

## 6.函数指针数组

用途--->转移表

```
1  int add(int x, int y);
2  int add2(int x, int y);
3  int add3(int x, int y);
4  int add4(int x, int y);
5
6  int main()
7  {
8      //指针数组
9      int* arr[5];
10     int (*pa)(int, int) = add;
11     int (*pa[4])(int, int) = {add, add2, add3, add4};
12
13     //调用
14     printf("%d", pa[1](2, 3));
15     return 0;
16 }
17
18 char* my_strcpy(char* dest, const char* src);
19 1. 写一个函数指针pf能够指向my_strcpy
20 char* (*pf)(char*, const char*);
21 2. 写一个函数指针数组pfarr 能够存放4个my_strcpy函数的地址
22 char* (*pfarr[4])(char*, const char*);
23
```

使用案例:

计算器(简略)

```
1  int add(int x, int y);
2  int sub(int x, int y);
3  int mul(int x, int y);
4  int div(int x, int y);
5
6  int main()
7  {
8      int input = 0;
9      int x = 0;
10     int y = 0;
11     int (*pf[4])(int, int) = {0, add, sub, mul, div};
12     do{
```

```

13  printf("请选择");
14  scanf("%d", *input);
15  printf("%d%d", &x, &y);
16  int ret = pf[input](x, y);
17  printf(ret);
18  }while(input);
19  }
20

```

## 7.指向函数指针数组的指针

指向函数指针数组的指针是一个指针 指针指向一个数组 数组的元素都是函数指针

```

1  int main()
2  {
3      int arr[10] = {0};
4      int (*p)[10] = &arr; //取出数组的地址
5      int (*pp)(int, int);
6      int (*pfarr[4])(int, int); //pfarr是一个数组 函数指针的数组
7      int (*(ppfarr)[4])(int, int) = &pfarr;
8      //ppfarr是一个数组指针 指针指向的数组有四个元素
9      // 指向的数组的每个元素的类型是一个函数指针
10
11     return 0;
12
13 }

```

## 8.回调函数

通过函数指针调用的函数 如果把函数的地址作为参数传递给另一个函数 当这个指针被用来调用其他所指的函数时 我们就说这是回调函数，回调函数不是由该函数的实现方直接调用 而是在特定的时间或者条件发生时由另外的一方调用的 用于该事件或条件进行响应

## 9.指针和数组面试题解析