

OS Assignment 2 Report

name: 刘乐奇

sid: 12011327

Ubuntu用户名: lynchrocket

1. 解释指令中每个参数的作用

```
# 模拟 64-bit 的 RISC-V 机器
qemu-system-riscv64 \
    # 指定 qemu 模拟平台是 virt
    -machine virt \
    # 指定无图形界面
    -nographic \
    # 指定 bootloader 为默认
    -bios default \
    # 指定设备的文件
    -device loader,file=bin/ucore.bin,addr=0x80200000
```

2. 解释 /lab/tools/kernel.ld 片段中每一行的作用

```

/* SECTIONS描述了输出文件的内存布局 */
SECTIONS
{
    /* . 是location counter, 如果不指定output section的地址, 该地址会被默认设置为location counter的当前值。location counter
    /* Load the kernel at this address: "." means the current address */
    . = BASE_ADDRESS;

    /* .text 指定代码段。此处指定output section的代码段要将input section中的.text.kern_entry, .text, .stub, .text.*,
    .text : {
        *(.text.kern_entry)
        *(.text .stub .text.* .gnu.linkonce.t.*)
    }

    /* PROVIDE 定义一个 (目标文件内被引用但没定义) 符号。相当于定义一个全局变量的符号表, 其他C文件可以通过该符号来操作对应
    PROVIDE(etext = .); /* Define the 'etext' symbol to this value */

    /* .rodata 指定只读代码段。此处指定output section的只读代码段要将input section中的.rodata, .rodata.*, .gnu.link
    .rodata : {
        *(.rodata .rodata.* .gnu.linkonce.r.*)
    }

    /* 分页地址对齐 */
    /* Adjust the address for the data segment to the next page */
    . = ALIGN(0x1000);

```

3. 解释 /lab/kern/init/init.c 中 main函数中 memset(edata, 0, end - edata); 的参数及语句作用

memset(edata, 0, end - edata); 将 edata 所指向的字符串的前 end-edata 个字符置为 0

注意到 /lab/kern/init/init.c 中有如下两行。extern 关键字说明 edata 和 end 都是外部引用。

```

extern char edata[], end[];
memset(edata, 0, end - edata);

```

注意到 /lab/tools/kernel.ld 中有如下几行。说明上面的 init.c 中的 memset(edata, 0, end - edata); 将 .bss 段中的内容全部置为0

```

PROVIDE(edata = .);

.bss : {
    *(.bss)
    *(.bss.*)
    *(.sbss*)
}

PROVIDE(end = .);

```

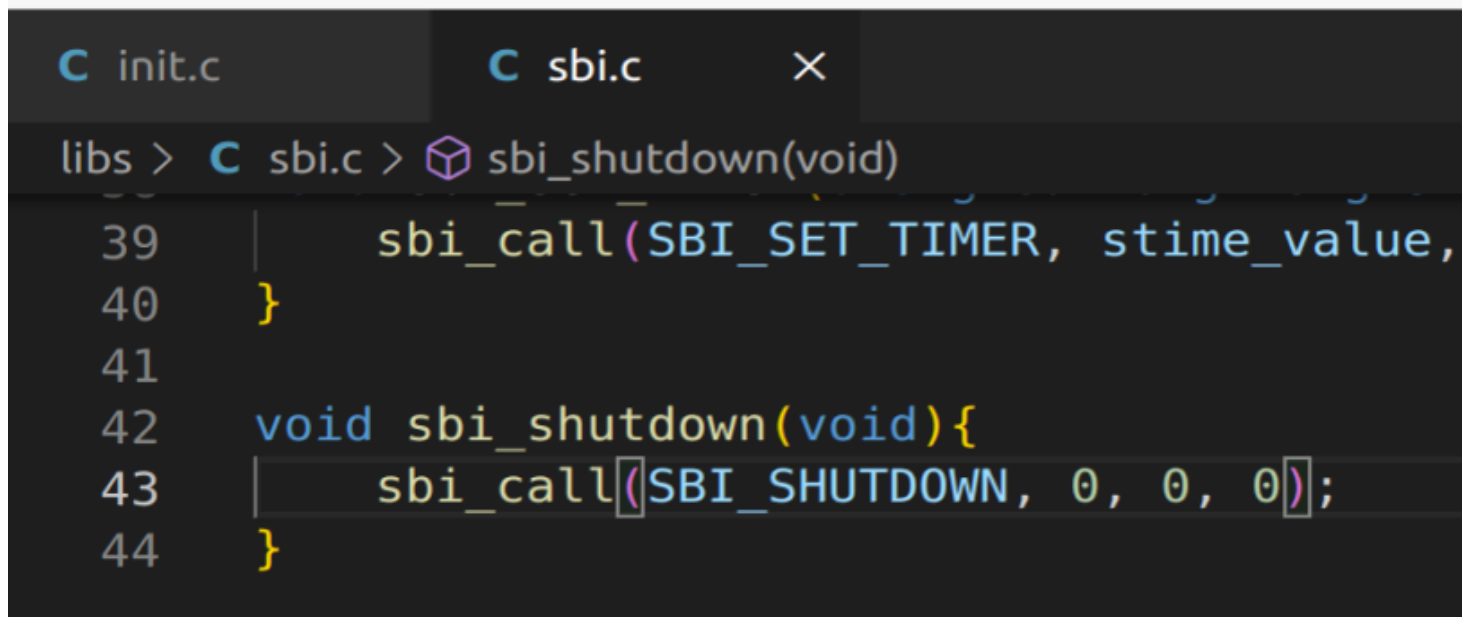
4. 描述cputs()指令是如何通过sbif打印字符的

cputs() 调用了 cputch(), cputch() 调用了 cons_putc(), cons_putc() 调用了 sbi_console_putchar(), sbi_console_putchar() 调用了 sbi_call(), sbi_call() 有如下的内联汇编。

```
uint64_t sbi_call(uint64_t sbi_type, uint64_t arg0, uint64_t arg1, uint64_t arg2) {
    uint64_t ret_val;
    __asm__ volatile (
        "mv x17, %[sbi_type]\n"
        "mv x10, %[arg0]\n"
        "mv x11, %[arg1]\n"
        "mv x12, %[arg2]\n"
        "ecall\n"
        "mv %[ret_val], x10"
        : [ret_val] "=r" (ret_val)
        : [sbi_type] "r" (sbi_type), [arg0] "r" (arg0), [arg1] "r" (arg1), [arg2] "r" (arg2)
        : "memory"
    );
    return ret_val;
}
```

5. 使用ecall打印字符的原理, 实现一个
shutdown() 关机函数

/libs/sbi.c





```
libs > C sbi.c > sbi_shutdown(void)
39     |     sbi_call(SBI_SET_TIMER, stime_value,
40     | }
41
42 void sbi_shutdown(void){
43     |     sbi_call(SBI_SHUTDOWN, 0, 0, 0);
44     | }
```


/kern/init/init.c

```
init.c  ×  sbi.c
kern > init > C init.c > kern_init(void)

10  #include <string.h>
11  #include <trap.h>
12  #include <sbi.h>
13
14  int kern_init(void) __attribute__((noreturn));
15  void grade_backtrace(void);
16  static void lab1_switch_test(void);
17  void shutdown(void) {sbi_shutdown();}
18
19  int kern_init(void) {
20      extern char edata[], end[];
21      memset(edata, 0, end - edata);
22
23      const char *message = "os is loading ...\n";
24      cputs(message);
25
26      cputs("The system will close.\n");
27      shutdown();
28      // idt_init();
```

运行结果

```
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
```

The system will close.

lynchrocket@lynchrocket-virtual-machine:~/Desktop/OSlab/ucore\$