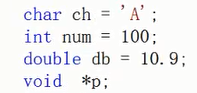
第29讲 5.3.632- 5.3.63 Void指针与空指针--详细讲解

1、空类型指针可以指向任何类型的数据，包含他们的地址。



p=&ch;

p=&num;

p=&db;

printf(“%d”,\*p); //\*p不明确从地址开始，前进几个字节，所以出错

\*((double \*)p)=20.8 // 明确了从地址开始，前进了几个字节

2、任何指针都可以赋值给空类型的指针，用于保存地址

3、memset函数

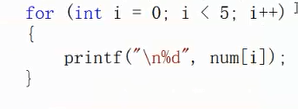
从数组str的首地址开始，前进 5个字节，进行赋值，赋值一个字符 ’ A ’ 。

memset(str, ’A’ ,5) ;

对20个字节全部赋值为0，对于数组清零 （num是数组）

memset(num, 0 ,20);

遍历数组num的所以元素，元素值都为0。



4、malloc函数

malloc的全称是memory allocation，中文叫动态内存分配，当无法知道内存具体位置的时候，想要绑定真正的内存空间，就需要用到动态的分配内存。原型为extern void \*malloc(unsigned int num\_bytes)。

malloc(1024\*1024\*1024) //分配内存函数，100M

void \*p=malloc(20); //20个字节的内存

int \* px=(int \*)p; //空类型指针可以转换为任何类型的指针

5、空类型指针可以转换为任何类型的指针

一个类型的指针包含三个信息:

地址、 步长、内容如何解析