

Name: 陆荻芸 SID: 12011537

最小化内核的启动过程:

首先相当于模拟计算机插电，然后 qemu 调用内置的 OpenSBI 作为 bootloader 准备环境。OpenSBI 把 CPU 从 M Mode 切换到 S Mode，然后跳转到一个固定地址开始执行内核代码

ELF 和 BIN 文件的区别:

ELF 文件是 linux 系统上的主要可执行文件的格式，比较复杂。BIN 文件是二进制可执行文件，内部没有地址标记。

链接脚本的作用:

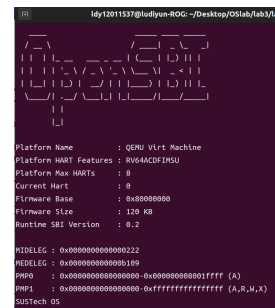
描述如何把输入文件的 section 映射到输出文件的 section, 同时规定这些 section 的内存布局。

```

1  C init.c      X  C libc
2
3  kern > init > C init.c > kern_init(void)
4      #include <intr.h>
5      #include <debug.h>
6      #include <monitor.h>
7      #include <spin.h>
8      #include <criscv.h>
9      #include <stdio.h>
10     #include <string.h>
11     #include <trap.h>
12
13     int kern_init(void) __attribute__((noreturn));
14     void grade_basics(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 = "SUStech OS\n";
22         printf(message);
23
24
25
26

```

(a) init.c with cputs



(b) QEMU output

图 1: A4

```

70 int double_puts(const char *str){
71     int cnt = 0;
72     char c;
73     while ((c = *str++) != '\0') {
74         cputchar(c, &cnt);
75         cputchar(c, &cnt);
76     }
77     cputchar('\n', &cnt);
78     return cnt;
79 }
80

```

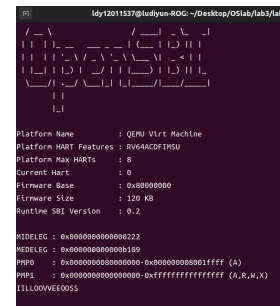
(a) `stdio.c` with `double_puts`

```

17 int kern_init(void) {
18     extern char edata[], end[];
19     memset(edata, 0, end - edata);
20
21     // const char *message = "SUSTech OS\n";
22     // cputs(message);
23     const char *message = "ILOVEOS\n";
24     double_puts(message);
25
26
27

```

(b) `init.c` with `double_puts`



(c) QEMU with double_puts

图 2: A5