Lab1实验报告

习题题解

习题一

使用`memset(edata, end - edata, 0)`语句把这段位置清空，完成bss段的初始化。

习题二

调用console.c中的`console_init()`函数即可完成console的初始化。

使用console.c中的`cprintf("hello, world!")`在控制台中打印"hello, world"。

代码解释

编译流程

`make`会先通过gcc编译出kern目录下的内核程序的.s文件，之后再和entry.S一起链接，得到kernel8.elf。由于操作系统启动时无法加载elf头的内容，需要用objcopy去除elf头。

在elf头的.txt段中entry.S被放于最前的位置，运行时会先完成cpu和Exception Level的相关设置，之后再执行*.c的代码。

习题一

在linker.ld中，bss段起始位置被保存在edata中，终止位置被保存在end中。bss段存的是声明未定义的静态变量，这时它们的默认值为0。只需要直接把bss段赋为0。

`memset(void *str, int c, size_t n)`：注意由于接口的地址是按字节对齐的，故可以用`char*`作为的指针变量，但不能用`int*`。

习题二

console.c自带`console_init()`负责完成控制台的初始化。由于交互是通过串口传输完成的，初始化控制台即初始化串口。`console_init()`调用了uart.c中的`uart_init()`程序来完成串口的初始化。

"hello world"通过console.c自带的`cprintf()`来完成输出。`cprintf()`函数调用格式处理程序`vprintfmt()`并通过库函数`putch()`向控制台打印。