第六讲虚拟存储概念

第 7 节 RISC-V 缺页异常

向勇、陈渝

清华大学计算机系

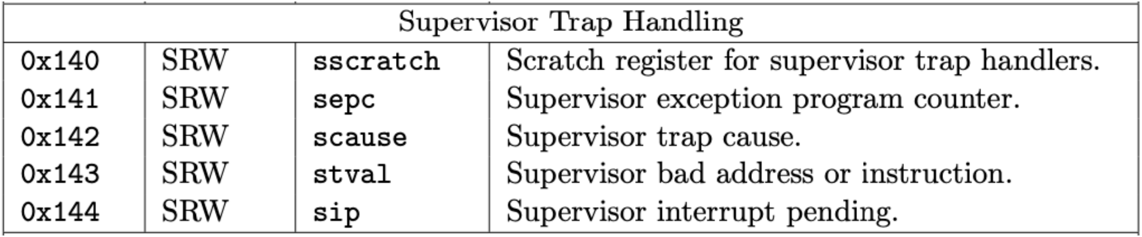
xyong,yuchen@tsinghua.edu.cn

2020 年 5 月 5 日

提纲

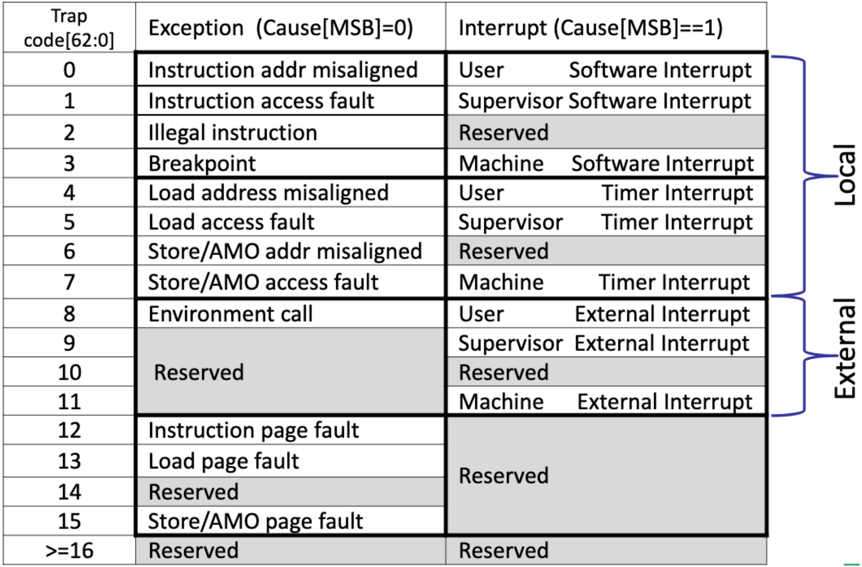
1. 第 7 节 RISC-V 缺页异常
   - 内核态的中断处理机制回顾
   - RISC-V 缺页异常
   - rCore 中的 RISC-V 缺页异常处理函数

内核态的中断处理机制回顾



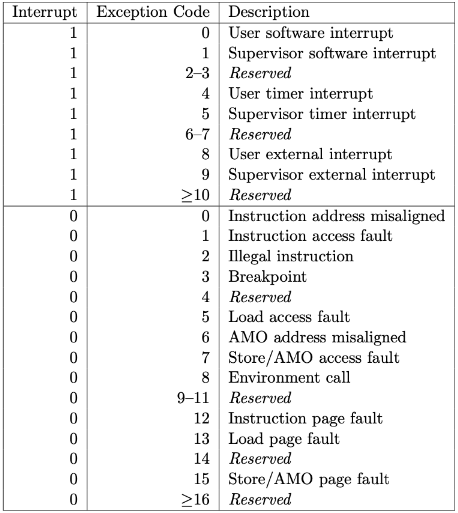
Supervisor Trap Handling			
`0x140`	SRW	`sscratch`	Scratch register for supervisor trap handlers.
`0x141`	SRW	`sepc`	Supervisor exception program counter.
`0x142`	SRW	`scause`	Supervisor trap cause.
`0x143`	SRW	`stval`	Supervisor bad address or instruction.
`0x144`	SRW	`sip`	Supervisor interrupt pending.

中断原因寄存器（scause）



Trap code[62:0]	Exception (Cause[MSB]=0)	Interrupt (Cause[MSB]==1)
0	Instruction addr misaligned	User Software Interrupt
1	Instruction access fault	Supervisor Software Interrupt
2	Illegal instruction	Reserved
3	Breakpoint	Machine Software Interrupt
4	Load address misaligned	User Timer Interrupt
5	Load access fault	Supervisor Timer Interrupt
6	Store/AMO addr misaligned	Reserved
7	Store/AMO access fault	Machine Timer Interrupt
8	Environment call	User External Interrupt
9	Reserved	Supervisor External Interrupt
10		Reserved
11		Machine External Interrupt
12	Instruction page fault	Reserved
13	Load page fault	
14	Reserved	
15	Store/AMO page fault	
>=16	Reserved	Reserved

Volume II: RISC-V Privileged Architectures V1.10



Interrupt	Exception Code	Description
1	0	User software interrupt
1	1	Supervisor software interrupt
1	2–3	*Reserved*
1	4	User timer interrupt
1	5	Supervisor timer interrupt
1	6–7	*Reserved*
1	8	User external interrupt
1	9	Supervisor external interrupt
1	≥10	*Reserved*
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	*Reserved*
0	5	Load access fault
0	6	AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call
0	9–11	*Reserved*
0	12	Instruction page fault
0	13	Load page fault
0	14	*Reserved*
0	15	Store/AMO page fault
0	≥16	*Reserved*

图: Supervisor cause register (scause) values after trap

rCore 的缺页异常处理

- 缺页异常只会在 MMU 启用后，虚拟地址翻译失败时产生，这时候根据是取指还是访存，分别触发不同的缺页异常。
- 当状态码是 Instruction page fault、Load page fault、Store page fault 时，将被判断为是缺页异常，并调用 ‘handle_page_fault()‘ 处理缺页异常。
- 发生缺页异常时，虚拟地址将会被保存到 stval 寄存器中；再调用 ‘crate::memory::page_fault_handler(addr)‘ 来做具体的缺页处理。
- 出处：rCore (Commit ID cd81f4c)

rCore 中 RISC-V 的缺页处理函数 handle_page_fault

- kernel/src/arch/riscv/interrupt.rs

  ```
  fn rust_trap(tf: &mut TrapFrame)
  ```

- kernel/src/arch/riscv/interrupt.rs

  ```
  fn page_fault(tf: &mut TrapFrame)
  ```

- kernel/src/memory.rs

  ```
  pub fn handle_page_fault(addr: usize) -> bool
  ```

- crate/memory/src/memory_set/mod.rs

  ```
  pub fn handle_page_fault(&mut self, addr: VirtAddr) -> bool
  ```

- crate/memory/src/memory_set/handler/delay.rs

  ```
  fn handle_page_fault(&self, pt: &mut dyn PageTable, addr: VirtAddr)
  ```

crate/memory/src/memory_set/handler/delay.rs

```
fn handle_page_fault(&self, pt: &mut dyn PageTable, addr: VirtAddr) -> b
```

```
......
let frame = self.allocator.alloc().expect("failed to alloc frame");
                          ///分配物理页面
entry.set_target(frame);  ///设置物理页号
entry.set_present(true);  ///设置页表项标志位
entry.update();           ///TLB刷新
......
```