中南大學
CENTRAL SOUTH UNIVERSITY

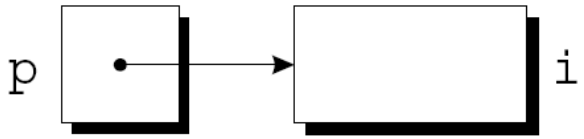
第十一讲

指　　针

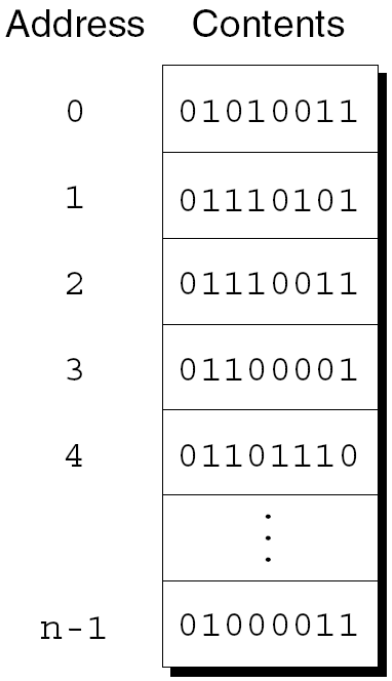
11、指针

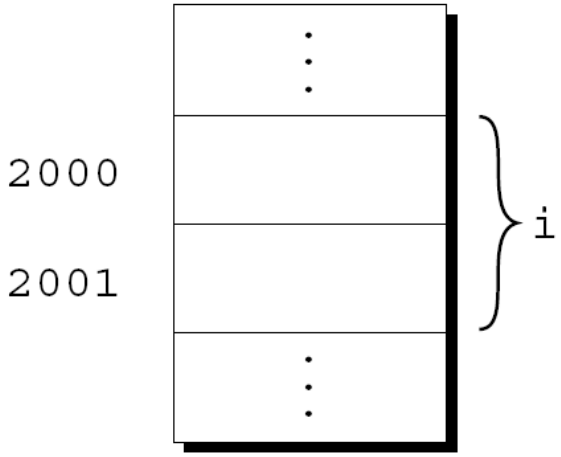
11.1 指针变量

- 地址：内存单元编号
- 每个字节（Byte）有一个唯一的地址
- 变量地址：变量的第一个字节的地址
- 指针变量（Pointer Variable）用来存储地址



p 指向 i





11.1 指针变量

指针变量的声明

```
int *p
```

p 是指向int类型对象的指针变量

可以和其他变量声明一起出现

```
int i, j, a[10], b[20], *p, *q;
```

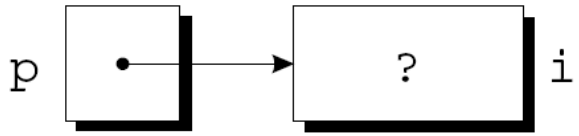
指针变量只能指向一种特定类型

```
int *p;     /* points only to integers   */
double *q;  /* points only to doubles    */
char *r;    /* points only to characters */
```

11.2取地址运算符和间接寻址运算符

- 取地址运算符&

```
int i, *p;
...
p = &i;
```



scanf（" %d",&i）；　　scanf（" %d",**p**）；

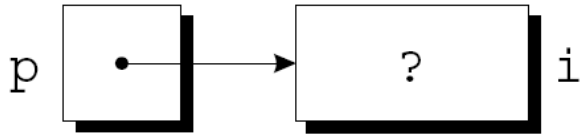
- 间接寻址运算符
  - *p：访问p所指向的变量的内容

p指向i，*p就是i的别名

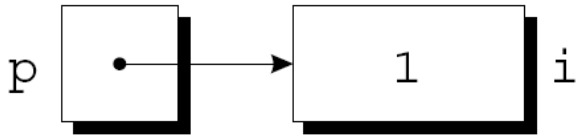
```
printf("%d\n", *p);
j = *&i;   /* same as j = i; */
```

11.2取地址运算符和间接寻址运算符

```
p = &i;
```

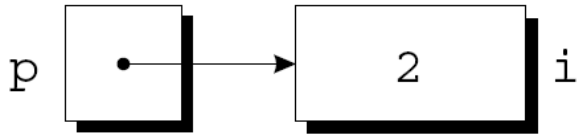



```
i = 1;
```




```
printf("%d\n", i);     /* prints 1 */
printf("%d\n", *p);    /* prints 1 */
```

```
*p = 2;
```

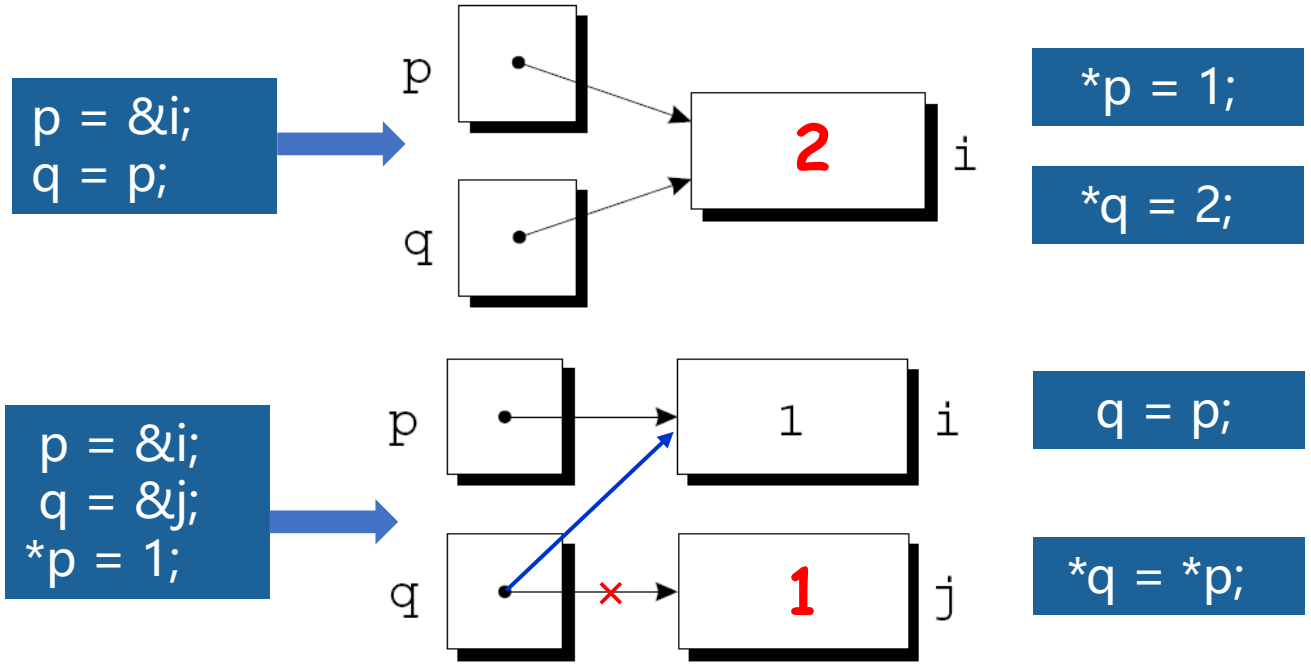



```
printf("%d\n", i);     /* prints 2 */
printf("%d\n", *p);    /* prints 2 */
```

11.3 指针赋值

- 使用赋值运算进行指针的复制，前提是两个指针具有相同的类型

```
int i ,j, *p,*q;
```



11.4 指针作为参数

- 例题9-6返回实数x的整数部分和小数部分

函数定义

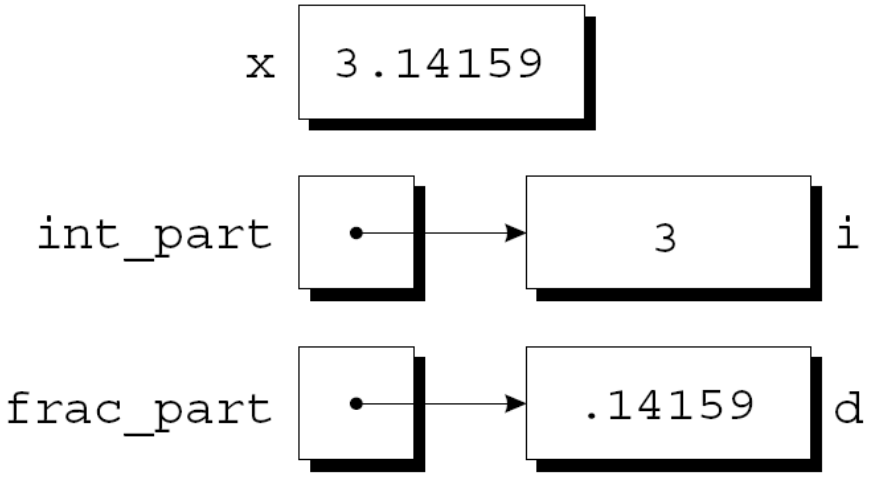
```
void decompose(double x, long int_part, double frac_part)
{
        int_part = (long) x;
        frac_part = x - int_part;
}
```

函数调用

```
decompose(3.14159, i, d);
```

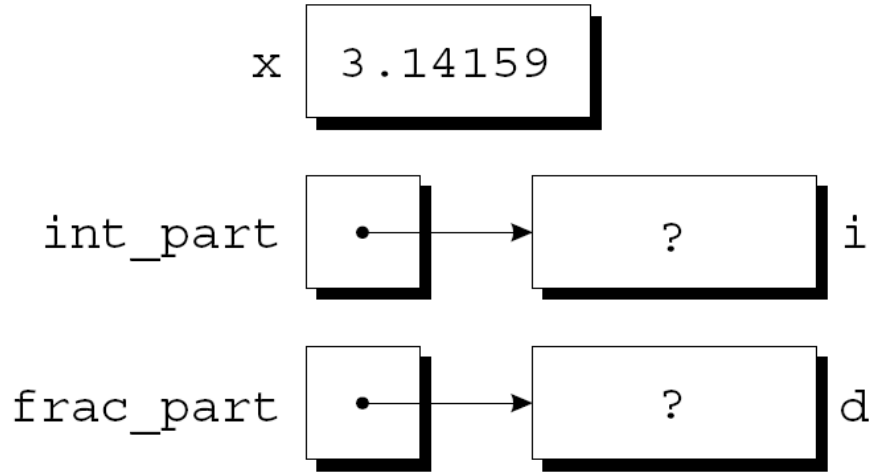
11.4 指针作为参数

- 例题9-6返回实数x的整数部分和小数部分



函数定义

```
void decompose(double x, long *int_part,double *frac_part)
{
      *int_part = (long) x;
      *frac_part = x - *int_part;
}
```



函数调用

```
decompose(3.14159, &i, &d);
```

11.4 指针作为参数

- 示例：找出数组中最大元素和最小元素

```
Enter 10 numbers: 34 82 49 102 7 94 23 11 50 31
Largest: 102
Smallest: 7
```

```
void max_min(int a[], int n, int *max, int *min);

max_min(b, N, &big, &small);
```

11.4 指针作为参数

- 示例：找出数组中最大元素和最小元素

```
void max_min(int a[], int n, int *max, int *min);
int main(void){
   int b[N], i, big, small;
   printf("Enter %d numbers: ", N);
   for (i = 0; i < N; i++)
      scanf("%d", &b[i]);

   max_min(b, N, &big, &small);

   printf("Largest: %d\n", big);
   printf("Smallest: %d\n", small);
   return 0;
}
```

11.4 指针作为参数

- 示例：找出数组中最大元素和最小元素

```
void max_min(int a[], int n, int *max, int *min){
   int i;
   *max = *min = a[0];
   for (i = 1; i < n; i++) {
      if (a[i] > *max)
         *max = a[i];
      else if (a[i] < *min)
         *min = a[i];
   }
}
```

11.4 指针作为参数

- 用const保护参数
  - const表明函数不会改变指针参数所指向的对象
  - 应该放在形式参数声明中

```
void f(const int *p)     {

    *p = 0;     /*** WRONG ***/
}
```

11.5 指针作为返回值

```
int *max(int *a, int *b){
  if (*a > *b)
    return a;
  else
    return b;
}
```

- 函数可以返回外部变量或声明为static的静态变量的指针
- 不能返回局部指针

```
int *f(void){
  int i;
  ...
  return &i;
}
```