

指针与函数

- 指针变量的引入有很多作用，其中有一个很重要的作用：就是用指针变量作为函数参数。

指针作为函数参数

例: 用函数实现交换2个变量的值

```
#include <stdio.h>
```

```
void swap(int a, int b)
```

```
{ int t;
```

```
    t=a; a=b; b=t;
```

```
    /*printf("交换后: %d,%d\n", a, b); */
```

```
}
```

```
int main(void)
```

```
{ int x=5, y=10;
```

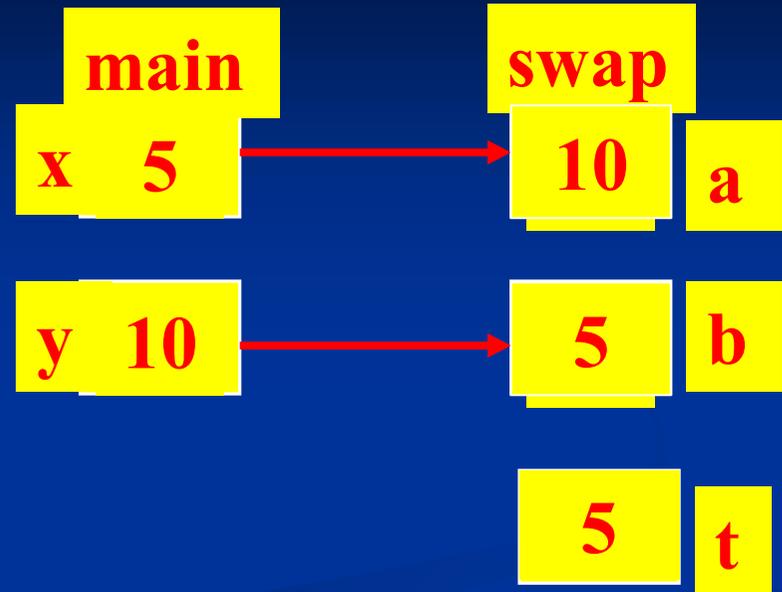
```
    printf("交换前: %d, %d\n", x,y);
```

```
    swap(x, y);
```

```
    printf("结束后: %d, %d\n", x, y);
```

```
    return 0;
```

```
}
```



输出结果:

交换前: 5, 10

结束后: 5, 10

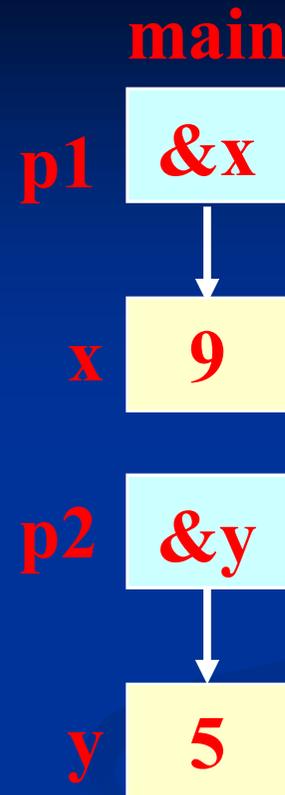
```
#include <stdio.h>
```

```
void swap(int *a, int *b)
```

```
{ int t;  
  t=*a; *a=*b; *b=t;  
}
```

```
int main(void)
```

```
{ int x=5, y=9;  
  int *p1, *p2;  
  p1 = &x; p2 = &y;  
  printf("交换前: %d, %d\n", x,y);  
  swap(p1, p2);  
  printf("结束后: %d, %d\n", x, y);  
  return 0;  
}
```



输出结果:
交换前: 5, 9
结束后: 9, 5

例：交换两个数，用指针。

```
#include <stdio.h>
void swap(int *a, int *b)
{ int t;
  t=*a; *a=*b; *b=t;
}
```

改成

```
#include <stdio.h>
void swap(int *a, int *b)
{ int *t;
  t=a; a=b; b=t;
}
```

```
int main(void)
{ int x=5, y=9;
  printf("交换前: %d, %d\n", x,y);
  swap(&x, &y);
  printf("结束后: %d, %d\n", x, y);
  return 0;
}
```

什么结果？

■ 如何通过函数调用达到改变某个变量的值？

分三步：

- 1、在主调函数中，把该变量的地址作为实参
- 2、在被调函数中，用形参（指针）接受该地址
- 3、在被调函数中，改变形参（指针）所指向变量的值，而不是形参本身的值（地址）

例:求三个数的最大值和平均值，用函数实现。

```
#include <stdio.h>

void mav (int a, int b, int c, int *p1, float *p2)
{
    *p1 = a;
    if(*p1 < b) *p1 = b;
    if(*p1 < c) *p1 = c;
    *p2 = (a+b+c)/3.0;
}

int main(void)
{
    int a, b, c, max, *p1;
    float ave, *p2;
    scanf("%d%d%d",&a,&b,&c);
    p1 = &max; p2 = &ave;
    mav(a, b, c, p1, p2);
    printf("max=%d, ave=%f",max,ave) ;
    return 0;
}
```

返回指针值的函数

函数的返回值可以是一个指针类型数据(即地址)

返回指针值函数的定义格式:

```
数据类型 *函数名(形参列表)  
{ 函数体 ; }
```

说明:

定义一个返回指针值的函数与定义普通函数的格式基本类似,只是在函数名前加 *****,表明该函数返回一个指针值

```
例: int *fun ( int a , int b )  
    { 函数体 ; }
```

例：利用指针函数求两个数中的最大值。

```
#include <stdio.h>
```

```
int *maxp(int *, int *);
```

```
int main(void)
```

```
{ int x, y, *p;
```

```
scanf("%d%d", &x, &y);
```

```
p=maxp(&x, &y);
```

```
printf("\nmaxp=%d", *p);
```

```
return 0;
```

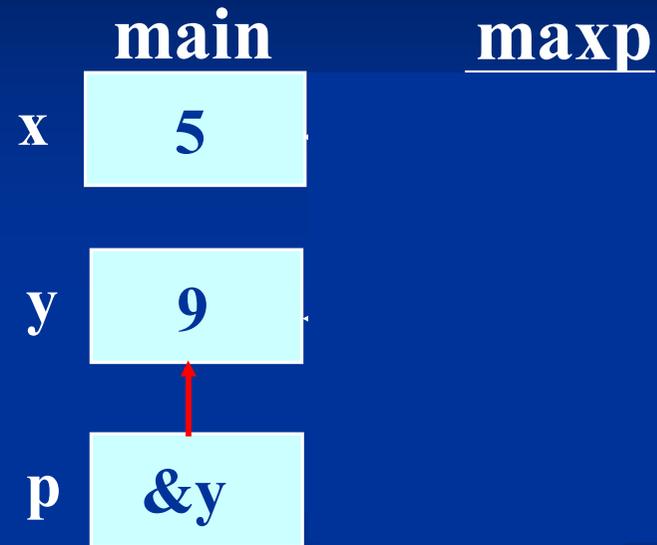
```
}
```

```
int *maxp(int *a, int *b)
```

```
{
```

```
return *a>*b? a:b;
```

```
}
```



指向函数的指针变量

- 一个函数占用一段连续内存空间，而函数名和该区域首地址对应。这样就可以把函数的首地址赋给一个指针变量，使其指向该函数。这种指向函数的指针变量称为函数指针变量。
- 一般形式：
数据类型 (*指针变量名)(函数形参列表)
- 例：

```
int func(int x); //声明一个函数
int (*f)(int x); //声明一个指向函数的指针变量
f = func; //将func()函数的首地址赋给指针f
```
- 调用时，可用如下形式：

```
int a; a = (*f)(25); 或： a = f(5); 它和 a = func(5); 等价
```
- 常见用途：转换表和作为参数传递给另一个函数

小结

- 函数的定义
- 函数的调用
- 函数的声明
- 全局变量和局部变量
- 指针的概念
- 指针的基本操作
- 指针作为函数参数

Class is over