

OS lab4 report

Name: 刘乐奇

sid: 12011327

Ubuntu用户名: lynchrocket

请自己总结执行ebreak后，我们的操作系统是如何进行断点中断处理的

ebreak指令会产生一个断点中断信号，操作系统会在抓到中断后先进行现场保护（保存上下文），然后CPU会寻找stvec寄存器中的值并跳转到这个位置（即中断响应函数所在的代码入口地址）进行中断处理，执行完对应代码后再回到中断响应时的现场（恢复上下文）。

请阅读手册，描述epc寄存器的作用


epc寄存器会记录触发中断的那条指令的地址，用于在中断结束时返回到中断之前的地址。

编程题：触发一条非法指令异常（ILLEGAL_INSTRUCTION），在 kern/trap/trap.c 的异常处理函数中捕获，并对其进行处理，简单输出异常类型和指令即可。截图你涉及到的代码。

(提示：可以在S态执行mret汇编指令进行触发)

修改代码

kern/init/init.c




```
const char *message = "os is loading ...\\n";
cputs(message);
idt_init();
intr_enable();
asm volatile("mret"::);
```

kern/trap/trap.c

```

void exception_handler(struct trapframe *tf) {
    switch (tf->cause) {
        case CAUSE_MISALIGNED_FETCH:
            break;
        case CAUSE_FAULT_FETCH:
            break;
        case CAUSE_ILLEGAL_INSTRUCTION:
            cprintf("ILLEGAL_INSTRUCTION: %x\n", tf->cause);
            tf->epc += 4;
            break;
        case CAUSE_BREAKPOINT:

```



执行结果

```

lynchrocket@lynchrocket-virtual-machine: ~/Desktop/OSlab/...
OpenSBI

Platform Name       : QEMU Virt Machine
Platform HART Features : RV64ACDFIMSU
Platform Max HARTs   : 8
Current Hart        : 0
Firmware Base       : 0x80000000
Firmware Size       : 120 KB
Runtime SBI Version  : 0.2

MIDELEG : 0x00000000000000222
MEDELEG : 0x0000000000000b109
PMP0    : 0x0000000080000000-0x000000008001ffff (A)
PMP1    : 0x0000000000000000-0xffffffffffffffff (A,R,W,X)
os is loading ...

ILLEGAL_INSTRUCTION: 2

```