

Week5 Report

姓名: Yitong WANG(王奕童) 11910104@mail.sustech.edu.cn

学号: 11910104

实验课时段: 周五5-6节

实验课教师: Yun SHEN(沈昀) sheny@mail.sustech.edu.cn

实验课SA:

- Yining TANG(汤怡宁) 11811237@mail.sustech.edu.cn
- Yushan WANG(王宇杉) 11813002@mail.sustech.edu.cn

Q1 ebreak 后中断点处理

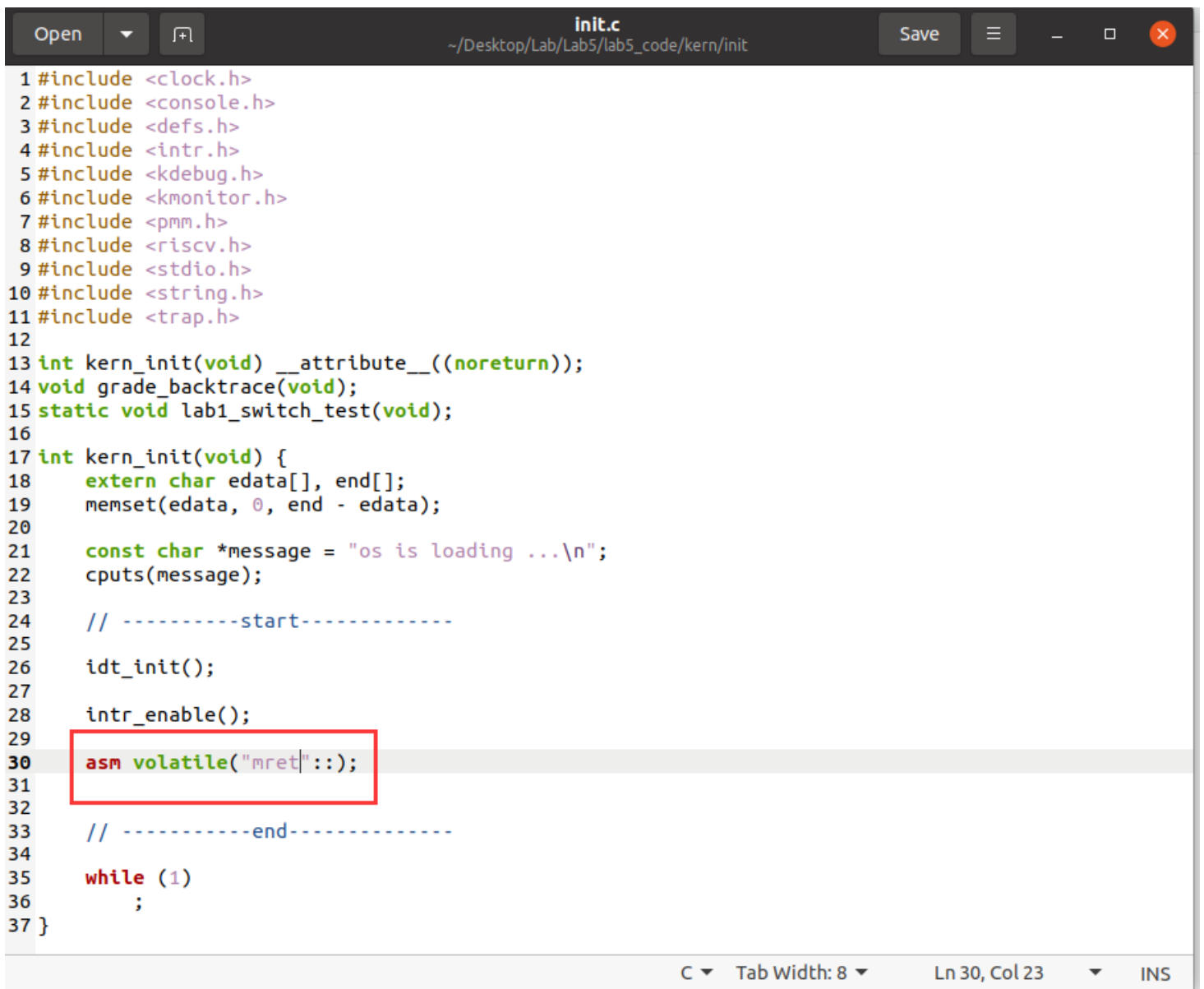
ebreak 指令会触发一个断点中断从而进中断处理流程, 简要流程如下:

- 寻找 stvec 寄存器 (中断向量表基址) 中的值, 跳到中断处理程序的入口点
- 跳转到这个位置进行中断处理, 将 __alltraps 函数的地址放入 stvec 寄存器中。
 - 保存上下文: 使用汇编语言实现, 将所有寄存器保存到栈顶
 - 中断处理: 寄存器 cause CSR (CSR: Control and Status Register) 写入一个指示导致trap产生的原因的数值。中断处理工作有中断处理和异常处理两种, 会根据中断或者异常的不同类型完成处理。
 - 恢复上下文: 恢复顺序与保存顺序相反, 先加载两个CSR, 再加载通用寄存器。
 - 执行 sret, 将S态转换回U态, 返回到先前通过 ebreak 发生中断时, S态对应的地址。

Q2 非法指令异常实现

按照如下步骤做依次操作:

- 修改 kern/init.c 中的汇编代码为 mret :



```
1 #include <clock.h>
2 #include <console.h>
3 #include <defs.h>
4 #include <intr.h>
5 #include <kdebug.h>
6 #include <kmonitor.h>
7 #include <pmm.h>
8 #include <riscv.h>
9 #include <stdio.h>
10 #include <string.h>
11 #include <trap.h>
12
13 int kern_init(void) __attribute__((noreturn));
14 void grade_backtrace(void);
15 static void lab1_switch_test(void);
16
17 int kern_init(void) {
18     extern char edata[], end[];
19     memset(edata, 0, end - edata);
20
21     const char *message = "os is loading ...\n";
22     cputs(message);
23
24     // -----start-----
25
26     idt_init();
27
28     intr_enable();
29
30     asm volatile("mret"::);
31
32
33     // -----end-----
34
35     while (1)
36         ;
37 }
```

C Tab Width: 8 Ln 30, Col 23 INS

- 修改 kern/trap/trap.c 中对于 AUSE_ILLEGAL_INSTRUCTION 的处理分支，输出相关信息：

```
Open  trap.c  Save  -  x
~/Desktop/Lab/Lab5/lab5_code/kern/trap

129     break;
130     default:
131         print_trapframe(tf);
132         break;
133 }
134 }
135
136 void exception_handler(struct trapframe *tf) {
137     switch (tf->cause) {
138         case CAUSE_MISALIGNED_FETCH:
139             break;
140         case CAUSE_FAULT_FETCH:
141             break;
142         case CAUSE_ILLEGAL_INSTRUCTION:
143             cprintf("illegal instruction caught at 0x%016llx\n", tf->epc);
144             tf->epc += 4;
145             break;
146         case CAUSE_BREAKPOINT:
147             cprintf("ebreak caught at 0x%016llx\n", tf->epc);
148             tf->epc += 2;
149             break;
150         case CAUSE_MISALIGNED_LOAD:
151             break;
152         case CAUSE_FAULT_LOAD:
153             break;
154         case CAUSE_MISALIGNED_STORE:
155             break;
156         case CAUSE_FAULT_STORE:
157             break;
158         case CAUSE_USER_ECALL:
159             break;
160         case CAUSE_SUPERVISOR_ECALL:
161             break;
162         case CAUSE_HYPERVISOR_ECALL:
163             break;
164         case CAUSE_MACHINE_ECALL:
165             break;
    }
}
```

C Tab Width: 8 Ln 177, Col 8 INS

- 重新 make qemu 即可输出相关的提示信息，其内包括异常类型与指令的地址：

```
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab5/lab5_code$ make clean
rm -f -r obj bin
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab5/lab5_code$ make qemu
+ cc kern/init/entry.S
+ cc kern/init/init.c
+ cc kern/libs/stdio.c
+ cc kern/debug/panic.c
+ cc kern/debug/kdebug.c
+ cc kern/debug/kmonitor.c
+ cc kern/driver/clock.c
+ cc kern/driver/console.c
+ cc kern/driver/intr.c
+ cc kern/trap/trap.c
+ cc kern/trap/trapentry.S
+ cc kern/mm/pmm.c
+ cc libs/string.c
+ cc libs/printfmt.c
+ cc libs/readline.c
+ cc libs/sbi.c
+ ld bin/kernel
riscv64-unknown-elf-objcopy bin/kernel --strip-all -O binary bin/ucore.bin
```

OpenSBI v0.6



```
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 : 0x00000000000000b109
PMP0    : 0x0000000080000000-0x000000008001ffff (A)
PMP1    : 0x0000000000000000-0xffffffffffff (A,R,W,X)
os is loading ...
```

```
illegal instruction caught at 0x000000008020003a
```