行该程序并分析运行结果

变量的输入和输出

• 例2.2 日期格式转换

| 日期格式分类 | 日期格式 | 示 例 |
|--------|-------------|------------|
| 标准 | YYYY-MM-DD | 2018-9-10 |
| 中文 | YYYY年MM月DD日 | 2018年9月10日 |
| 美国 | MM/DD/YYYY | 10/9/2018 |

思考题1：若日期输入格式为YYYYMMDD，且严格按4位年、2位月和2位日的宽度进行输入，不足宽度的需在前面补0。程序应如何修改？

思考题2：若输入scanf语句中的变量前忘写了取地址符&，程序运行结果会怎样？

变量的输入和输出——getchar、putchar

- 字符型变量的输入和输出函数
 - `<变量> = getchar();`
 - `putchar(<参数>);`
- 例2.3 作业等级的输入和输出
 - 在VS2010环境下运行该程序并分析运行结果

**思考题：若用户从键盘输入ABC<回车>，
上例的运行结果会怎样？**

用const修饰符限定变量



- 只读变量 (Read-Only Variable)

- const 数据类型 只读变量名 ;

- `const double pi=3.14159;`

比较符号常量 : #define PI 3.14159

- 例2.4

- 如果修改pi ?

设计程序时如何定义变量

- 根据求解问题**确定变量的类型**。（字符型、整型、实型）
- 估计变量的取值范围，**防止溢出**，并兼顾**数据精度**等要求。
- 给变量取名字，**符合编程规范**。

*2.5 基本数据类型在计算机内部的表示

- 整型数据在内存中的存储形式
 - 原码 (True Code)
 - 反码 (Ones-Complement Code)
 - 补码 (Complement Code)
- 整数都是以二进制补码的方式存储的
 - 正数的补码就是其原码
 - 负数的补码是其反码+1

整型数据在内存中的存储形式

- short型整数-32645和-123的补码**

-32645 的原码

| | | |
|---|----------|-----------|
| 1 | 111 1111 | 1000 0101 |
|---|----------|-----------|

-123 的原码

| | | |
|---|----------|-----------|
| 1 | 000 0000 | 0111 1011 |
|---|----------|-----------|

-32645 的反码

| | | |
|---|----------|-----------|
| 1 | 000 0000 | 0111 1010 |
|---|----------|-----------|

-123 的反码

| | | |
|---|----------|-----------|
| 1 | 111 1111 | 1000 0100 |
|---|----------|-----------|

-32645 的补码

| | | |
|---|----------|-----------|
| 1 | 000 0000 | 0111 1011 |
|---|----------|-----------|

思考：为什么short整数的取值范围为：-32768~32767？

字符型数据在内存中的存储形式

- 字符型数据的长度为1个字节
- 在内存中是以其对应的ASCII码（0~127）的二进制形式存放存储的。

字符'A'↵

0100 0001↵

实型数据在内存中的存储形式

- 二进制的浮点方式将实数分为阶码和尾数两部分进行存储



- 例如：
 - 十进制数12.625对应的二进制数1100.101，则
 - $1100.101 = 0.1100101 \times 2^{100}$
 - 尾数 $S=0.1100101$ ，阶码 $j=100$

这一章我们学到了



- 数据类型
 - 每一种类型的数据都是有范围的
 - 实数是不精确的
- 常量的使用
- 如何定义一个变量
- 技巧：
 - 程序总体上由输入、处理、输出三部分组成
 - 交换两个变量通常要借助于第三变量
 - 大小写字母的转换
 - 数字与数字字符的转换
 - sizeof()

- MOOC单元测验
- 书上习题一、二、三、四(1、2)。
- 预习第三章的内容。
- 记住0-16的十六进制、二进制编码。
- 记住'\n'、'\'、'\\"、'\\"、'\t'等常用转义字符。
- 记住空格、'0'、'A'、'a'的ASCII码。