

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

E04_a

从框架看HAL和 Linux驱动开发(a)

By 高煥堂

内容

1. 复习：函数表的定义与实例化
2. Linux驱动框架的函数表
3. 活用工厂EIT造形
4. 撰写用户态的应用程序

1、复习：函数表的 定义与实例化

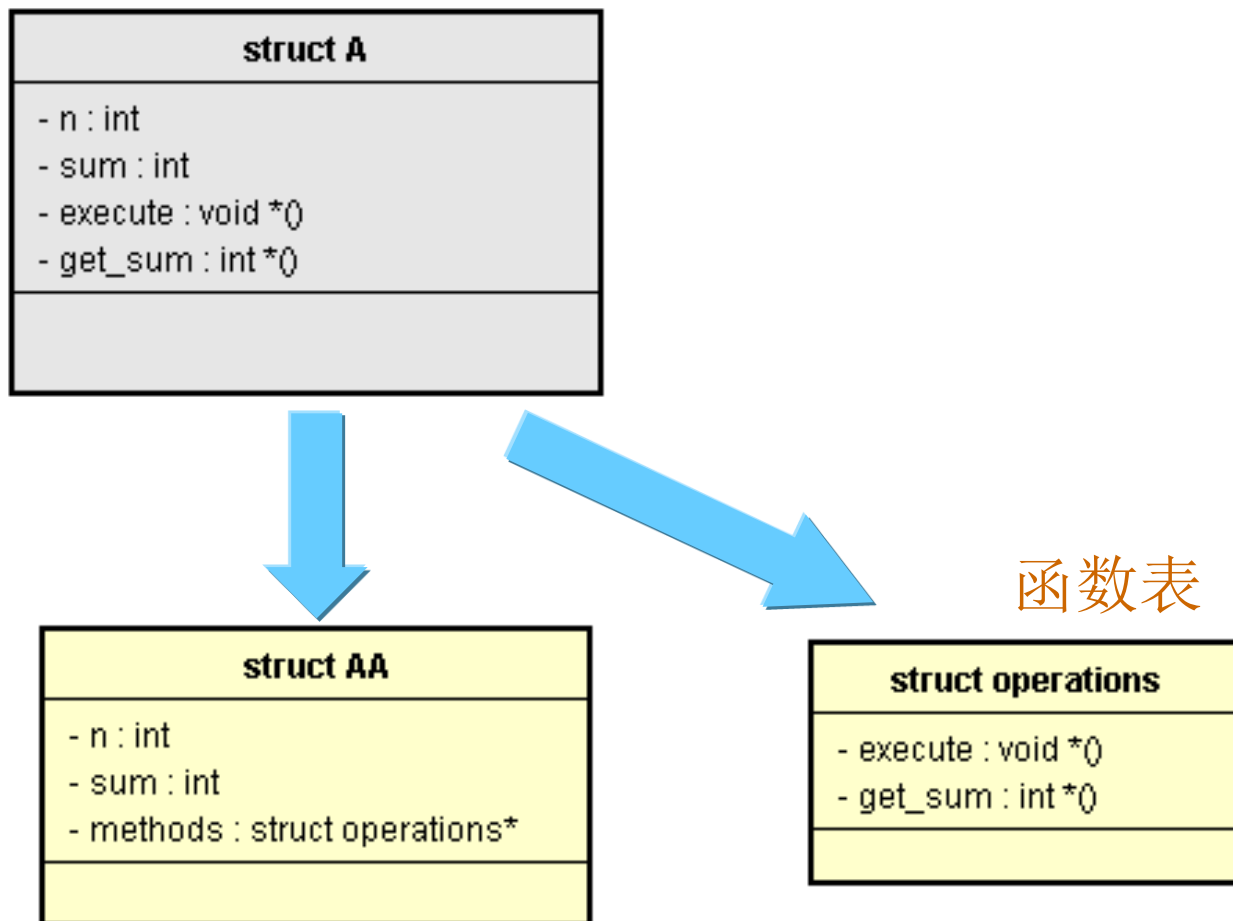
Function Table

C的函数表(function)概念

- 从一个C的struct谈起

struct A
- n : int - sum : int - execute : void *() - get_sum : int *()

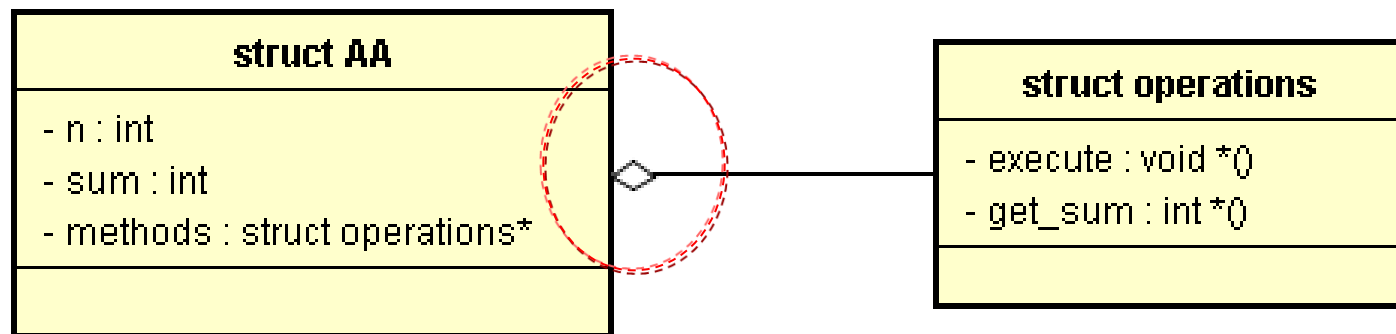
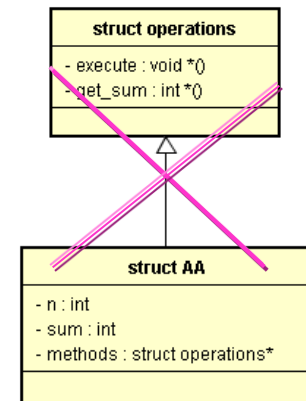
- 把函数部分独立出来，成为一个函数表(function table)。



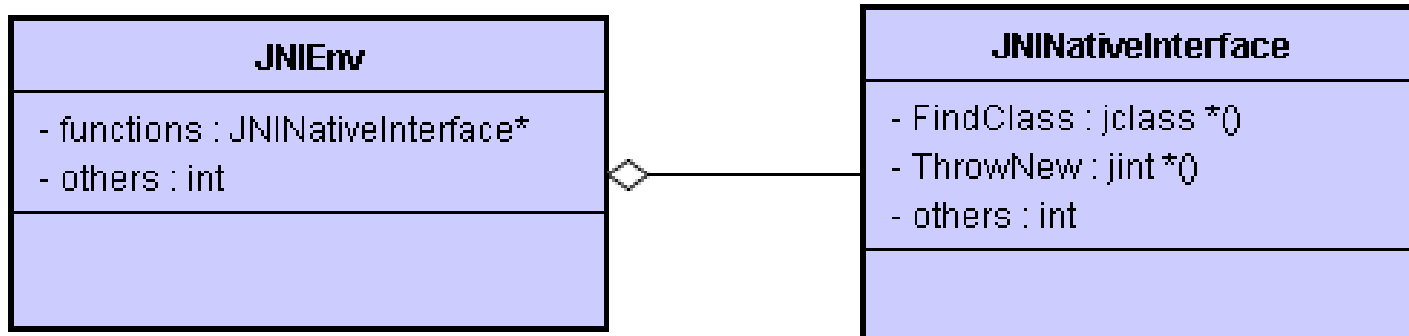
两者之间是什么关系呢？

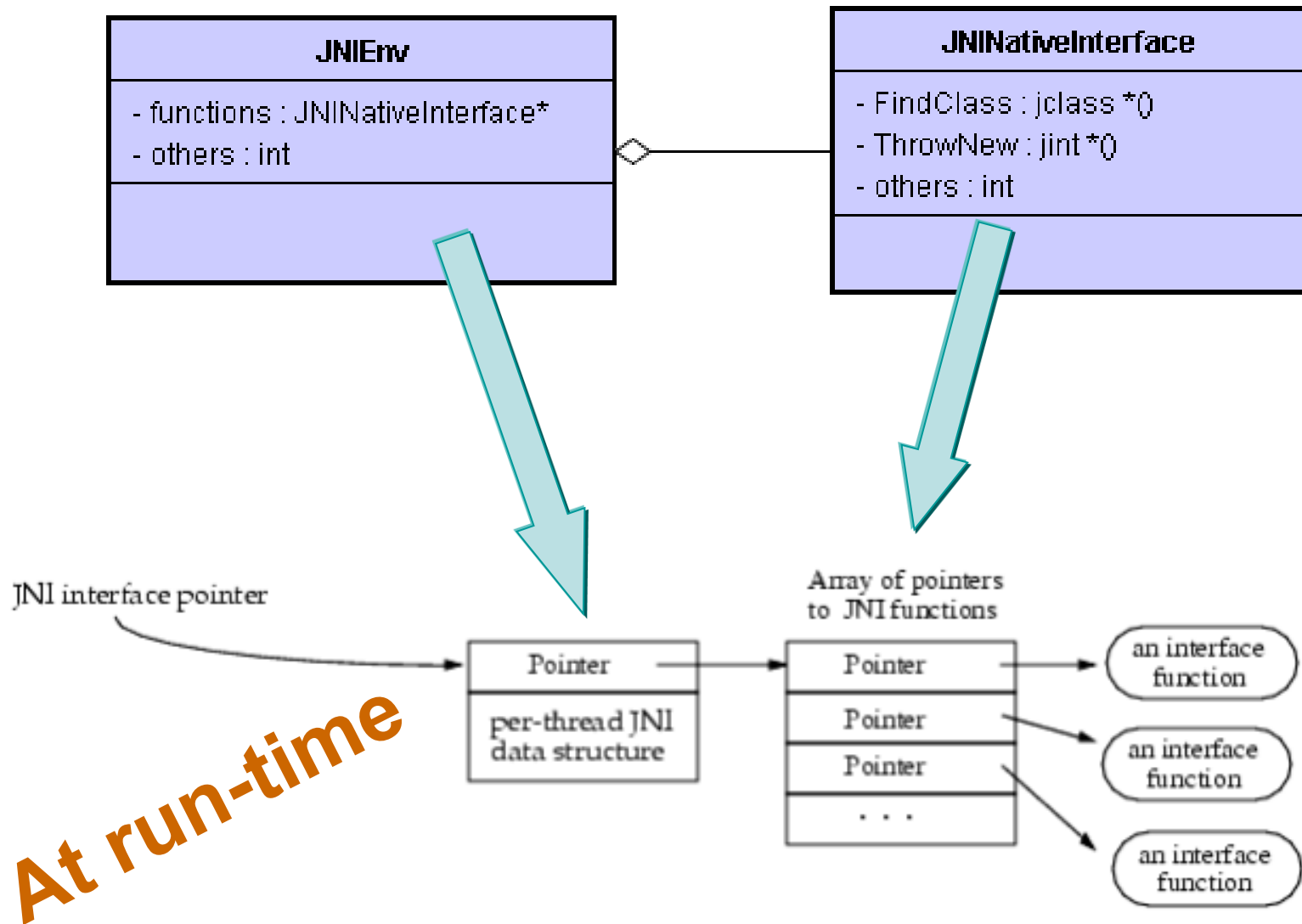
- 两者之间是一种Whole-Part组合 (Aggregation)关系；而不是继承 (Inheritance)关系。

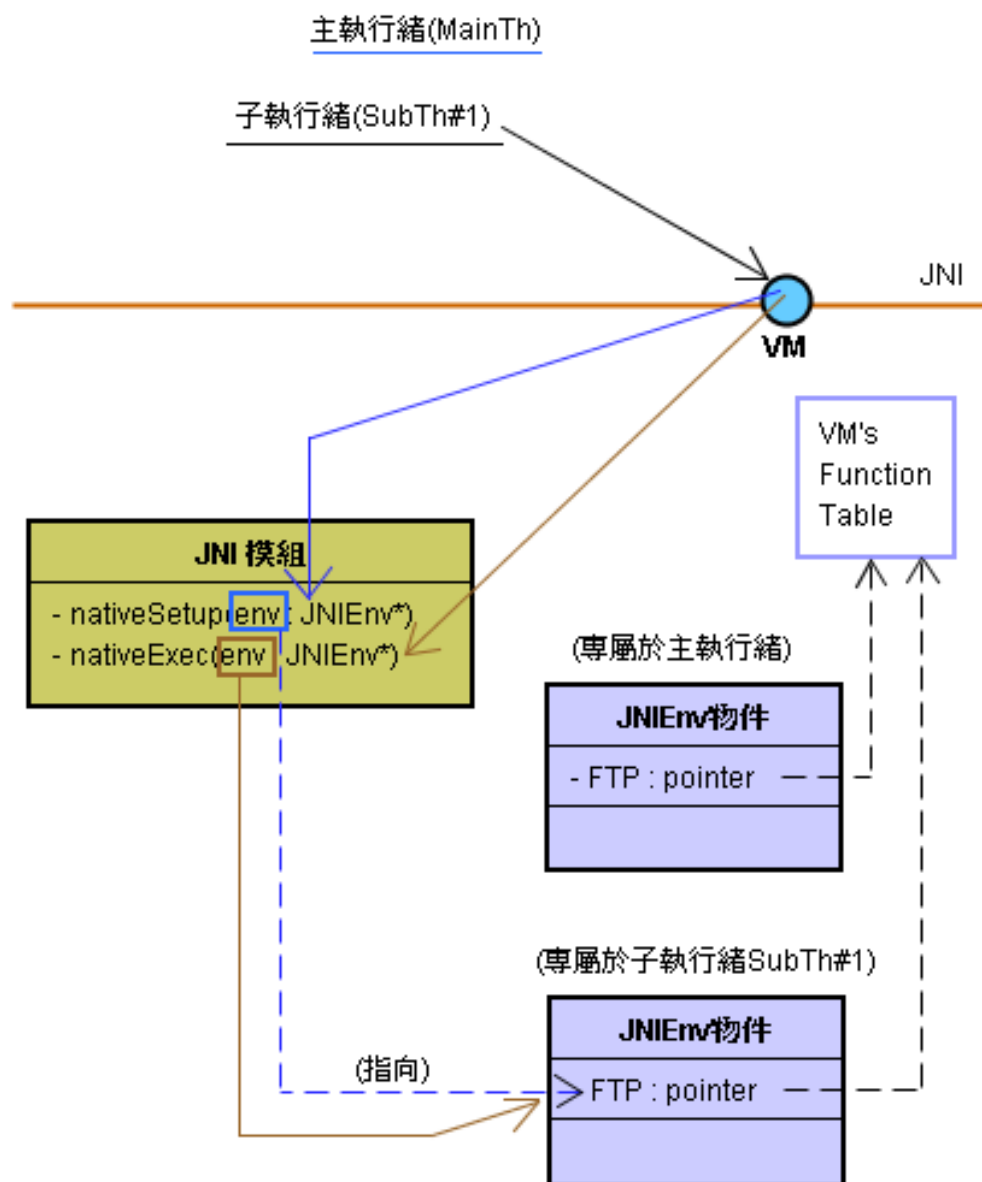
Whole-Part关系



以JNI的JNINativeInterface函数表为例



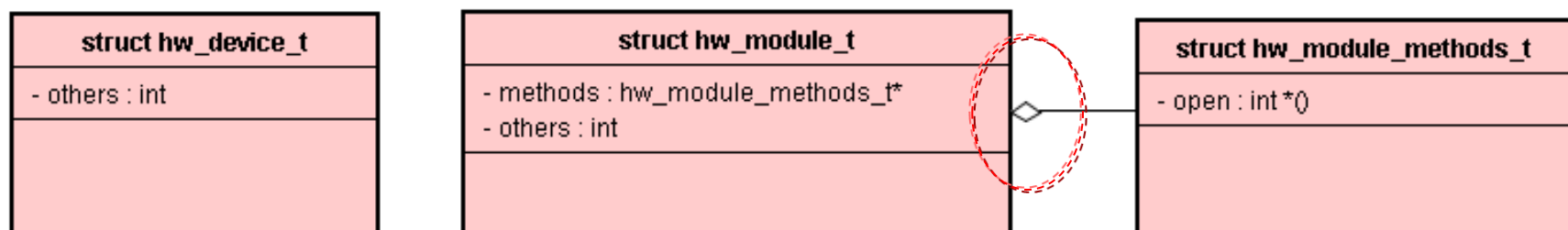




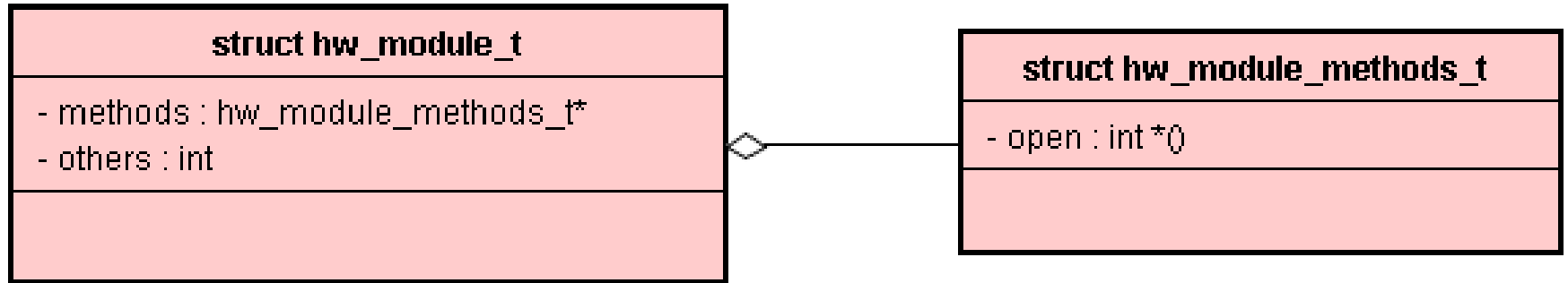
以HAL的hw_module_t函数表为例

- HAL框架里只有3个主要的struct结构。
- 其中的hw_module_methods_t是从hw_module_t独立出来的<函数表定义>。

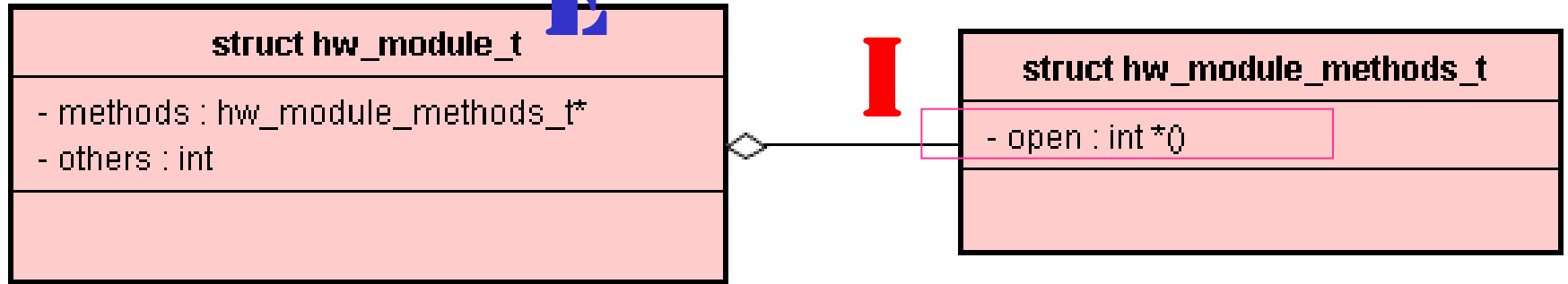
函数表定义



函数表定义



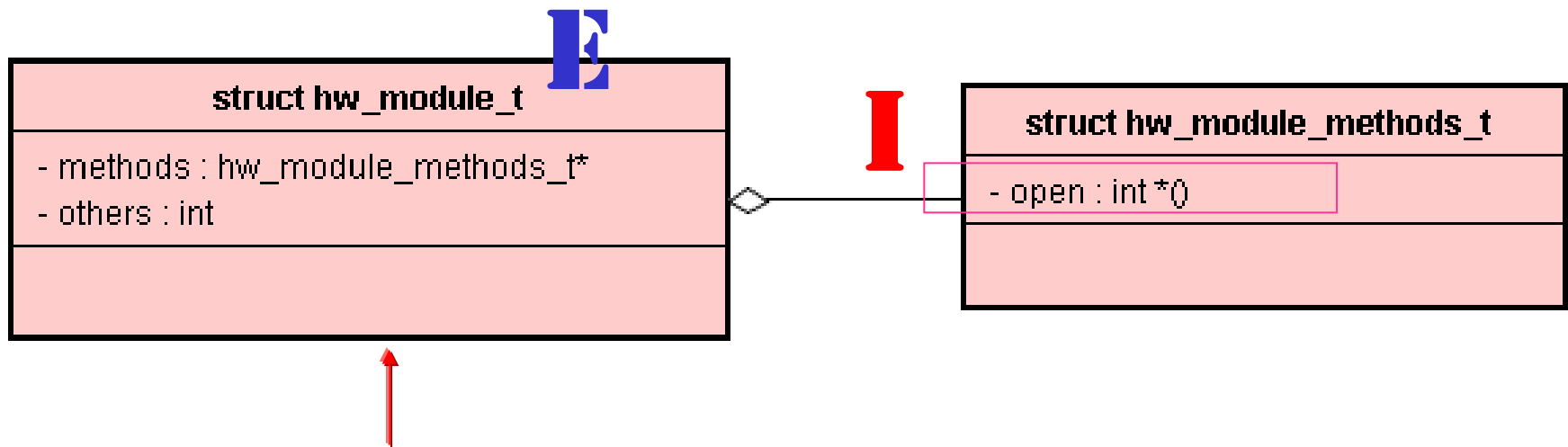
函数表定义



T

创建对象&
设定函数指针

撰写<I>的函数
的实现代码



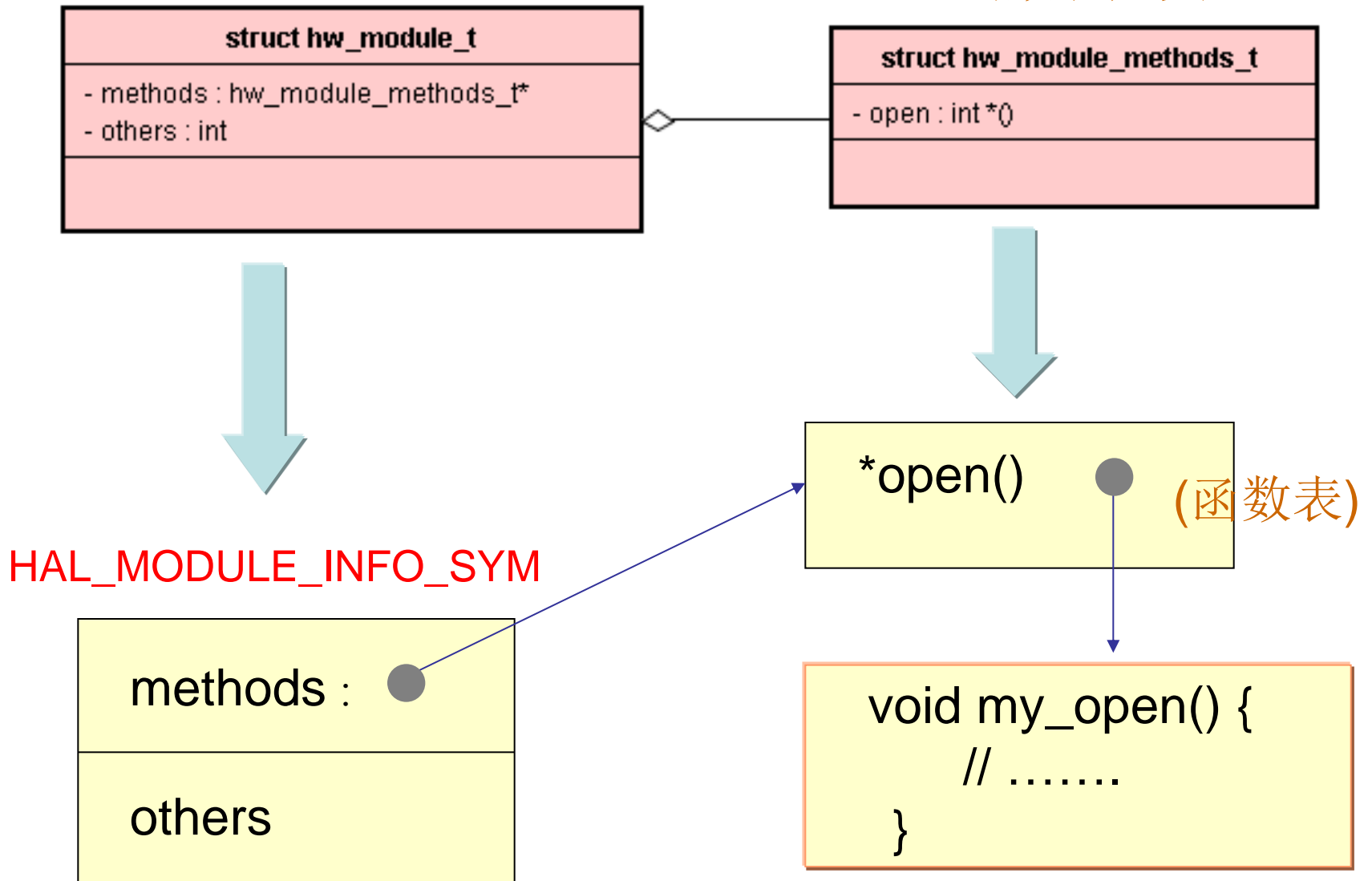
```
static struct hw_module_methods_t my_methods = {
    open: my_open
};
```

```
const struct hw_module_t
HAL_MODULE_INFO_SYM = {
    // .....
    methods: &my_methods,
};
```

```
void my_open() {
    // .....
}
```

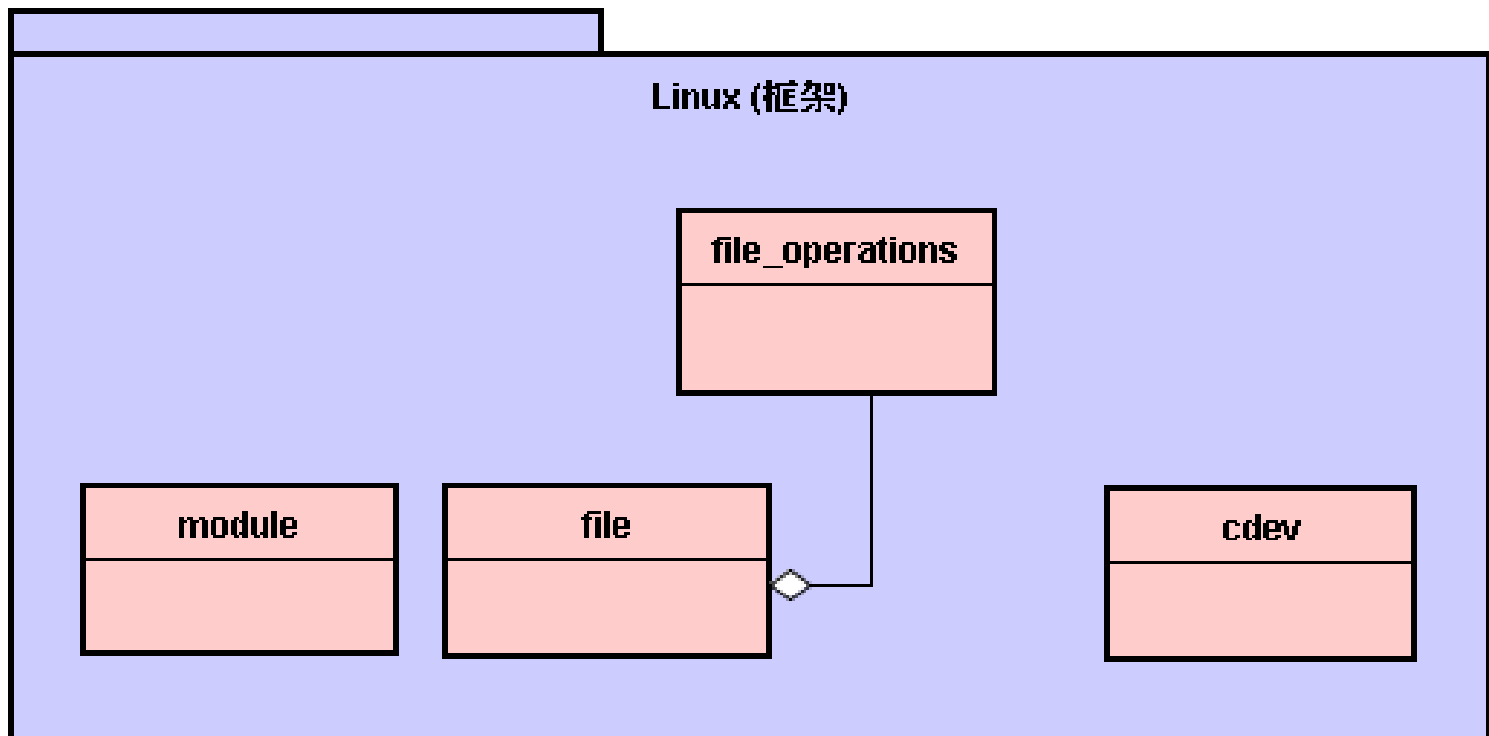

At run-time

函数表定义

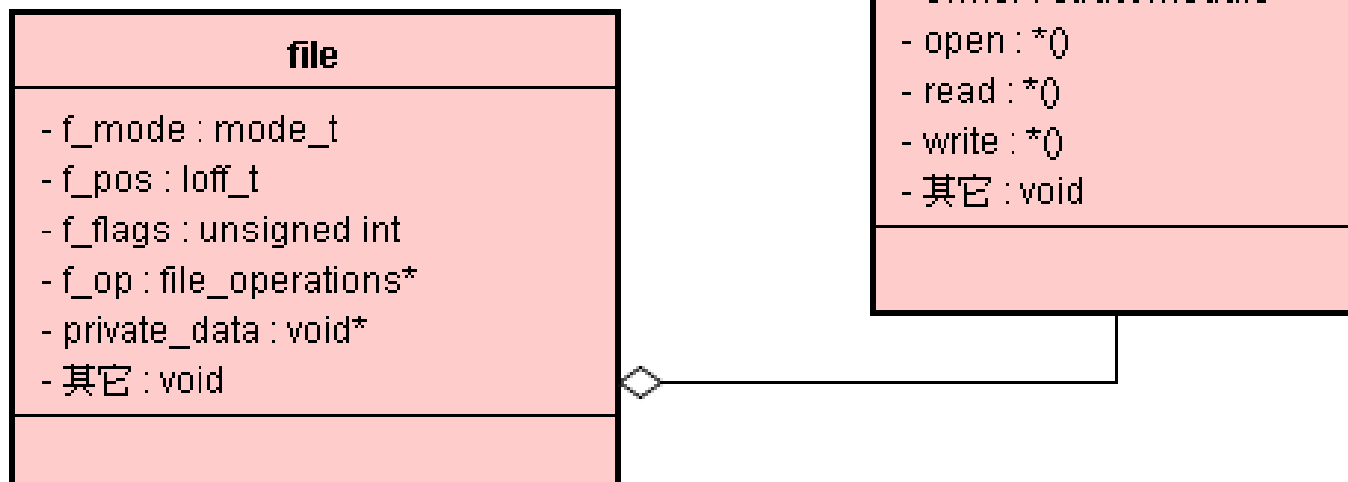


Linux的struct file函数表为例

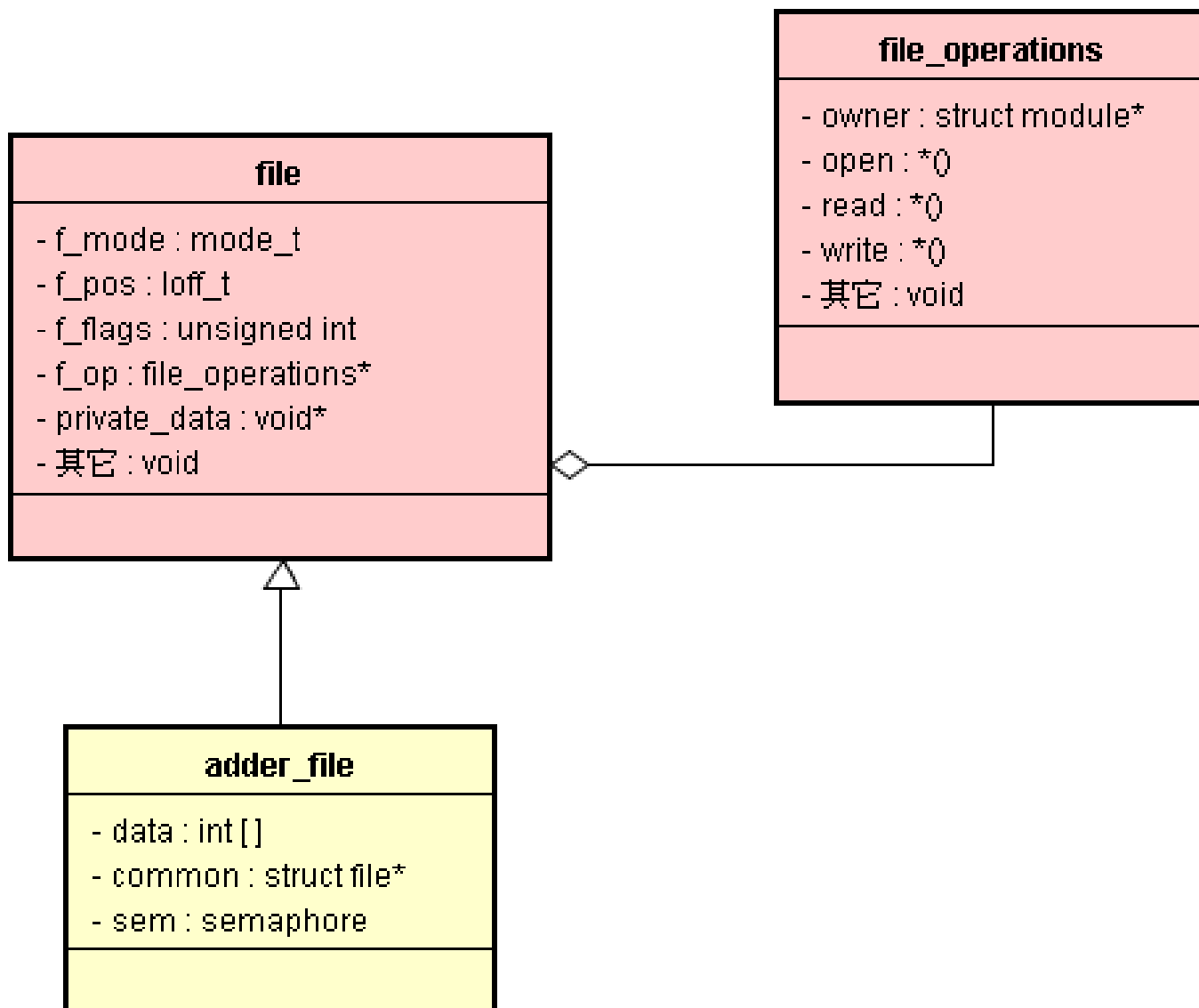
- 将struct file里的一部分函数定义独立出来，成为函数表(function table)，也就是接口(interface)了。
- 例如，从struct file独立出来，成为struct file_operations：



函数表定义



函数表定义



file_operations的對象

*open()
*read()
*write()
其它

adder_file的對象

data[]
*common
sem

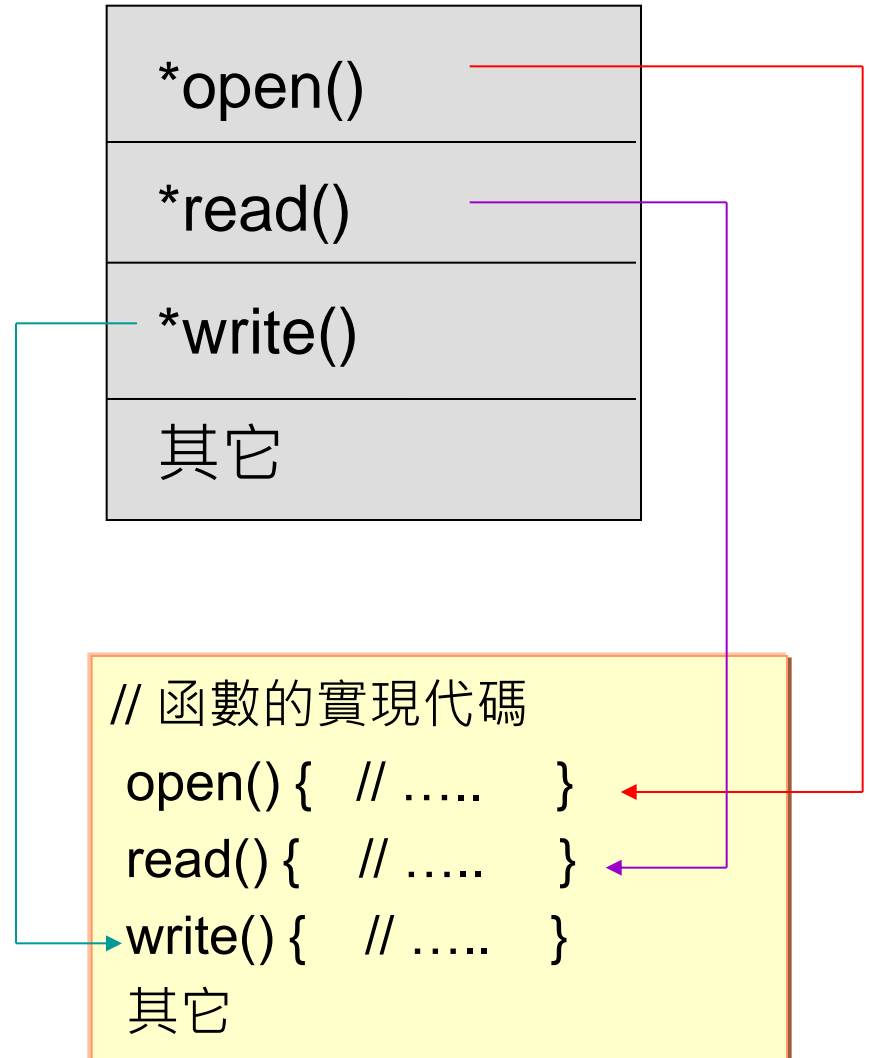
// 函數的實現代碼

open() { // }

read() { // }

write() { // }

其它



- 从上述的3个例子，可以领悟到，函数表在C/C++领域里，是很基础的概念。





~ Continued ~