

绕来绕去终于大概明白了。

STM32有43个channel的settable的中断源；AIRC(Application Interrupt and Reset Register)寄存器中有用于指定优先级的4 bits。这4个bits用于分配preemption优先级和sub优先级，在STM32的固件库中定义如下

```
/* Preemption Priority Group -----*/
#define NVIC_PriorityGroup_0      ((u32)0x700) /* 0 bits for pre-emption priority
                                                4 bits for subpriority */
#define NVIC_PriorityGroup_1      ((u32)0x600) /* 1 bits for pre-emption priority
                                                3 bits for subpriority */
#define NVIC_PriorityGroup_2      ((u32)0x500) /* 2 bits for pre-emption priority
                                                2 bits for subpriority */
#define NVIC_PriorityGroup_3      ((u32)0x400) /* 3 bits for pre-emption priority
                                                1 bits for subpriority */
#define NVIC_PriorityGroup_4      ((u32)0x300) /* 4 bits for pre-emption priority
                                                0 bits for subpriority */
```

形象化的理解是：

你是上帝，
造了43个人，这么多人要分社会阶级和社会阶层了；
因为“阶级”的词性比较重；“阶层”比较中性，
所以preemption优先级->阶级；每个阶级内部，有一些阶层，sub优先级->阶层；

如果按照NVIC_PriorityGroup_4这么分，就分为了16个阶级（1个阶层就是1个preemption优先级），0个阶层；高级别的人，可以打断低级别的正在做事的人（嵌套），最多可以完成1个中断和15级嵌套。

每个阶级（每个preemption优先级），你来指定这43人中，谁进入该阶级；一个人叫EXTI0_IRQChannel，你指定他进入“阶级8”，则

43个中断任何一个都可以分配在不同的阶级中不同的阶层

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQChannel;
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 8; // 指定抢占式优先级1，可取0-15
```

阶级

但如果同时申请中断，要看阶层数决定

另外，在同一阶级内部，一个人在做事的时候，另外一个人不能打断他；（preemption优先级相同的中断源之间没有嵌套关系）

还有，如果他们两个同时想做事，因为没有阶层，那么就根据Vector table中的物理排序，让排名靠前的人去做；

又有1个人SPI1_IRQChannel，设定如下

```
NVIC_InitStructure.NVIC_IRQChannel = SPI1_IRQChannel;
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; // 指定抢占式优先级1，可取0-15
```

SPI1_IRQChannel的阶级高，EXTI0_IRQChannel做事的时候可以打断（嵌套）。

因为阶级决定嵌套，而阶层绝对同阶级内优先级的高低。

如果按照NVIC_PriorityGroup_3这么分，就分为了8个阶级（1个阶级是1个preemption优先级），每个阶级内有2个阶层（sub优先级）；高级别的人，可以打断低级别的正在做事的人（嵌套），最多可以完成1个中断和7级嵌套。

每个阶级（每个preemption优先级），你来指定这43人中，谁进入该阶级；一个人叫EXTI0_IRQChannel，你指定他进入“阶级3”，则：

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQChannel;
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 3; // 指定抢占式优先级1，可取0-7
```

还需要指定他的阶层：

```
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0; // 指定响应优先级0，可取0-1
```

另有1个人叫EXTI9_5_IRQChannel，他的阶级和阶层设定如下

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQChannel;
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 3; // 指定抢占式优先级0，可取0-7
```

```
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1; // 指定响应优先级1
```

那么这两个人是同一阶级的兄弟，一个人在做事的时候，另外一个人不能打断他；（preemption优先级相同的中断源之间没有嵌套关系）

如果他们两个同时想做事，因为前者的阶层高，所以前者优先。

还有一个人叫USART1_IRQChannel，他的阶级和阶层设定如下

```
NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQChannel;
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 2; // 指定抢占式优先级0，可取0-7
```

```
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1; // 指定响应优先级1
```

USART1_IRQChannel的优先级最高，当前面两个人做事的时候，他都可以打断（嵌套）。

以下的类推。

如果按照NVIC_PriorityGroup_0这么分，那么没有阶级，只有16个阶层了。

需要给各个人指定阶层编号。sub优先级的范围0-15

当一个人做事的时候，另外的人不能打断他（就没有嵌套了）；当多人同时想做事的时候，按照阶层编号的排序，排名靠

前的先做事。

~~阶层编号~~一样的人同时想做事，那么按照Vector Table硬件排序，排名靠前的先做。