指针—1.基本概念

如左图。

0x01a0

0x01a1

0x01a2

0x01a3

0x01a4

0x01a5

0x01a6

0x01a7

0x01a8

0x01a9

0x01aa

0x01ab

。。。

。。。

10100100

00000001

00000000

00000000

01100001

00000001

01110000

11000000

。。。

假设右图是内存中一段空间，地址从0x01a0-0x01ab共12个字节。

假设字节0x01a0-0x01a3空间分配给了变量p，

从左图可看出，p内存储值为0x1a4。

若p是int型，则p的值是0x1a4，表示一个整数，按照十进制输出为420。

若p为int \*型，则p的值是0x1a4，表示是个整型变量的地址。p指向0x1a4-0x1a7这四个字节。

若p为char \*型，则p的值是0x1a4，表示是个char型变量的地址。p指向0x1a4这个字节。

若p为short \*型，则p的值是0x1a4，表示是个短整型变量的地址。p指向0x1a4-0x1a5这两个字节。

若p为unsigned int \*型，则p的值是0x1a4，表示是个无符号整型变量的地址。p指向0x1a4-0x1a7这四个字节，这四个字节内的数据将被当做无符号整型处理。

若p为float \*型，则p的值是0x1a4，表示是个单精度变量的地址。p指向0x1a4-0x1a7这四个字节，这四个字节内的数据将被当做单精度实数处理。

若p为int \*\*型，则p的值是0x1a4，表示是个整型指针变量的地址，p指向0x1a4-0x1a7这四个字节，这四个字节内的数据将被当做一个整型变量的地址来处理。

若p为int \*\*\*型，则p的值是0x1a4，表示是个int\*\*型指针变量的地址，p指向0x1a4-0x1a7这四个字节，这四个字节内的数据将被当做int\*\*型变量的地址来处理。

依次类推。。。

注意：上述所有指针指向的起始地址由p变量的值决定，指向的字节范围由指针变量的类型决定。

**从上面表述可以看出：**

**1、指针变量也是个变量，在内存有存储空间（一个字长，和整型相同），有变量名。**

**2、指针变量内存储的数据被当做地址处理。**

**3、此地址是一段内存空间的起始地址，至于空间所占长度和空间内数据如何被解释，由指针变量的类型决定。**

**4、不管p的类型被如何转换，只要p没有被重新赋值，p所指向的起始地址就不会变，变化的只是从该起始地址开始寻址的范围和对范围内数据的不同解释而已。**

所以：任何指针变量，都是一个字长。对指针变量最重要的有两点：

1. 该指针变量内存储的值，决定了它指向范围的起始地址，没有这个值或者这个值无效就是个野指针了。
2. 指针变量所指向的类型，在根据指针变量内存储值找到起始地址后，需要由指针变量的类型来决定多大的范围有效，否则依然无法确定所指向变量。

如上图所示：

p=0x1a4.则：

\*(char \*)p=’a’;

\*(short \*)p=353;

\*(int \*)p=’a’;

\*(int \*)p=-1066401439;

\*(unsigned int \*)p=3228565857;

\*(float \*)p=-3.750084

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

练习1：请编程验证上述结果。

自己完成。

练习2：完成一个函数，该函数输入一个变量，在标准输出显示该变量的二进制形式。

参考代码：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

函 数 名：ToBinDisplay

函数功能：将输入的变量按二进制形式输出到标准输出

输入参数：需要处理的变量

输出参数：无

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define ToBinDisplay(data) ToBinDisplayfunc(&(data),sizeof(data))

void ToBinDisplayfunc(const void \*buf,int n)

{

int i,j;

const char \*p=(const char \*)buf;

for(j=0;j<n;j++)//从低字节到高字节输出

{

printf("%x:",p+j);

for(i=7;i>=0;i--)

{

printf("%d ",((\*(p+j))>>i)&1);

}

printf("\n");

}

}

练习3：使得程序从指定地址处开始执行。

提示：使用函数指针。

参考代码：

void (\*p)(void)

p=ADDR; //ADDR为一指定地址

p(); //此句开始，程序将从ADDR所指地址处开始执行。