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

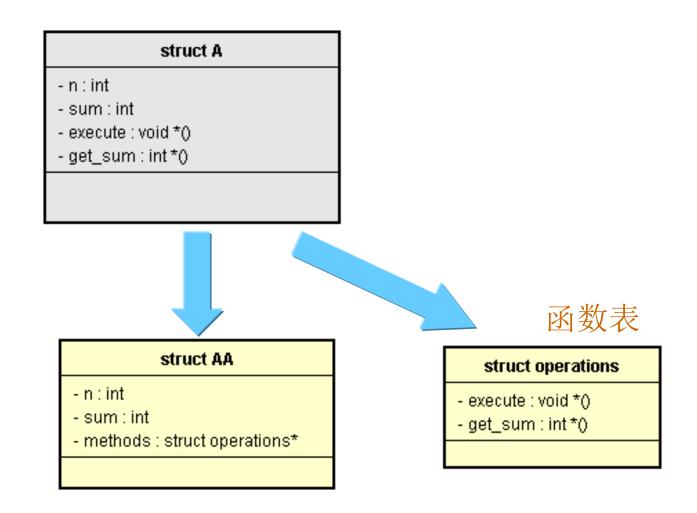
C的函数表(function)概念

• 从一个C的struct谈起

struct A

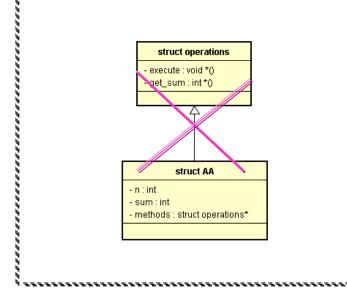
- n : int
- sum : int
- execute : void *()
- get_sum : int *()

 把函数部分独立出来,成为一个函数表 (function table)。



两番之间是什么关系呢?

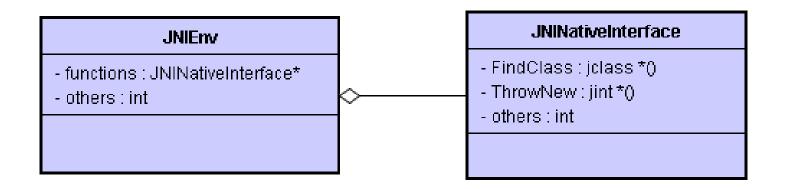
 两者之间是一种Whole-Part组合 (Aggregation)关系;而不是继承 (Inheritance)关系。

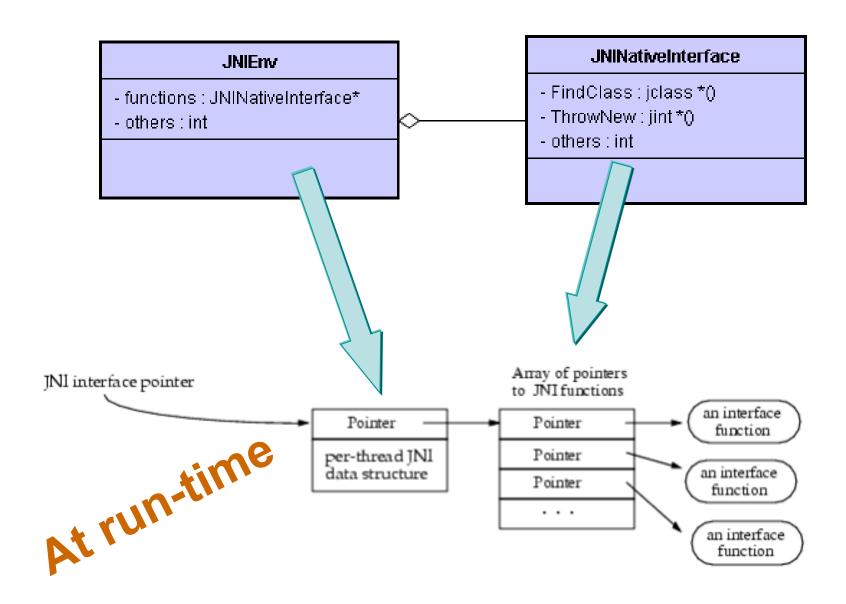


Whole-Part关系

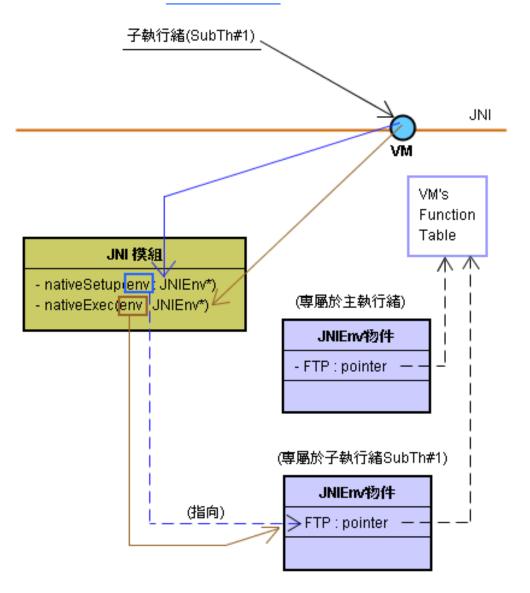
struct AA - n : int - sum : int - methods : struct operations* struct operations - execute : void *() - get_sum : int *()

以JNI的JNIEnv函数表为例





主執行緒(MainTh)

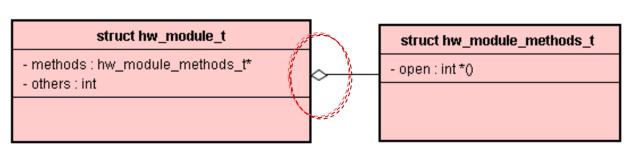




以HAL的hw_module_t函数表为例

- 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>。

struct hw_device_t - others : int

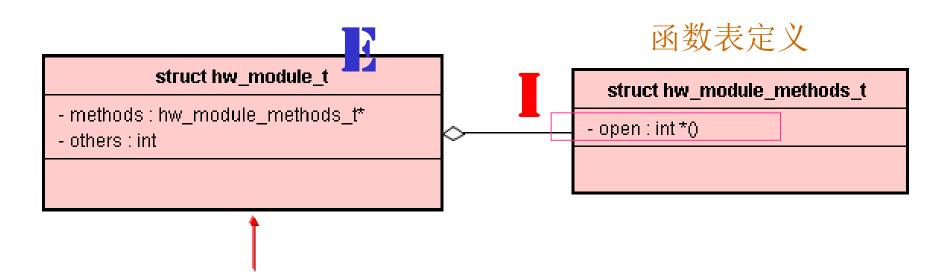


struct hw_module_t

- methods : hw_module_methods_t*
- others : int

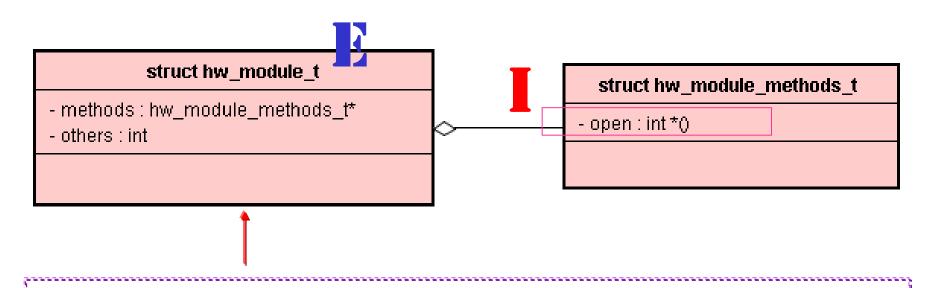
struct hw_module_methods_t

- open : int *()



T

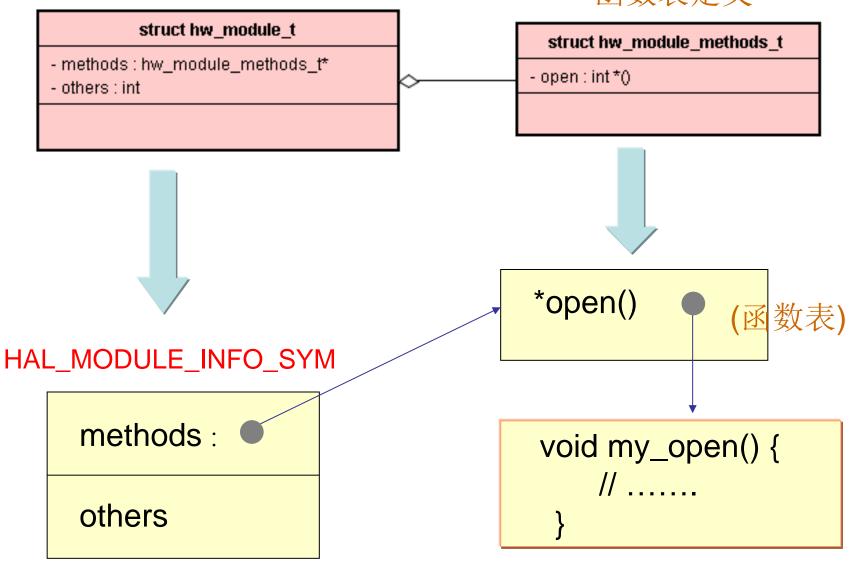
创建对象& 设定函数指针 撰写<I>的函数的实现代码



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

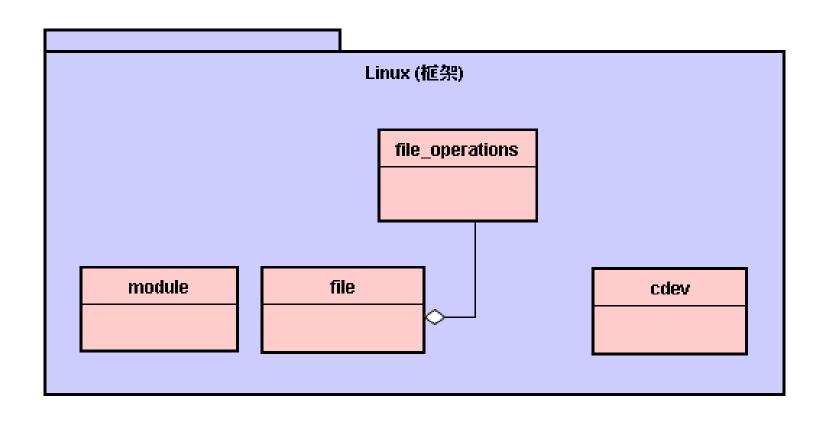
```
const struct hw_module_t
  HAL_MODULE_INFO_SYM = {
    // .....
    methods: &my_methods,
};
```

At run-time



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:



file_operations

- owner : struct module*

- open : *() - read : *() - write : *()

- 其它 : void

file

- f_mode : mode_t

- f_pos : loff_t

- f_flags : unsigned int

f_op : file_operations*

- private_data : void*

- 其它: void

file_operations

file

- f_mode : mode_t

- f_pos : loff_t

- f_flags : unsigned int

- f_op : file_operations*

private_data : void*

- 其它: void

owner : struct module*open : *()

- read : *()

- write : *()

- 其它: void

adder_file

- data : int []

- common : struct file*

- sem : semaphore

file_operations的對象

*open() *read() *write() 其它 // 函數的實現代碼 read() { // } -→write() { // } 其它

adder_file的對象

data[]
*common
sem

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 ~