指针

指针及指针变量

内存单元的地址

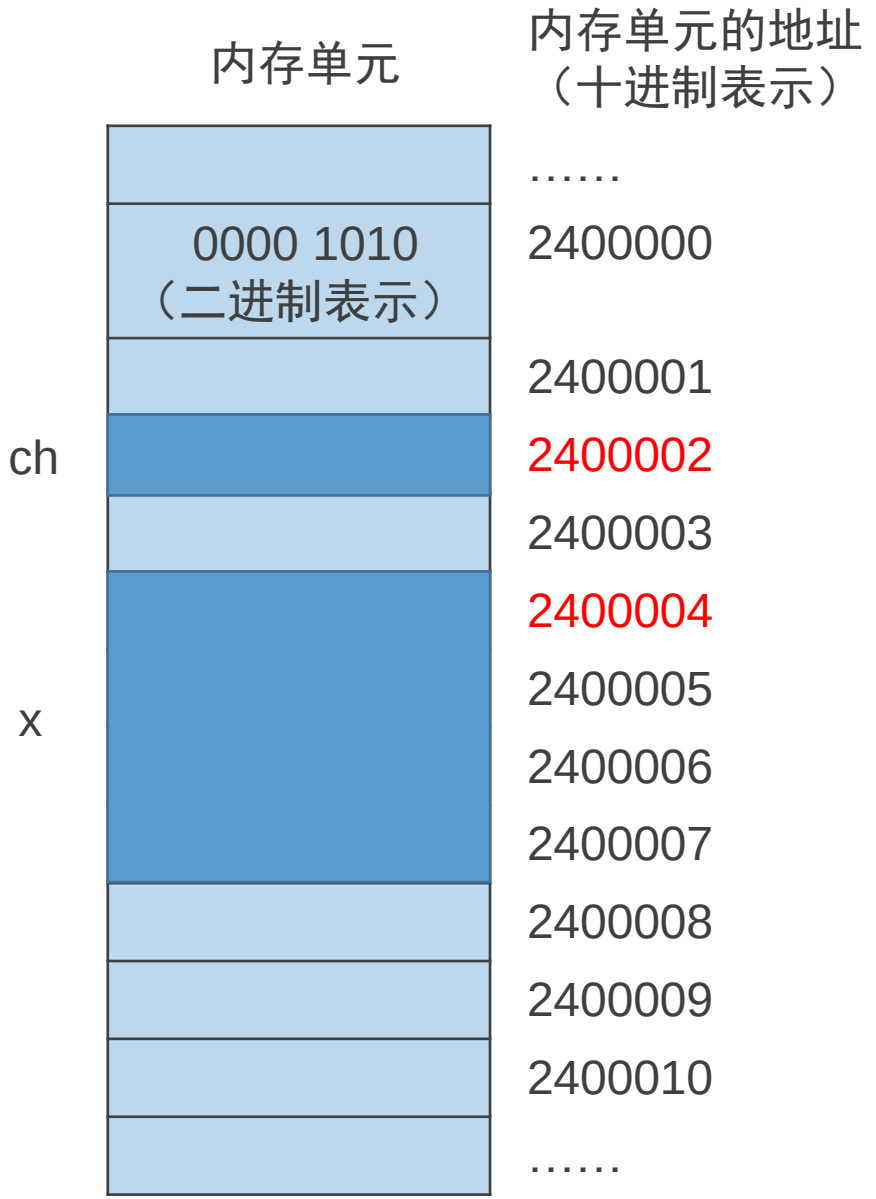
一个内存单元的大小是1个字节，含有8个二进制位

不同的编译系统给各类变量所分配的存储空间大小不一样，例如：

```
char  ch;
int   x;
```

若以vc++6.0为编译环境，则给 ch分配1个字节，给x分配4个字节，如右图所示：

变量的地址就是变量所占用的内存空间的首地址，称作变量的指针



查看变量的地址

变量的存储空间是由系统分配的，可以使用取地址运算符 & 得到变量的地址

```
#include"stdio.h"

int main( )

{  int x=12;

   printf("变量x的存储空间的起始地址:%d\n", &x);

   return 0;

 }
```

运行结果如下：

```
变量x的存储空间的起始地址:2424396
```

不同的对象所分配的存储空间在不同的内存区域

<table>
<tr><td>系统存储区</td><td colspan="3">操作系统（如 windows）、语言系统（如 Visual C++）</td></tr>
<tr><td rowspan="4">用户存储区</td><td colspan="3">程序区（C 程序代码）
如主函数、函数 complex_add() 、函数 complex_prod()等</td></tr>
<tr><td rowspan="3">数据区</td><td rowspan="2">静态存储区</td><td>全局变量
如 result_real, result_imag</td></tr>
<tr><td>静态局部变量</td></tr>
<tr><td rowspan="3">动态存储区
（如自动变量）</td><td>main()变量区：
real1,imag1,real2,imag2</td></tr>
<tr><td>complex_add()变量区：
real1,imag1,real2,imag2</td></tr>
<tr><td>……</td></tr>
</table>

存取变量值的方式

- 直接访问
- 间接访问

直接访问

按变量地址和变量名直接存取变量的值，这种方式称为“直接访问”，例如：

```
#include"stdio.h"
int main()
{  int x ;
   scanf("%d",&x);
   printf("x=%d\n",x);
   return 0;
 }
```

间接访问

把变量x的地址存放在一个专门用来存放地址信息的变量px中，通过px来存取变量x的值，这种方式称为“间接访问”，例如：

```
#include"stdio.h"
 int main()
{  int  x ;
   int  *px;
   px=&x;
   scanf("%d", px);
   printf("x=%d\n", *px);
   return 0;  }
```

用来存放一个变量的地址信息的变量，称为“**指针变量**”。

定义一个指针变量

定义指针变量的一般形式为

基类型　*指针变量名；

指针变量名前面的 * ，表示该变量的类型为指针型变量，不属于变量名部分，
例如：

```
int   x, y;        //整型变量x和y
int  *px, *py;     //指针变量px和py
px=&x;             //把x的地址保存在px里
py=&y;             //把y的地址保存在py里
```

因为不同类型的变量所占用的存储空间大小不一样，变量的指针就有所区别，因此指针变量也是有类型的，这就是用来定义指针变量的基类型，例如：

```
int   x ;        //整型变量x
char  ch;        //字符型变量ch
double  s;       //浮点型变量s

int  *p1 ;       //整型指针变量p1
char  *p1;       //字符型指针变量p2
double  *p3;     //浮点型指针变量p3

p1=&x;           //把x的地址保存在p1里
p2=&ch;          //把ch的地址保存在p2里
p3=&s;           //把s的地址保存在p3里
```

注意：指针变量里保存的地址所对应的变量类型要和相应的指针变量的基类型一致

THANKYOU

陕西科技大学
SHAANXI UNIVERSITY OF SCIENCE & TECHNOLOGY