

week3-lab-report

11911609-葛兆宁

1. 内核启动过程简述：qemu 会帮我们实现初硬盘外的硬件虚拟，而我们通过 lab3 的文件的 make 指令，通过 lab3 的程序实现了硬盘虚拟与操作系统的实现，接着，通过 makefile 里的 make qemu 指令使得 qemu 与我们模拟的硬盘相连，同时 qemu 会调用固件 opensbi，opensbi 会把 pc 放置到指定位置，将我们硬盘里的操作系统（ucore.bin）指令放置 pc 地址所指的内存中
2. 主要区别为 elf 的文件头较为复杂，里面有各个 section 的解析布局，bin 的文件头之后比较简单，只有解释自己应该被加载到的起始位置。
3. 链接脚本的作用是把输入文件(往往是 .o 文件)链接成输出文件(往往是 elf 文件)
4. init.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 *message = "os is loading ...\n";
    cputs(message);

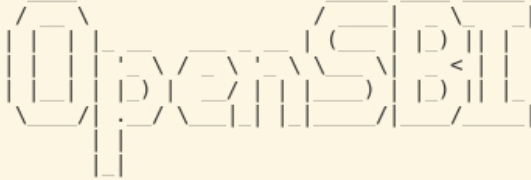
    const char *new_message = "SUSTech OS\n";
    cputs(new_message);

    new_message = "ILOVEOS\n";
    doubleputs(new_message);

    while (1)
        ;
}
```

```
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/lab3/code_lab3/lab3$ make
+ cc kern/init/init.c
+ ld bin/kernel
riscv64-unknown-elf-objcopy bin/kernel --strip-all -O binary bin/ucore.bin
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/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    : 0x0000000000000000-0xffffffffffffff (A,R,W,X)
os is loading ...
```

SUSTech OS



5.
stdio.c:

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

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

stdio.h:

```
#ifndef __LIBS_STDIO_H__
#define __LIBS_STDIO_H__

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

/* kern/libs/stdio.c */
int cprintf(const char *fmt, ...);
int vprintf(const char *fmt, va_list ap);
void cputchar(int c);
int cputs(const char *str);
int doubleputs(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__ */
```

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

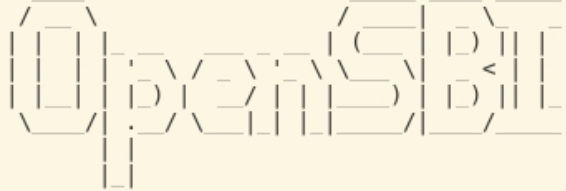
    const char *new_message = "SUSTech OS\n";
    cputs(new_message);

    new_message = "ILOVEOS\n";
    doubleputs(new_message);

    while (1)
        ;
}
```

```
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/lab3/code_lab3/lab3$ make
+ cc kern/init/init.c
+ ld bin/kernel
riscv64-unknown-elf-objcopy bin/kernel --strip-all -O binary bin/ucore.bin
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/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)
os is loading ...
```

SUSTech OS

IILL00VVEE00SS