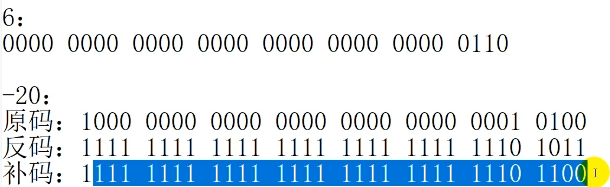
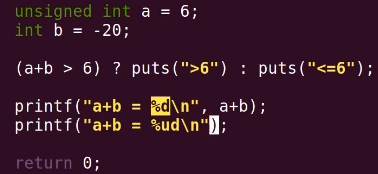
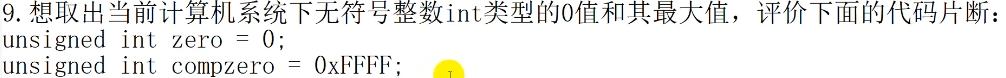


“AAA”放在只读区域，只是s指针指向该区域，因此无法对其进行更改

无符号整形和有符号整形相加时，有符号整形自动转化为无符号整形

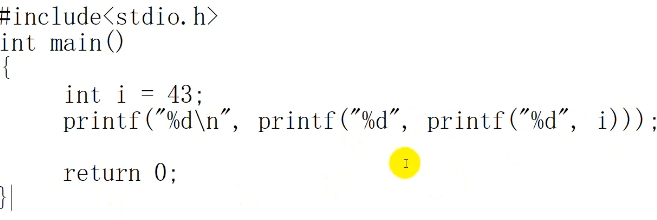


%d以有符号整数输出



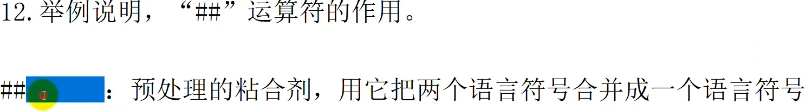
正确写法：

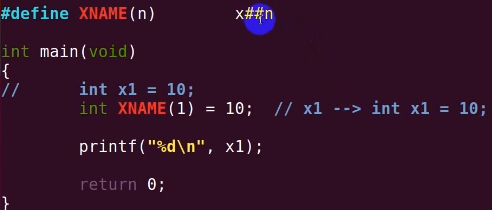
Printf返回值是你打印字符的个数

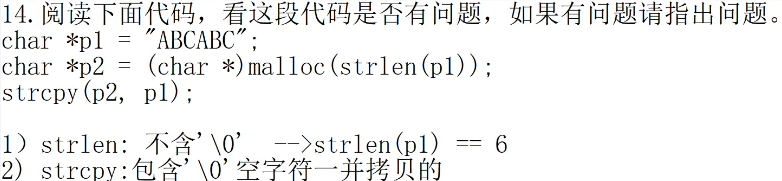


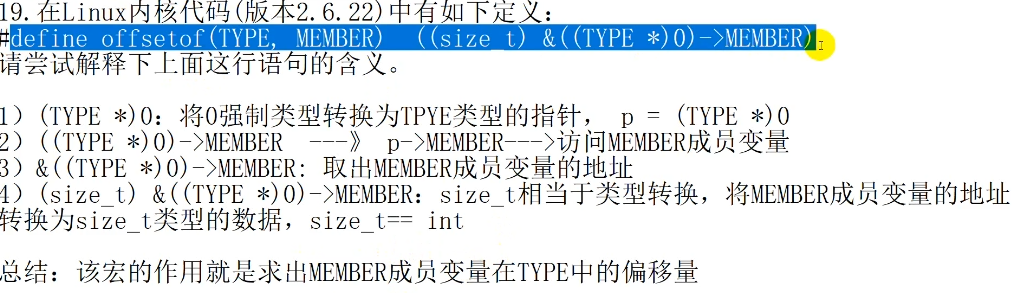
输出结果是4321

一开始打印43时，打印了两个字符



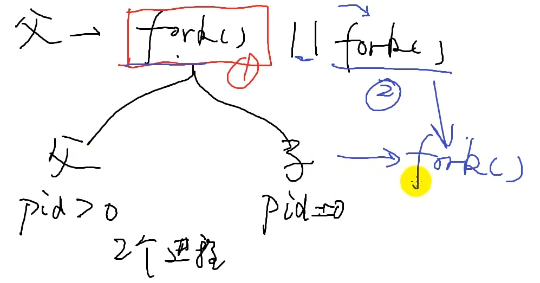


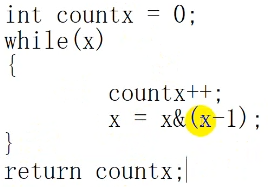




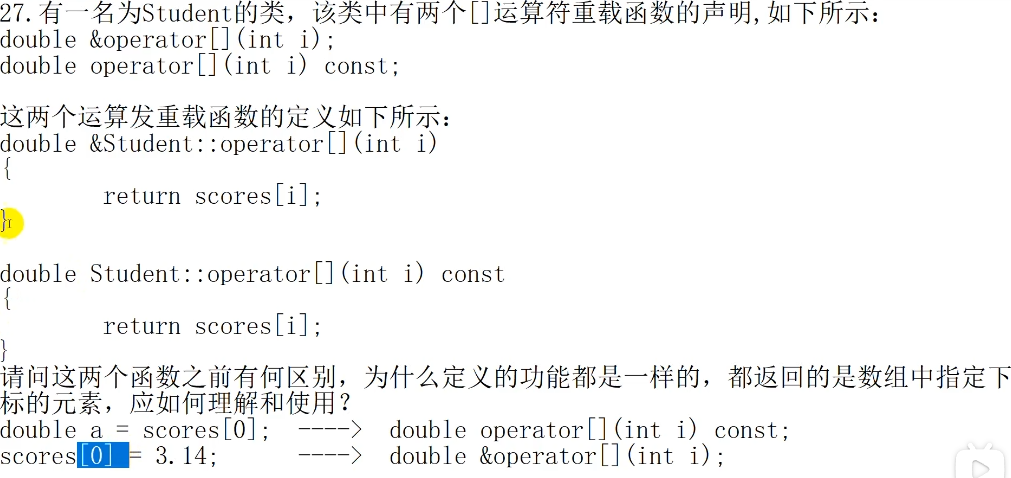


3个进程，fork函数，对子进程返回0，父进程返回1，因此父进程的第二个fork不会进行判定



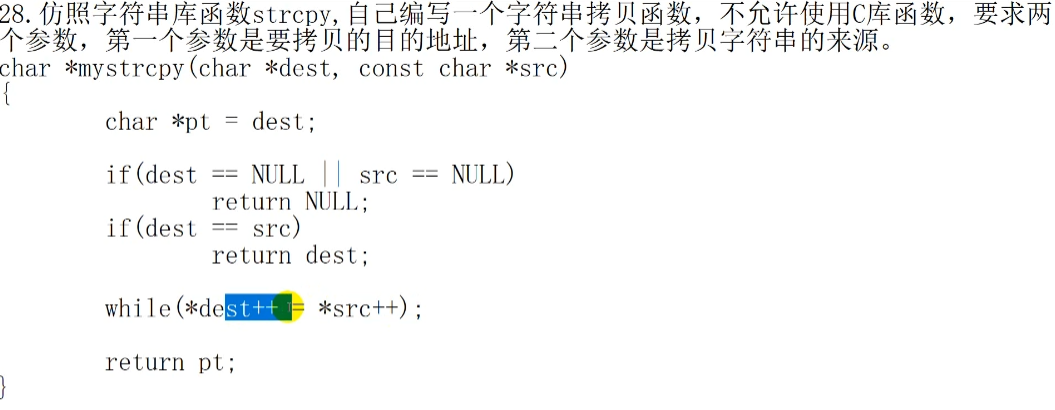
x转换成二进制之后其中“1”的个数

判定一个数是否是2的n次方？ x&(x-1)==0,则这个数是2的n次方



Double &operator[](int i); 数组的值可以被修改，用于左值

Double operator[](int i) const; 数组的值不可以被修改，用于右值

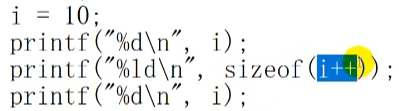


\*a=\*b;//将\*b赋予给\*a,结果是\*a=B,而该结果B成为while的条件，也即是只要B不为0(更准确来说是B转为二进制时不为00…0)，条件就为真(非0)，可以循环

位操作： 给a的bit3置1，给a的第bit3置0

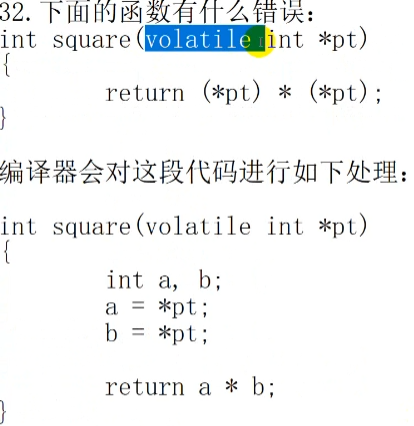
置1：a=a|(1<<3)

置0：a=a&~(1<<3)



sizeof是一个运算符，只判断括号里面是什么类型，并不进行运算，因此i并不会++；第三个printg打印的还是10

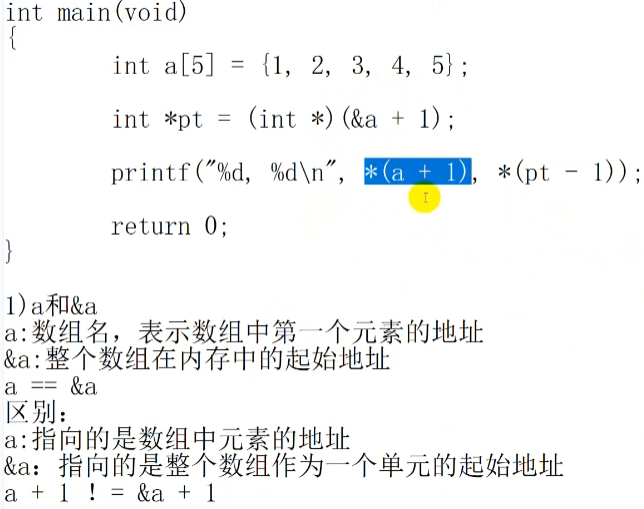
Volatile

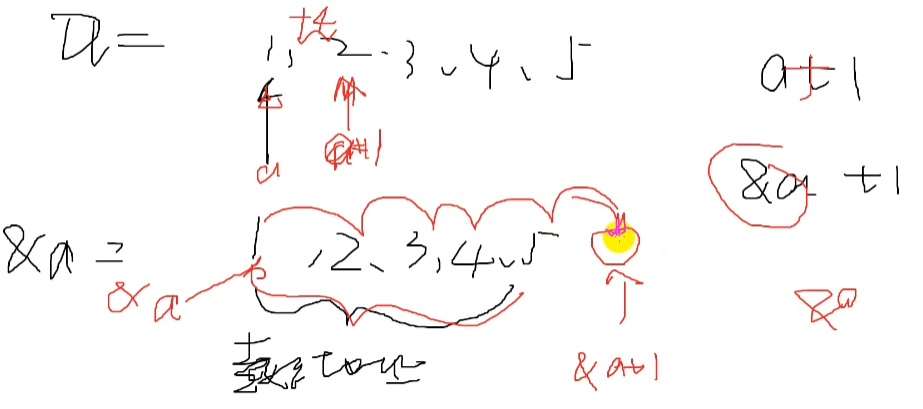
pt指向的内容随时会改变，a可能不等于b

因此为了实现平方的功能，应该如下操作

a=\*pt; return a\*a;

Sizeof()检测空的类的话，结果是1





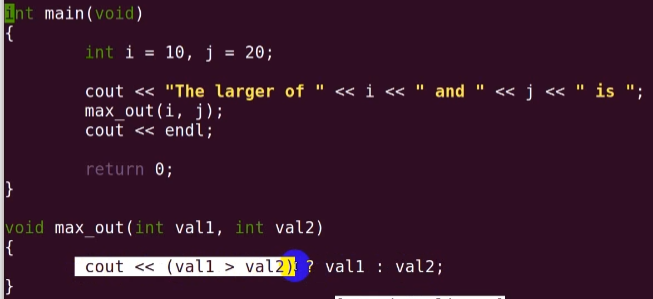
因此结果输出是2和5，pt最终指向数组的下一个位置（类型转换int\*），pt-1后就变成回去上一个位置

在不使用第三个变量的情况下，交换a,b两个变量的值

a=a+b;

b=a-b;

a=a-b;



会先进行cout << (val1>val2),因为<<优先级比“？”高

高四位和第四位互换

