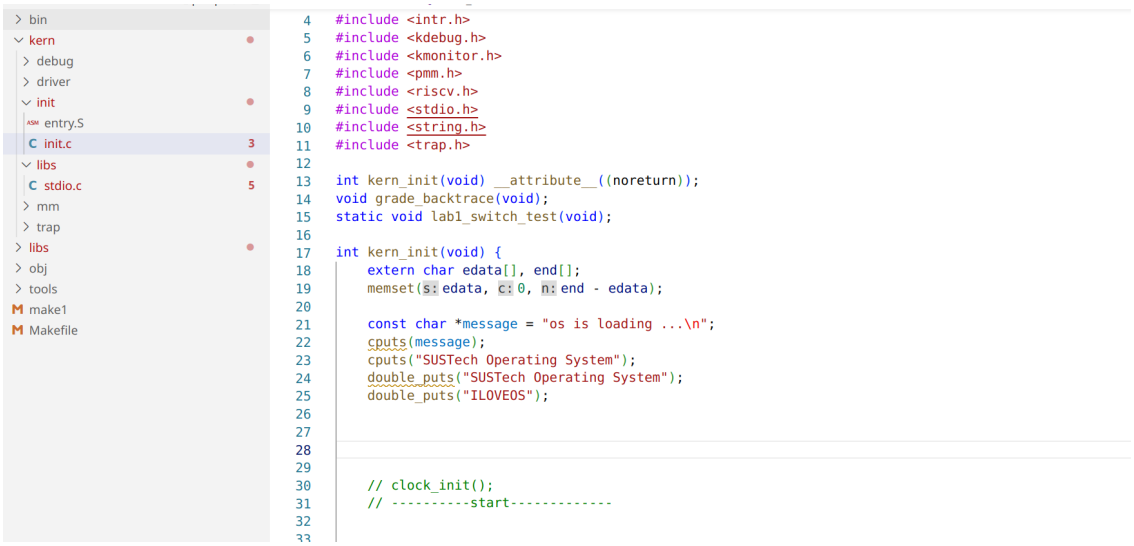
1. cpu电复位，跳转pc至 `0x80200000` 开始执行第一行代码， `.text`,`.kern_entry`被`linker`定位到`0x802000000`,初始化`stack pointer`，跳转到`kern_init`
2. `elf`是一种描述可执行文件的数据结构，`bin`文件是简单的`memory layout`
3. 链接脚本规定了`sections`和`labels`从哪里以及按照什么顺序进行排列
4. 5 一起做了




```
 * */
int cputs(const char *str) {
    int cnt = 0;
    char c;
    while ((c = *str++) != '\0') {
        cputch(c, cnt: &cnt);
    }
    cputch(c: '\n', cnt: &cnt);
    return cnt;
}

int double_puts(const char *str) {
    int cnt = 0;
    char c;
    while ((c = *str++) != '\0') {
        cputch(c, cnt: &cnt);
        cputch(c, cnt: &cnt);
    }
    cputch(c: '\n', cnt: &cnt);
    return cnt;
}
```

```
make: *** [Makefile:109: obj/kern/init/init.o] 错误 1
dgy@dgy-ubuntu ~/g/o/l/lab (master) [2]> make qemu
+ cc kern/init/init.c
+ cc kern/libs/stdio.c
+ cc kern/debug/kdebug.c
+ cc kern/debug/kmonitor.c
+ cc kern/debug/panic.c
+ cc kern/driver/clock.c
+ cc kern/trap/trap.c
+ cc libs/printfmt.c
+ cc libs/readline.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 : 0x0000000000000222
MEDELEG : 0x000000000000b109
PMP0    : 0x0000000080000000-0x000000008001ffff (A)
PMP1    : 0x0000000000000000-0xffffffffffffffff (A,R,W,X)
os is loading ...

SUSTech Operating System
SSUUSSTTeecchh  OOppeerraattiinngg  SSyyssttemm
IILLOOVVEEOOSS
QEMU: Terminated
```