ISCAS NIST

循序渐进，学习开发一个 RISC-V 上的操作系统

第 16 章 系统调用

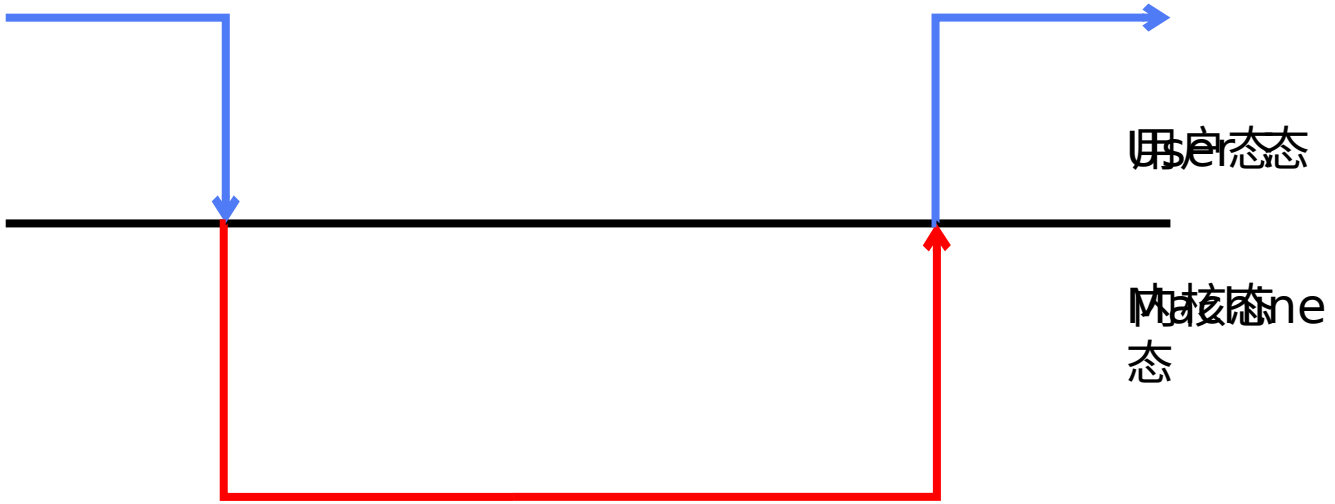
汪辰

2021/4

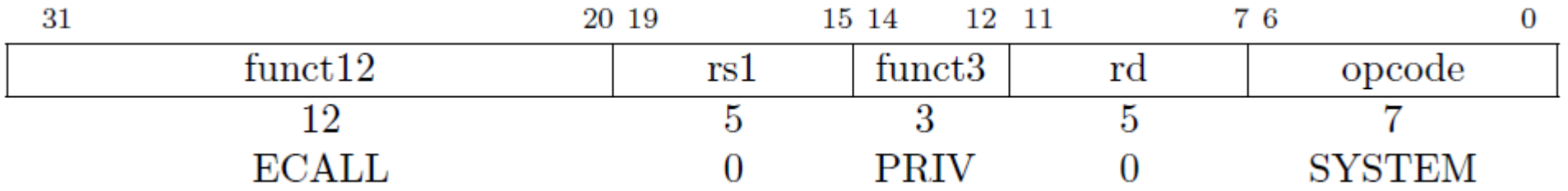
参考

- 【参考 1】： The RISC-V Instruction Set Manual， Volume I: Unprivileged ISA， Document Version 20191213
- 【参考 2】: The RISC-V Instruction Set Manual， Volume II: Privileged Architecture， Document Version 20190608-Priv-MSU-Ratified

用户态和内核态



并发（Concurrency）



【参考 2】 3.2.1 Environment Call and Breakpoint

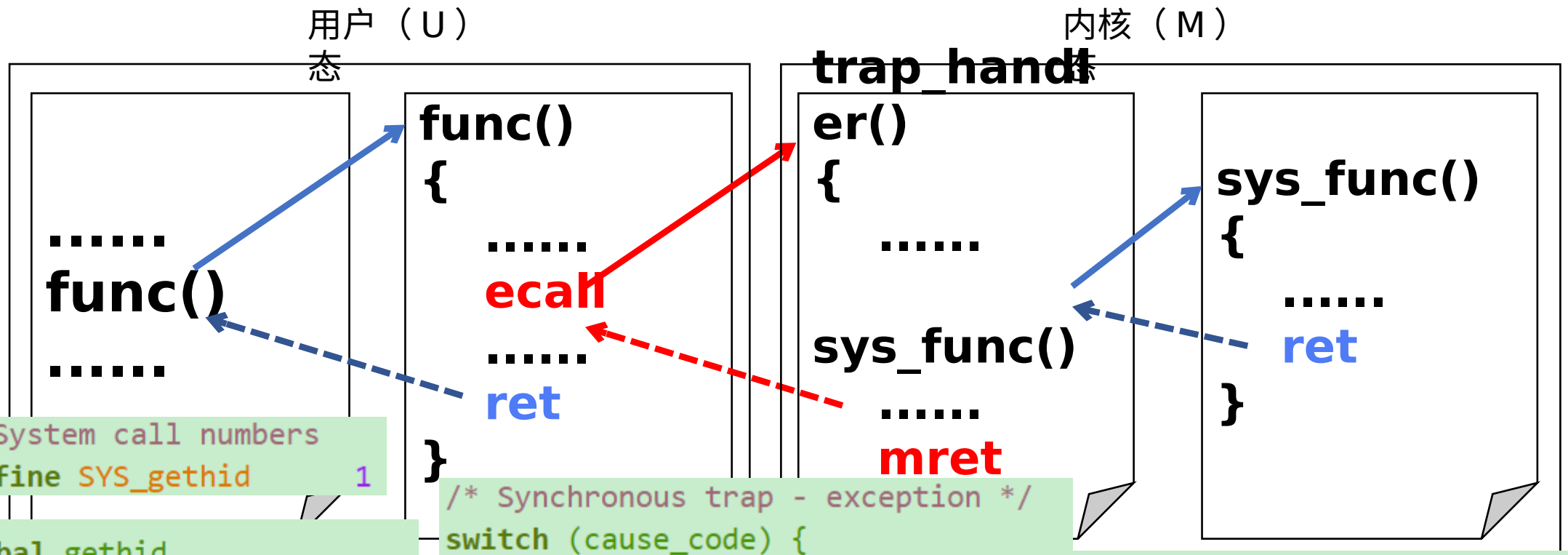
- ECALL 命令用于触发异常
- 根据调用 ECALL 的权限级别产生不同的 code
- 异常产生时 epc 寄存器的值存放的是 ECALL 指令本身的地址
  - 。

Interrupt	Exception Code	Description
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9	Environment call from S-mode
0	10	*Reserved*
0	11	Environment call from M-mode
0	12	Instruction page fault
0	13	Load page fault
0	14	*Reserved for future standard use*
0	15	Store/AMO page fault
0	16–23	*Reserved for future standard use*
0	24–31	*Reserved for custom use*
0	32–47	*Reserved for future standard use*
0	48–63	*Reserved for custom use*
0	≥64	*Reserved for future standard use*

Table 3.6: Machine cause register (mcause) values after trap.

【参考 2】

并发（Concurrency）




```
// System call numbers
#define SYS_gethid      1
```

```
.global gethid
gethid:
        li a7, SYS_gethid
        ecall
        ret
```

```
/* Synchronous trap - exception */
switch (cause_code) {
case 8:
        do_syscall(cxt);
        return_pc += 4;
        break;
```

```
int sys_gethid(unsigned int *ptr_hid)
{
        printf("--> sys_gethid, arg0 = 0x%x\n", ptr_hid);
        if (ptr_hid == NULL) {
                return -1;
        } else {
                *ptr_hid = r_mhartid();
                return 0;
        }
}
```

谢 谢

欢迎交流合作