# 变量的内存分析

## 字节和地址

为了更好地理解变量在内存中的存储细节，先来认识一下内存中的“字节”和“地址”。

1. 内存以“字节为单位”

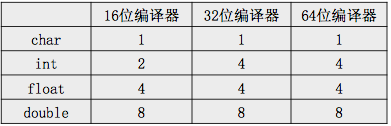


0x表示的是十六进制，不用过于纠结，能看懂这些数字之间谁大谁小就行了

1. 不同类型占用的字节是不一样的，数据越大，所需的字节数就越多

## 变量的存储

1. 所占用字节数跟类型有关，也跟编译器环境有关



1. 变量实例

int b = 10;

int a = 20;

* 内存由大到小寻址，优先分配内存地址较大的字节给变量。b的内存地址比a大
* 每个变量都有地址：第一个字节的地址就是变量的地址

1. 查看内存地址：

int a;

printf("a的地址是：%p\n", &a);

1. 注意

在变量未经初始化之前，不要尝试使用变量的值

int a;

printf("a的值是：%d\n", a);

上面的写法是不建议的

# scanf函数

## 简介

这也是在stdio.h中声明的一个函数，因此使用前必须加入#include <stdio.h>。调用scanf函数时，需要传入变量的地址作为参数，scanf函数会等待标准输入设备（比如键盘）输入数据，并且将输入的数据赋值给地址对应的变量

## 简单用法

int age;

scanf("%d", &age);

* scanf函数时，会等待用户的键盘输入，并不会往后执行代码。scanf的第1个参数是"%d"，说明要求用户以10进制的形式输入一个整数。这里要注意，scanf的第2个参数传递的不是age变量，而是age变量的地址&age，&是C语言中的一个地址运算符，可以用来获取变量的地址
* 输入完毕后，敲一下回车键，目的是告诉scanf函数我们已经输入完毕了，scanf函数会将输入的值赋值给age变量

## 其他用法

1. 用scanf函数接收3个数值，每个数值之间用中划线-隔开

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

3个%d之间是用中划线-隔开的，因此我们在每输入一个整数后都必须加个中划线-，比如这样输入http://images.cnitblog.com/blog/497279/201303/14202751-fe5d9b5d610c45cf868ec84b489d7e45.png，不然在给变量赋值的时候会出问题

* 注意：数值之间的分隔符是任意的，不一定要用中划线-，可以是逗号、空格、星号\*、井号#等等，甚至是英文字母

// 逗号,

scanf("%d,%d,%d", &a, &b, &c); // 输入格式：10,14,20

// 井号#

scanf("%d#%d#%d", &a, &b, &c); // 输入格式：10#14#20

// 字母x

scanf("%dx%dx%d", &a, &b, &c); // 输入格式：10x14x20

1. 用scanf函数接收3个数值，每个数值之间用空格隔开

scanf("%d %d %d", &a, &b, &c);

3个%d之间是用空格隔开的，我们在每输入一个整数后必须输入一个分隔符，分隔符可以是空格、tab、回车

## 注意

scanf的第一个参数中不要包含\n，比如scanf(“%d\n”, &a); 这将导致scanf函数无法结束

## 习题

提示用户输入两个整数n，然后计算并输出两个整数的和