Week3 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

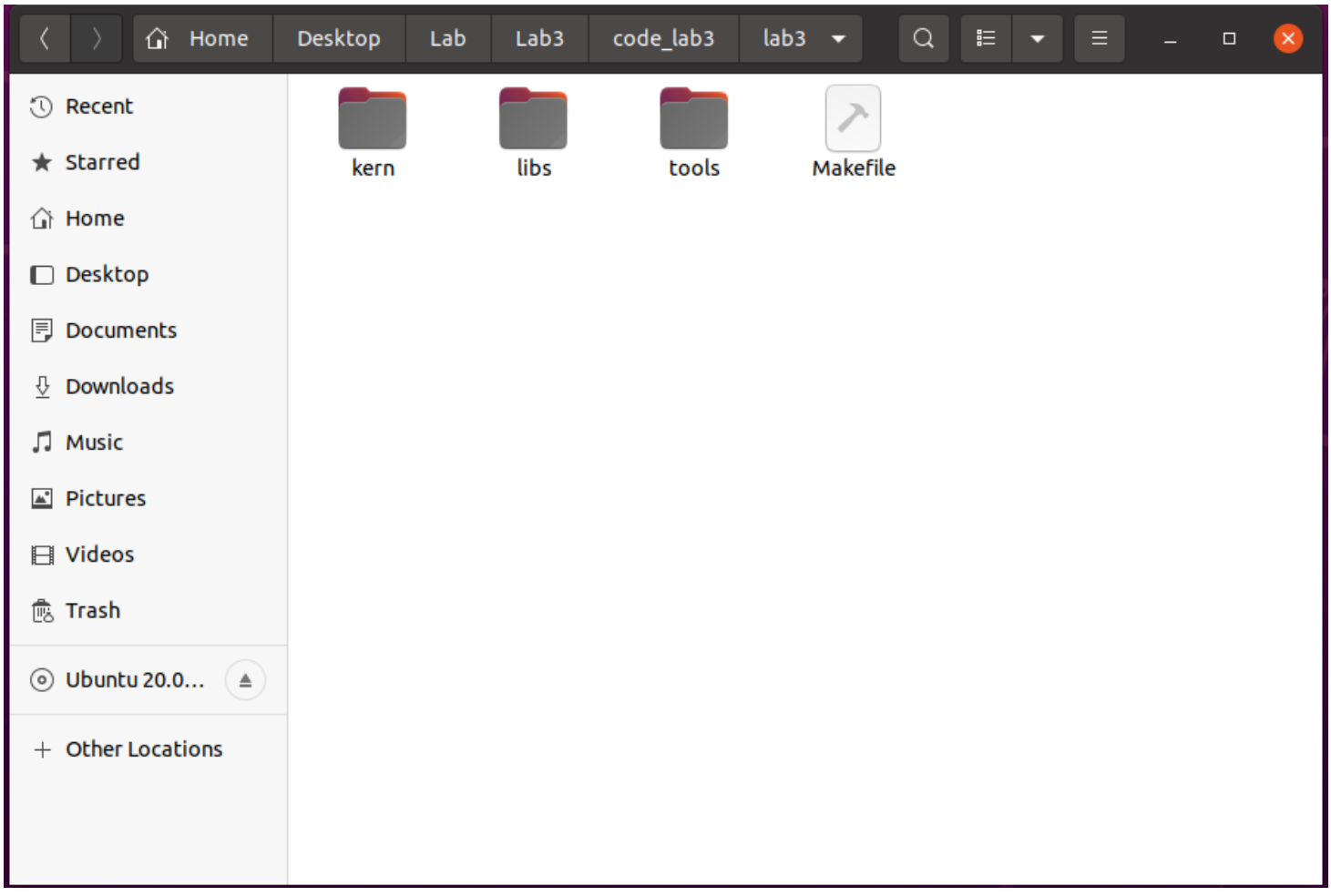
实验课教师：Yun SHEN(沈昀) sheny@mail.sustech.edu.cn

实验课SA:

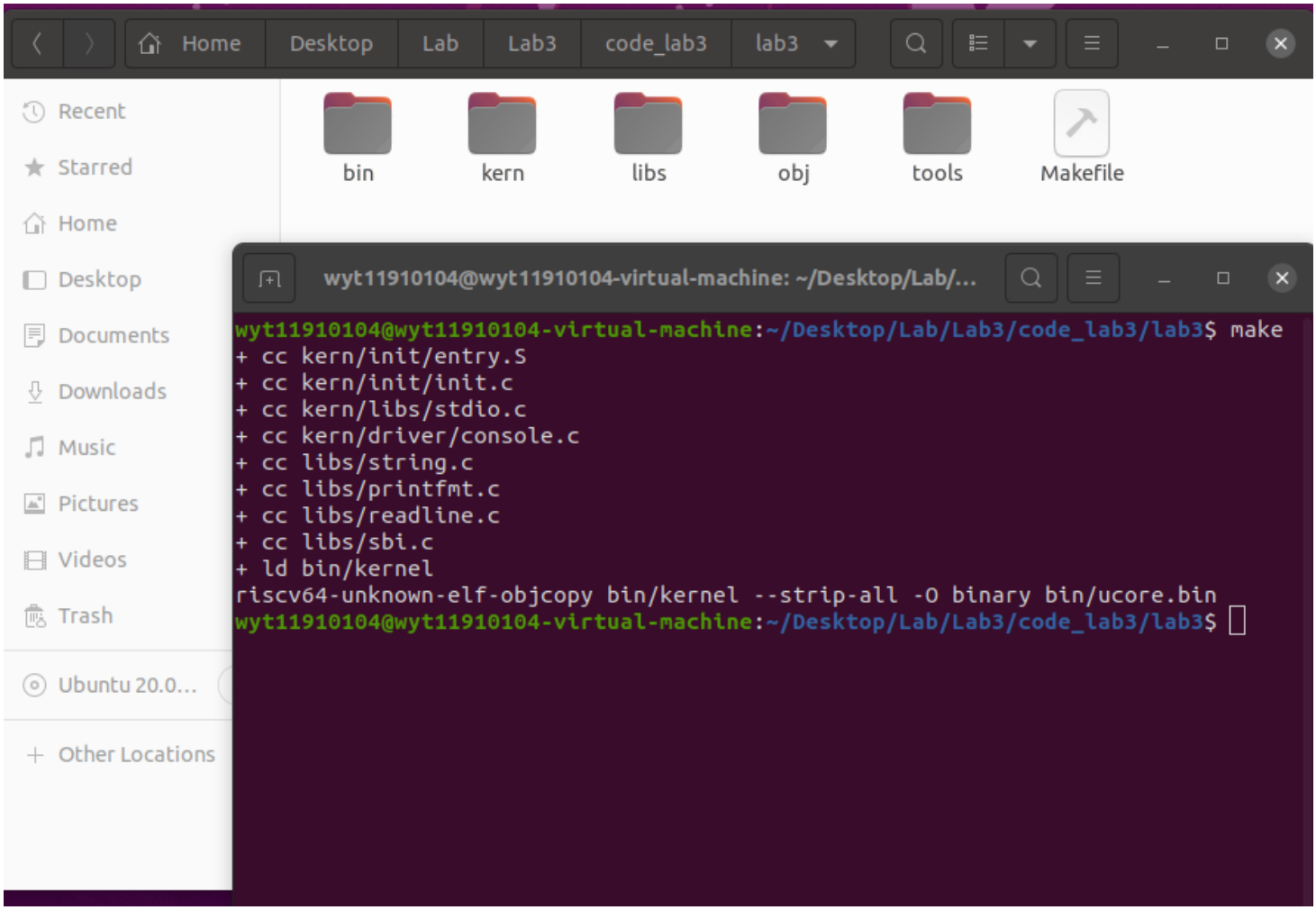
- Yining TANG(汤怡宁) 11811237@mail.sustech.edu.cn
- Yushan WANG(王宇杉) 11813002@mail.sustech.edu.cn

Q1: 最小化内核的启动过程

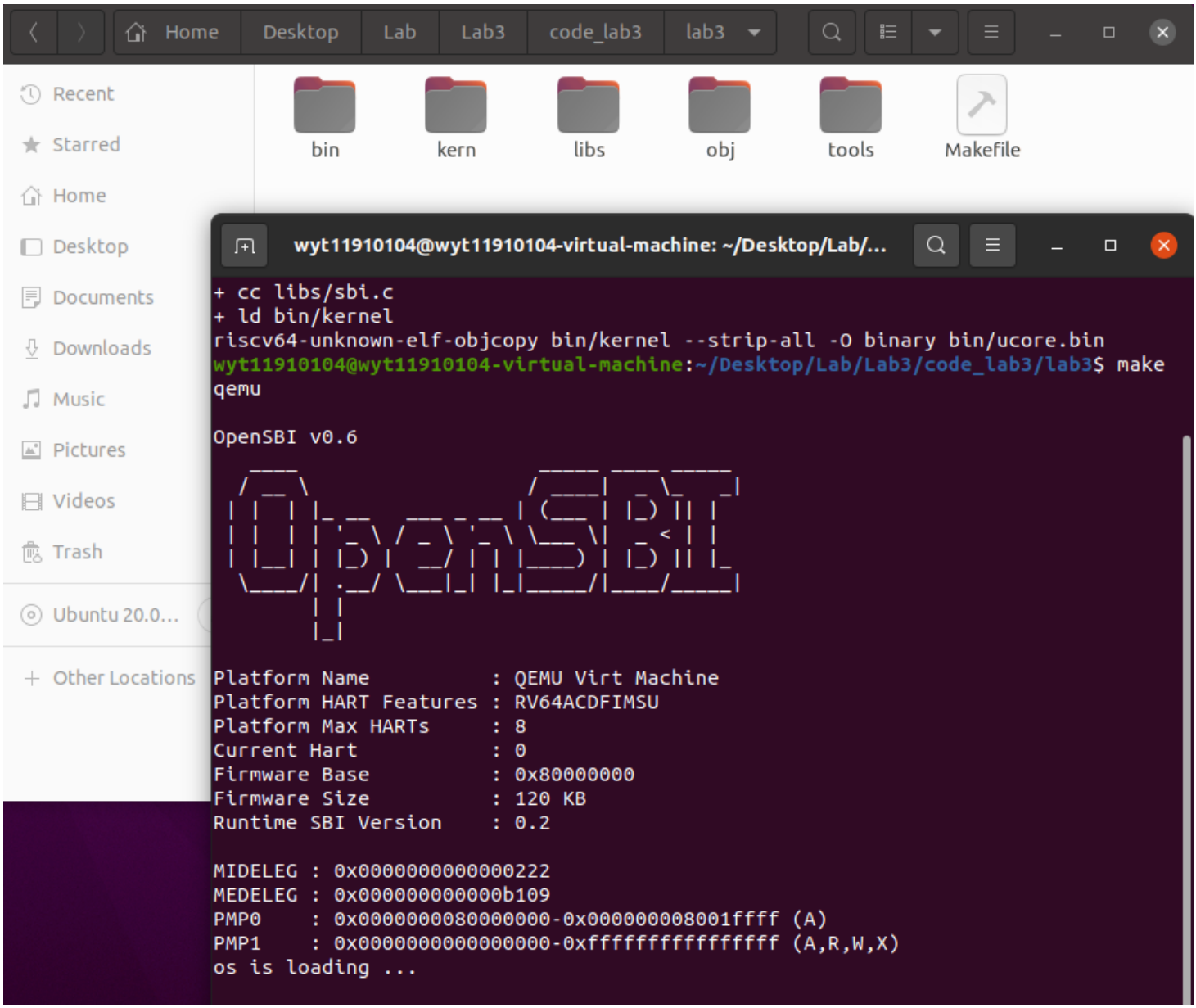
- 下载代码并解压到相关目录，然后切换目录到lab3下



- 在命令行使用 `make` 命令，生成模拟硬盘



- 使用 `make qemu` 命令，启动最小化内核



Q2: elf和bin文件的区别

- ELF: executable and link format, 包含符号表，汇编，调试信息等等，可以指定程序每个section的内存布局，不能直接运行，需要完整的操作系统来解析运行。OpenSBI不能直接运行elf文件。
- BIN：raw binary，只包含机器码，是将ELF文件中的代码段，数据段，以及其他自定义段抽取出来形成的一个内存的镜像。OpenSBI可以直接运行。

> Reference
> https://blog.csdn.net/chengf223/article/details/121639975
> http://blog.chinaunix.net/uid-25100840-id-2853463.html
> https://blog.csdn.net/soitis1121/article/details/8284425
> https://cloud.tencent.com/developer/ask/109491

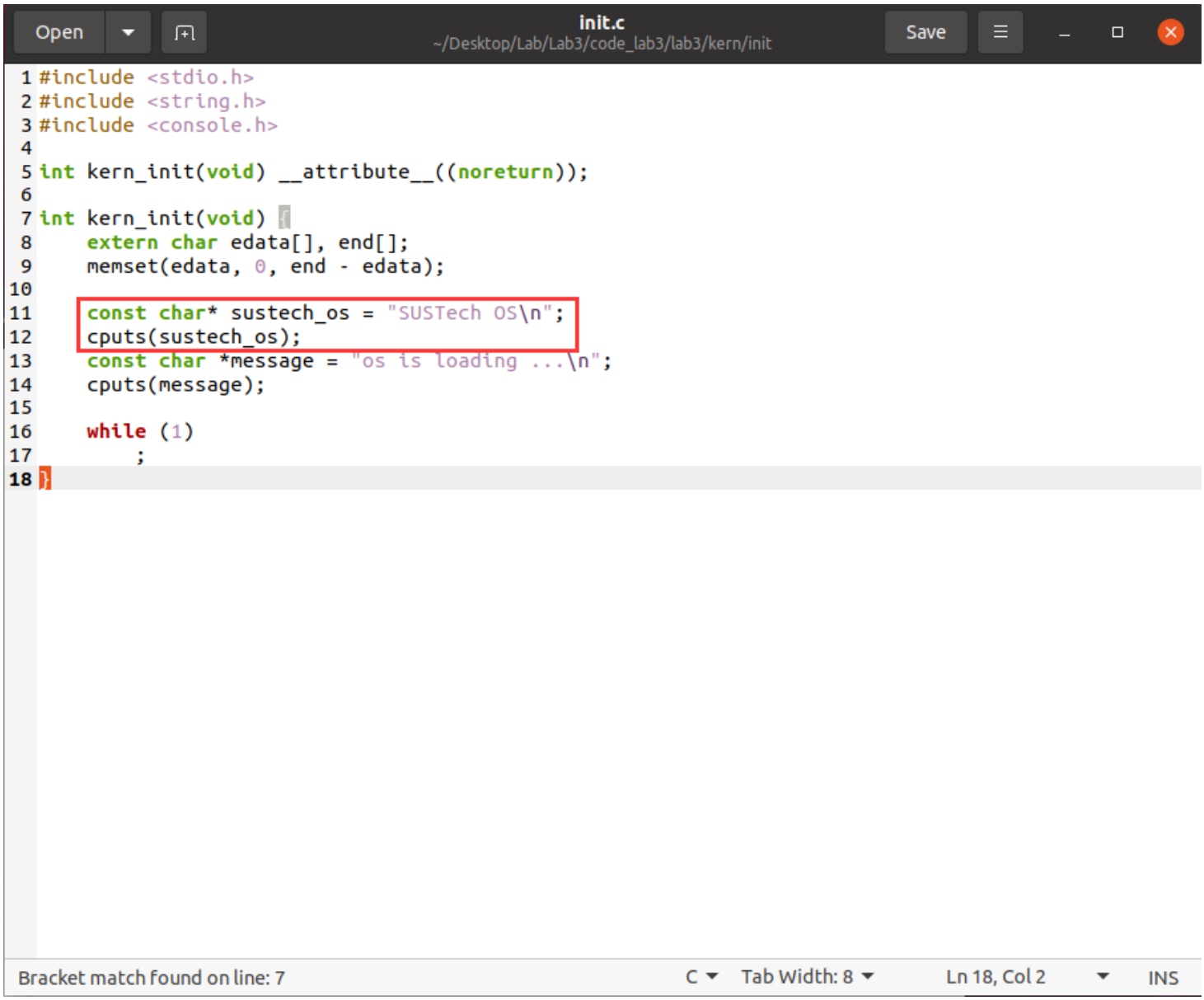
Q3: 链接脚本的作用

- 链接器：将输入文件链接成输出文件，可以将各种代码和数据片段收集起来组合成单一文件。链接过程可能发生在编译，内存加载，程序执行的时候。
- 链接脚本：描述如何将输入文件的section，映射到输出文件的section，并规定这些section的内存布局。链接脚本用于描述链接器处理目标文件和库文件的方式，如：
  - 合并各个目标文件的段
  - 重定位各个段的起始地址
  - 重定位各个符号的最终地址

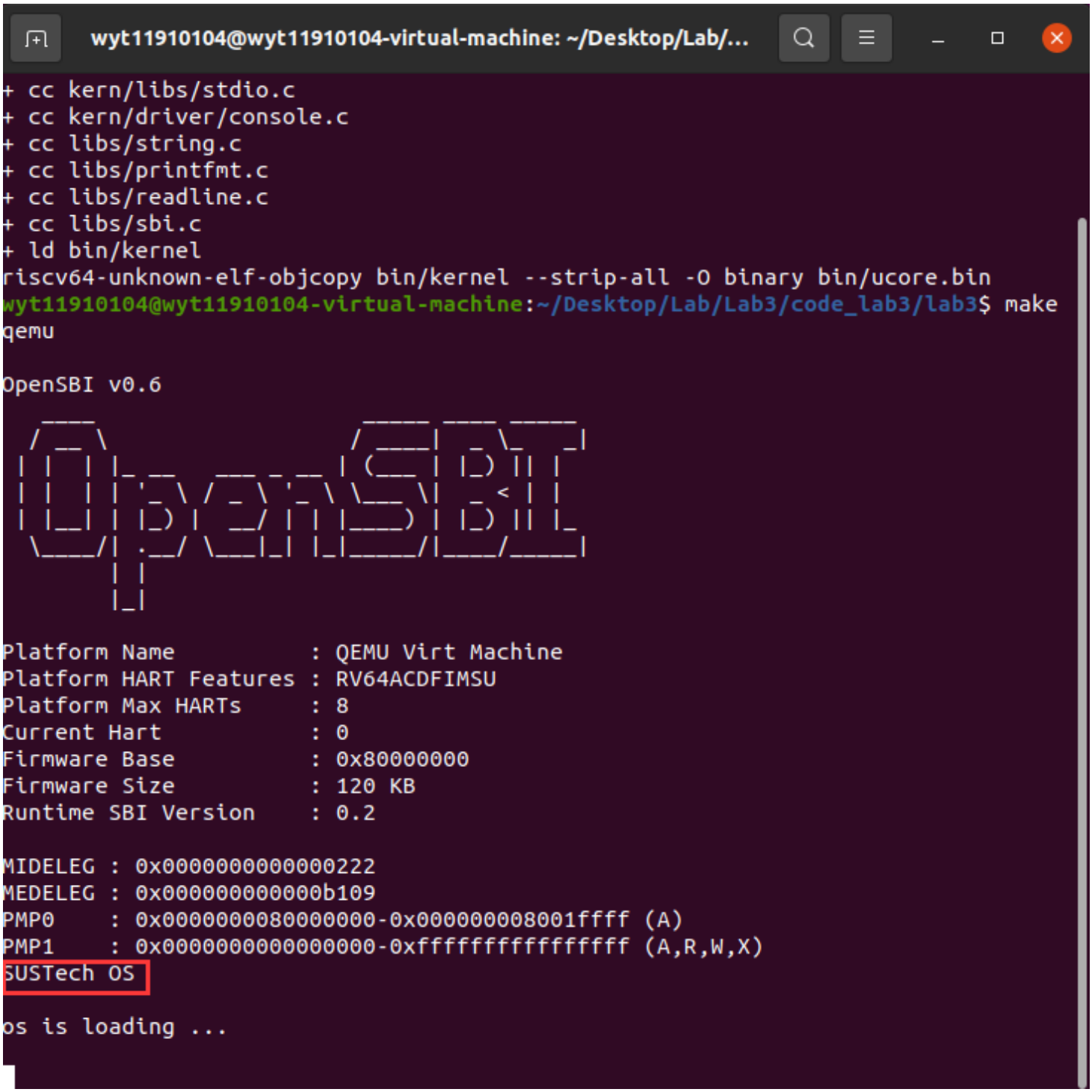
> Reference
> https://blog.csdn.net/ehuangdan5864/article/details/107744789
> https://www.yisu.com/zixun/5633.html

Q4: init.c打印特定字符串

- 打开 `kern/init/init.c` 文件，并添加打印 `SUSTech OS` 的相关语句。

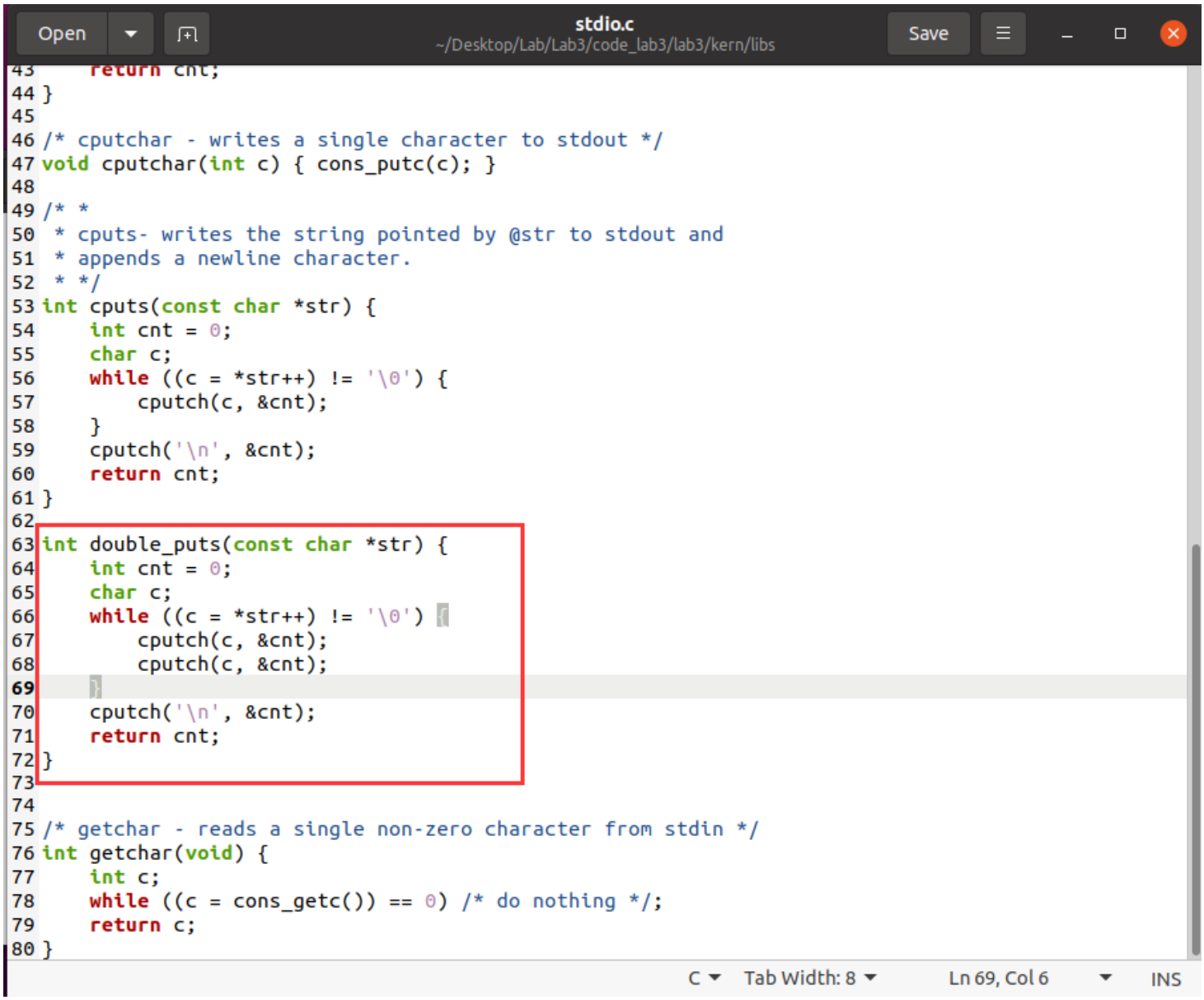


- 回到lab3目录下，使用 `make clean` 命令做清除操作。
- 再参考 `Q1` 的流程步骤，即可打印出 `SUSTech OS` 。



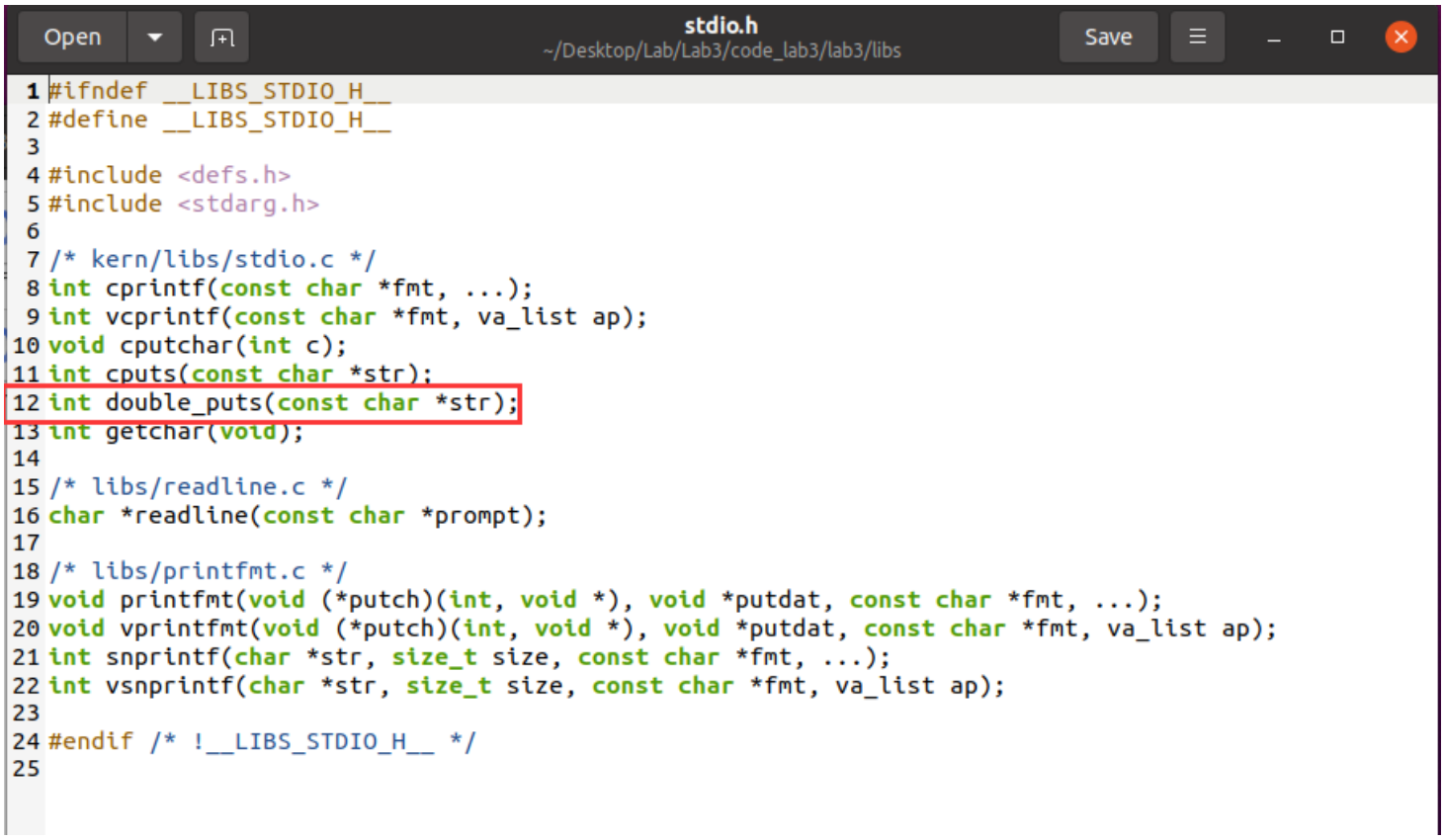
Q5: 参考cputs()函数实现double_puts()函数

- 打开 `lab3/kern/libs` 目录下的 `stdio.c` ， 添加 `double_puts()` 函数的具体实现。




```
43      return cnt;
44  }
45
46  /* cputchar - writes a single character to stdout */
47  void cputchar(int c) { cons_putc(c); }
48
49  /* *
50   * cputs- writes the string pointed by @str to stdout and
51   * appends a newline character.
52   * */
53  int cputs(const char *str) {
54      int cnt = 0;
55      char c;
56      while ((c = *str++) != '\0') {
57          cputch(c, &cnt);
58      }
59      cputch('\n', &cnt);
60      return cnt;
61  }
62
63  int double_puts(const char *str) {
64      int cnt = 0;
65      char c;
66      while ((c = *str++) != '\0') {
67          cputch(c, &cnt);
68          cputch(c, &cnt);
69      }
70      cputch('\n', &cnt);
71      return cnt;
72  }
73
74
75  /* getchar - reads a single non-zero character from stdin */
76  int getchar(void) {
77      int c;
78      while ((c = cons_getc()) == 0) /* do nothing */;
79      return c;
80  }
```

- 打开 `lab3/libs` 目录下的 `stdio.c` ， 添加添加 `double_puts()` 函数的声明。




```c
#ifndef __LIBS_STDIO_H__
#define __LIBS_STDIO_H__

#include <defs.h>
#include <stdarg.h>

/* kern/libs/stdio.c */
int cprintf(const char *fmt, ...);
int vcprintf(const char *fmt, va_list ap);
void cputchar(int c);
int cputs(const char *str);
int double_puts(const char *str);
int getchar(void);

/* libs/readline.c */
char *readline(const char *prompt);

/* libs/printfmt.c */
void printfmt(void (*putch)(int, void *), void *putdat, const char *fmt, ...);
void vprintfmt(void (*putch)(int, void *), void *putdat, const char *fmt, va_list ap);
int snprintf(char *str, size_t size, const char *fmt, ...);
int vsnprintf(char *str, size_t size, const char *fmt, va_list ap);

#endif /* !__LIBS_STDIO_H__ */
```

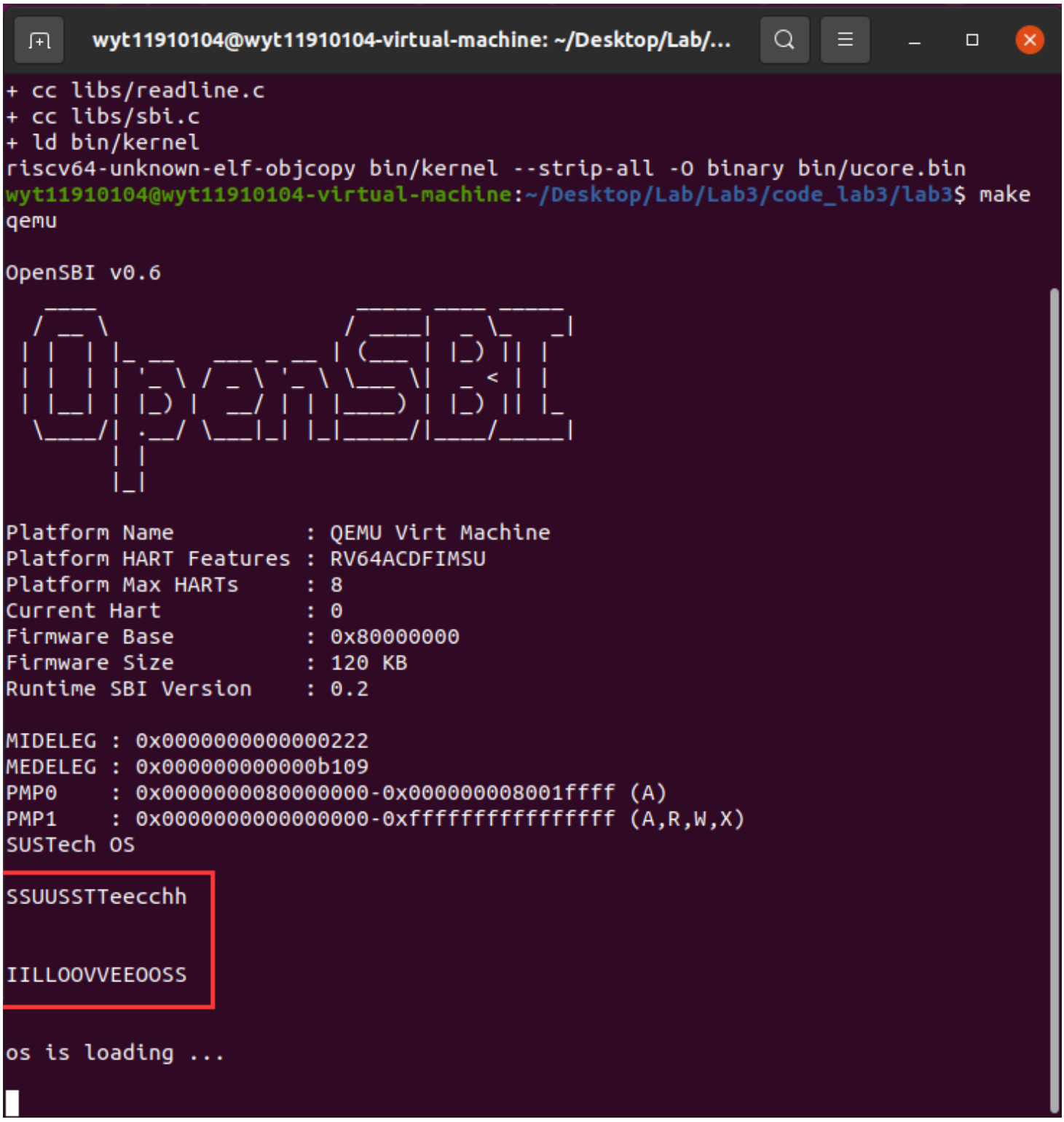
- 打开 `kern/init/init.c` 文件，并添加调用 `double_puts()` 函数以打印 `SSUUSSTTeecchh` 和 `IILLOOVVEEOOSS` 的相关语句。

Open | init.c ~/Desktop/Lab/Lab3/code_lab3/lab3/kern/init | Save

```c
#include <stdio.h>
#include <string.h>
#include <console.h>

int kern_init(void) __attribute__((noreturn));

int kern_init(void) {
    extern char edata[], end[];
    memset(edata, 0, end - edata);

    const char* sustech_os = "SUSTech OS\n";
    cputs(sustech_os);

    const char* sustech = "SUSTech\n";
    double_puts(sustech);

    const char* iloveos = "ILOVEOS\n";
    double_puts(iloveos);

    const char *message = "os is loading ...\n";
    cputs(message);

    while (1)
        ;
}
```

C | Tab Width: 8 | Ln 18, Col 26 | INS

- 回到lab3目录下，使用 `make clean` 命令做清除操作。
- 再参考 `Q1` 的流程步骤，即可打印出 `SSUUSSTTeecchh` 和 `IILLOOVVEEOOSS` 。


wyt11910104@wyt11910104-virtual-machine: ~/Desktop/Lab/...
+ cc libs/readline.c
+ cc libs/sbi.c
+ ld bin/kernel
riscv64-unknown-elf-objcopy bin/kernel --strip-all -O binary bin/ucore.bin
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab3/code_lab3/lab3$ make
qemu
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 : 0x0000000000000222
MEDELEG : 0x000000000000b109
PMP0 : 0x0000000080000000-0x000000008001ffff (A)
PMP1 : 0x0000000000000000-0xffffffffffffffff (A,R,W,X)
SUSTech OS
SSUUSSTTeecchh
IILLOOVVEEOOSS
os is loading ...