

趣味指针分析

/* 2352018 大数据 刘彦 */

1. 对指针的分析

```
void>(*f[3])(const char* p)(int x)
```

`f` 是一个数组，可以存储 3 个指向特定函数的指针。`(*f[3])` 表示 `f` 是一个数组，数组的元素是指向函数的指针。

`(const char* p)` 指定了这些函数接受一个 `const char*` 类型的参数。这意味着调用这些函数时，必须传入一个字符串（字符数组）的指针。

`(*)(int x)` 表示这些函数返回一个指向另一个函数的指针，而这个函数的参数是 `int` 类型。

总的来说：

`f` 是一个包含 3 个元素的数组。每个元素都是指向一个函数的指针，该函数接受一个 `const char*` 类型的参数。这些函数返回一个指向另外一个接受 `int` 类型参数并返回 `void` 的函数的指针。

2. 原代码

```
/* 2352018 大数据 刘彦 */
```

```
#include <iostream>
```

```
using namespace std;
```

```
void f_sub(int x)
```

```
{
```

```
    cout << "x = " << x << endl;
```

```
}
```

```
void (*f_func(const char* p))(int x)
```

```
{
```

```
    cout << "str = " << p << endl;
```

```
    return f_sub;
```

```
}
```

```
int main()
```

```
{
```

```
    void(*f[3])(const char* p)(int x) = { f_func, nullptr, nullptr };
```

```
    f[0]("Hello")(2);
```

```
    f[0]("Tongji")(7);
```

```
    return 0;  
}
```

3. 解释

- ① f_sub, 接收一个整数参数 x, 并打印其值。
- ② f_func 函数接受一个 const char* 类型的字符串参数 p, 并打印字符串 p。最后返回一个指向 f_sub 函数的指针, 类型为 void(*)(int)。
- ③ main 函数中
 - a) 初始化 f 数组的第一个元素为 f_func, 其余两个元素为空指针常量 nullptr。
 - b) f[0]("Hello") 调用 f_func, 传入字符串 "Hello", 打印 str = Hello 然后返回指向 f_sub 的指针, 调用返回的 f_sub, 输出 x = 2。
 - c) f[0]("Tongji") 再次调用 f_func, 传入字符串 "Tongji", 打印 str = Tongji。然后返回指向 f_sub 的指针。调用返回的 f_sub, 输出 x = 7。