```
vector_rst:
 ARM(   swi     SYS_ERROR0      )
 THUMB( svc     #0              )
 THUMB( nop                     )
        b       vector_und
```

反汇编后更readable

00001004 <vector_rst>:

1004: ef9f0000 svc 0x009f0000

1008: ea000064 b 11a0 <vector_und>

1. vector_und

2. vector_stub und macro

```
/*
 * Undef instr entry dispatcher
 * Enter in UND mode, spsr = SVC/USR CPSR, lr = SVC/USR PC
 */
        vector_stub     und, UND_MODE

        .long   __und_usr                       @  0 (USR_26 / USR_32)
        .long   __und_invalid                   @  1 (FIQ_26 / FIQ_32)
        .long   __und_invalid                   @  2 (IRQ_26 / IRQ_32)
        .long   __und_svc                       @  3 (SVC_26 / SVC_32)
        .long   __und_invalid                   @  4
        .long   __und_invalid                   @  5
        .long   __und_invalid                   @  6
        .long   __und_invalid                   @  7
        .long   __und_invalid                   @  8
        .long   __und_invalid                   @  9
        .long   __und_invalid                   @  a
        .long   __und_invalid                   @  b
        .long   __und_invalid                   @  c
        .long   __und_invalid                   @  d
        .long   __und_invalid                   @  e
        .long   __und_invalid                   @  f
```

kernel与application出现undefined instruction exception的处理差别比较大

3.1 __und_svc()

4. __und_fault()

5. do_undefinstr() in arch/arm/kernel/traps.c

G2 LSP CONFIG_THUMB2_KERNEL is disable

3.2 __und_usr()

需要区分application运行在不同状态(arm, thumb)