Week4 Report in Class (Fri56)

> 11911839 聂雨荷

Q1

> 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

1. 在 c 文件中插入 ebreak 汇编代码 `asm volatile("ebreak"::);` 会触发一个断点中断
2. 操作系统进入 S 态进行处理
3. cpu 寻找中断向量表 `stvec` 中寄存器的值，跳转到该位置
4. 初始化 `stevc` 中的寄存器，保存中断前程序的上下文
5. 进行中断处理， `trap` 函数分发 interrupt 或是 exception
6. 根据中断和异常的不同类型来处理
7. 使用 `stevc` 中的寄存器恢复上下文，cpu 返回中断代码的位置

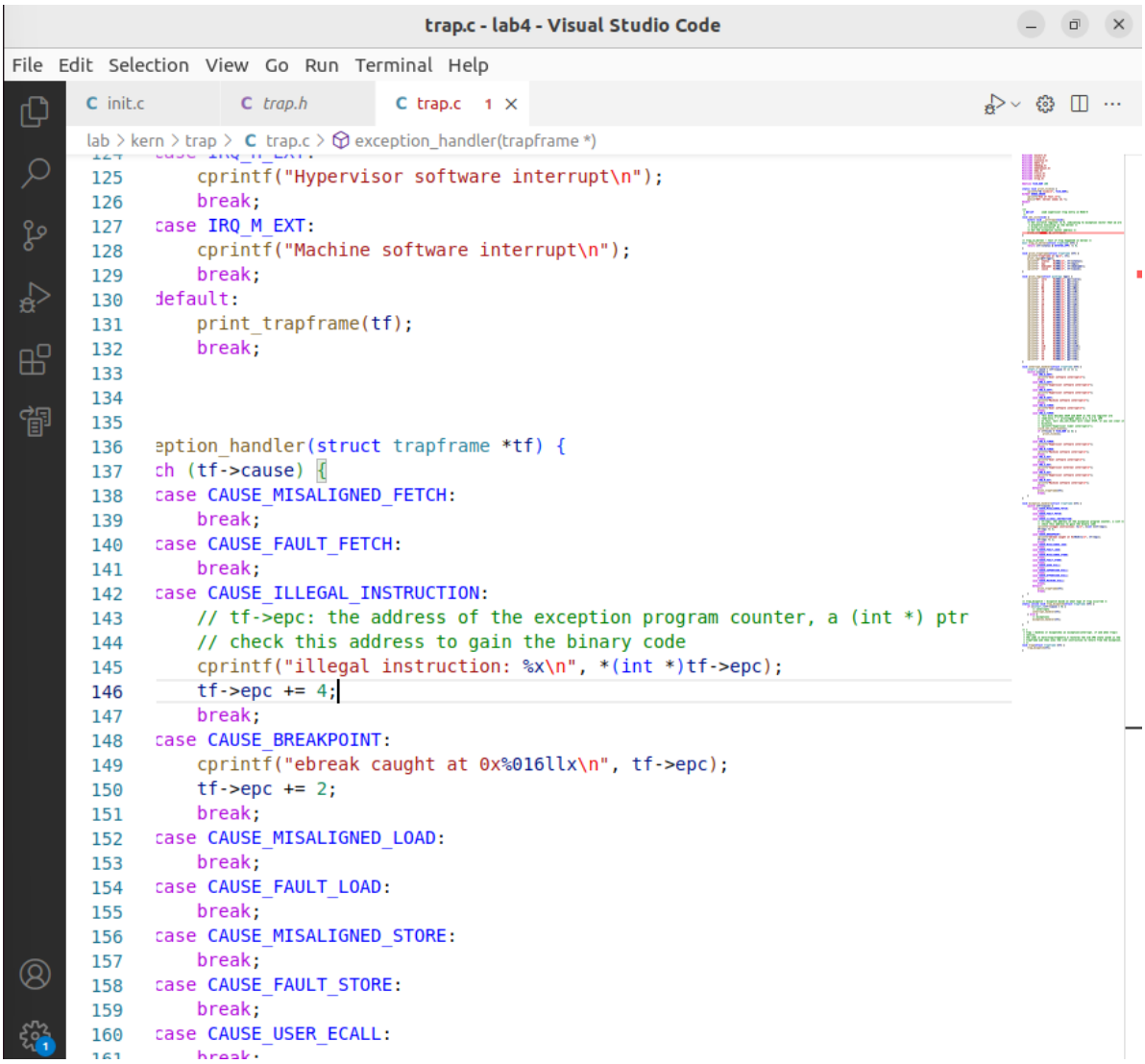
Q2

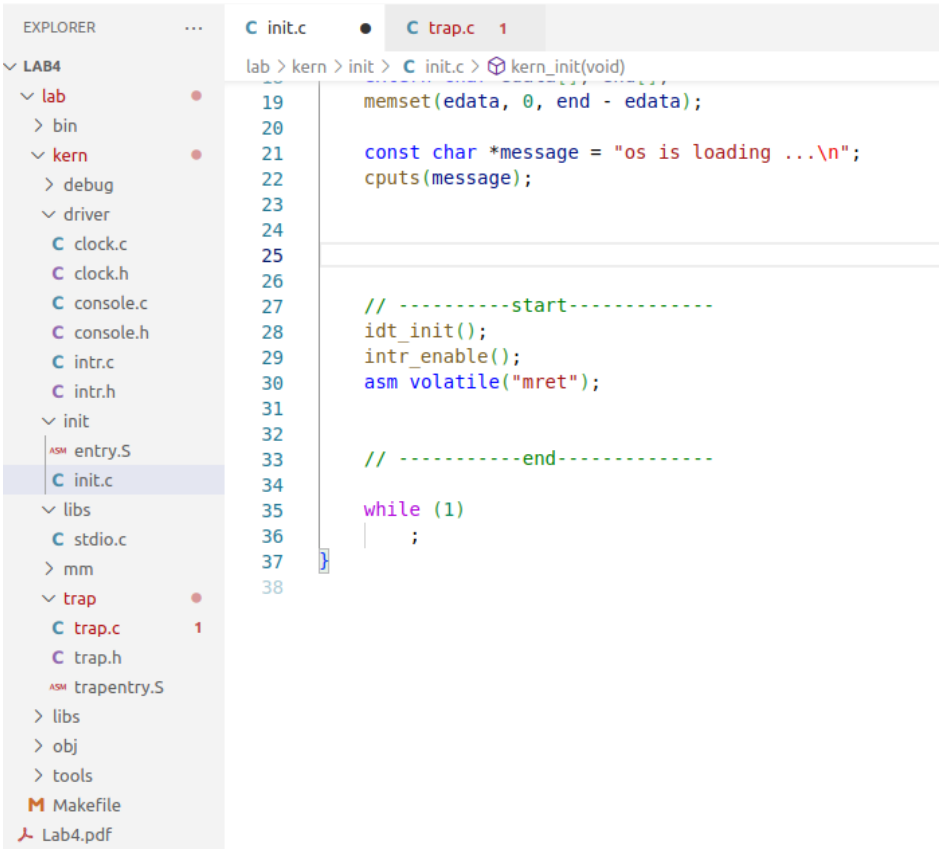
> 请阅读手册，描述epc寄存器的作用

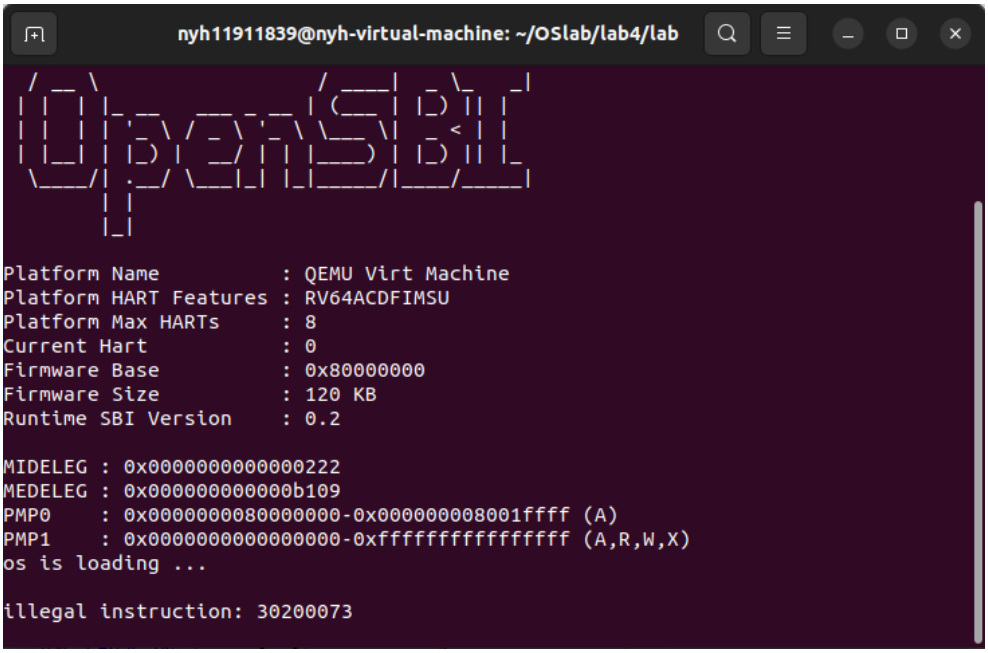
epc: exception program counter，记录中断发生的指令的地址，在中断处理开始前记录程序发生中断的位置，以在中断处理结束后跳转回相应的位置

Q3

> 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，输出异常类型和16进制指令即可。截图你涉及到的代码和运行结果。
>
> 提示：可以在S态执行 `mret` 汇编指令进行触发，注意mret指令在S态出发会导致非法指令异常，但并不代表mret本身是非法指令，也不代表非法指令一定是mret


trap.c - lab4 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
init.c
trap.h
trap.c 1
lab > kern > trap > trap.c > exception_handler(trapframe *)
125 cprintf("Hypervisor software interrupt\n");
126 break;
127 case IRQ_M_EXT:
128 cprintf("Machine software interrupt\n");
129 break;
130 default:
131 print_trapframe(tf);
132 break;
136 eption_handler(struct trapframe *tf) {
137 ch (tf->cause) {
138 case CAUSE_MISALIGNED_FETCH:
139 break;
140 case CAUSE_FAULT_FETCH:
141 break;
142 case CAUSE_ILLEGAL_INSTRUCTION:
143 // tf->epc: the address of the exception program counter, a (int *) ptr
144 // check this address to gain the binary code
145 cprintf("illegal instruction: %x\n", *(int *)tf->epc);
146 tf->epc += 4;
147 break;
148 case CAUSE_BREAKPOINT:
149 cprintf("ebreak caught at 0x%016llx\n", tf->epc);
150 tf->epc += 2;
151 break;
152 case CAUSE_MISALIGNED_LOAD:
153 break;
154 case CAUSE_FAULT_LOAD:
155 break;
156 case CAUSE_MISALIGNED_STORE:
157 break;
158 case CAUSE_FAULT_STORE:
159 break;
160 case CAUSE_USER_ECALL:


EXPLORER
LAB4
lab
bin
kern
debug
driver
clock.c
clock.h
console.c
console.h
intr.c
intr.h
init
entry.S
init.c
libs
stdio.c
mm
trap
trap.c 1
trap.h
trapentry.S
libs
obj
tools
Makefile
Lab4.pdf
init.c
trap.c 1
lab > kern > init > init.c > kern_init(void)
19 memset(edata, 0, end - edata);
20
21 const char *message = "os is loading ...\n";
22 cputs(message);
23
24
25
26
27 // ----------start-------------
28 idt_init();
29 intr_enable();
30 asm volatile("mret");
31
32
33 // -----------end--------------
34
35 while (1)
36 ;
37 }
38


nyh11911839@nyh-virtual-machine: ~/OSlab/lab4/lab
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 : 0x0000000000000222
MEDELEG : 0x000000000000b109
PMP0 : 0x0000000080000000-0x000000008001ffff (A)
PMP1 : 0x0000000000000000-0xffffffffffffffff (A,R,W,X)
os is loading ...
illegal instruction: 30200073