**lab3实验报告**

**李卓 Pb19000064**

**实验目的**：  
1. 实现简单的 shell 程序，提供 cmd 和 help 命令，允许注册新的命令。

2. 实现中断机制和中断控制器 i8259A 初始化。

3. 实现时钟 i8253 和周期性时钟中断。

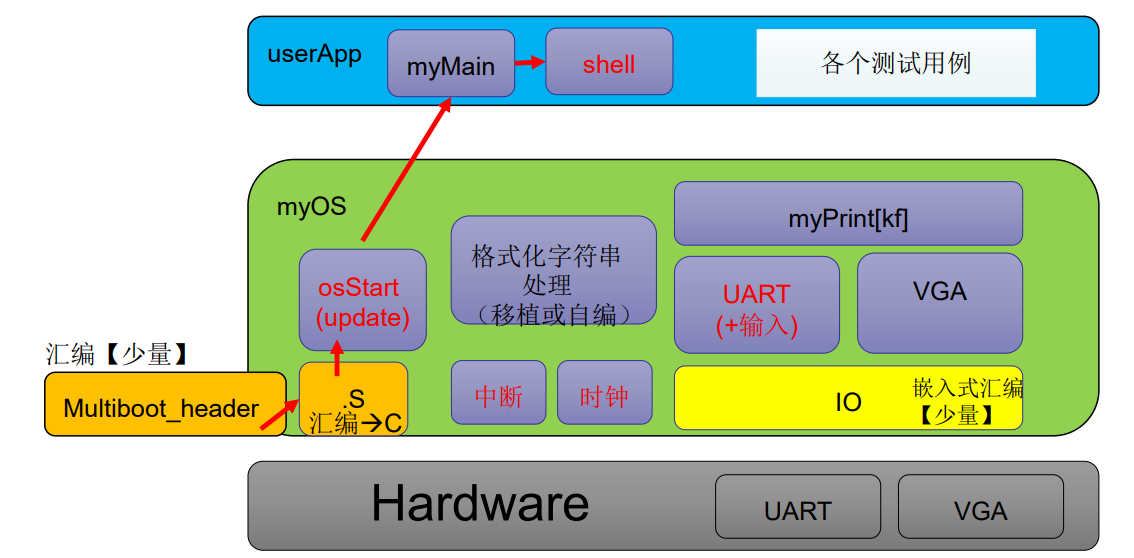
4. 实现 VGA 输出的调整：

– 左下角：时钟中断之外的其他中断，一律输出“Unknown interrupt1” 。

– 右下角：从某个时间开始，大约每秒更新一次，格式为：HH:MM:SS 。

5. 提供脚本完成编译和执行

**实验环境：**  
1. PC 一台 2. Windows 系统 3. Ubuntu 4. QEMU

**实验流程：**  
****

multiboot\_header 为系统启动部分，系统启动后进入 myOS，在 osStart.c 中调用 myMain.c 进入 userApp 部分。若 userApp 部分，myMain 及其调用的程序需要输出字符串 时，通过 myPrintk/f，将经 vsprintf 处理后的字符串通过 IO 接口，在 VGA 和 UART 上完成输出。在 myMain 中，调用 startShell 函数，进入 shell。

1. 在 multiboot\_header 中完成系统的启动。

2. 在 start32.S 中准备好上下文，最后调用 osStart.c 把进入 c 程序。

3. 在 osStart.c 中完成清屏，初始化 8259，初始化 8253，设置 WallClock 初始时间等初始化操作，调用 myMain， 进入 userApp 部分。

4. 运行 myMain 中的代码，进入 shell 程序。

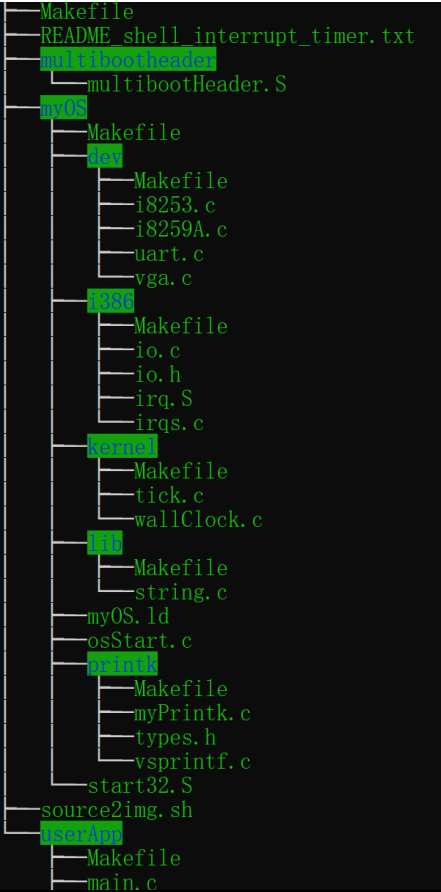
5. 等待用户输入命令，并对命令进行处理

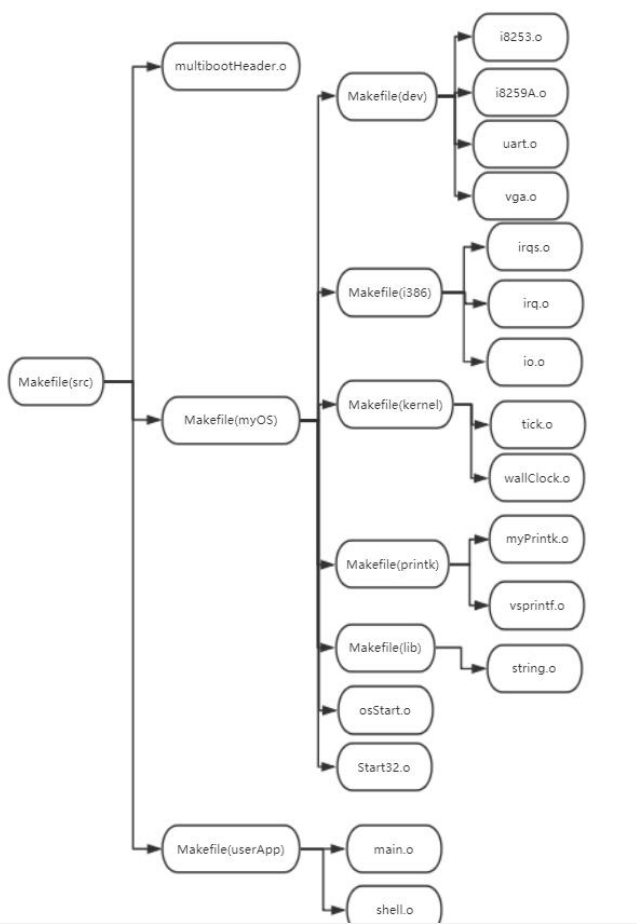
**主要功能：**

为中断描述符表分配(IDT)一块内存，并将所有中断处理程序初始化为合适的缺省处理函数， ignore\_int1，调用 ignoreIntBody，在屏幕左下角，输出“Unknown interrupt1”。 初始化寄存器 IDTR。初始化中断控制器 PIC i8259，调用 enable\_interrupt 函数，启动中断。

时钟中断会调用 tick 函数，tick 函数中的计数器初始置为 0，每次调用时，计数器加一，当调用满一百次时，调用 hook 函数，更新时钟。

在 help 命令处理函数中，判断 help [cmd]中的 cmd 是否为已定义命令，则调用该命令的 帮助函数，否则，输出错误信息。 在 cmd 命令处理函数中，执行其函数。

**代码组织布局：  
**

****

从物理内存 1M 的位置开始放代码和数据，前面 12 个字节为 multiboot\_header,向后对齐 8 个字节，放代码。再向 后对齐 16 个字节，用于放初始化的数据（数据段）。在数据段之后，再向后对齐 16 个字节。之后为 BSS段,用于存放程序中未初始化的全局变量和静态变量。并在 BSS 段后，再向后对齐 16 个字节。剩余部分为 堆栈段。

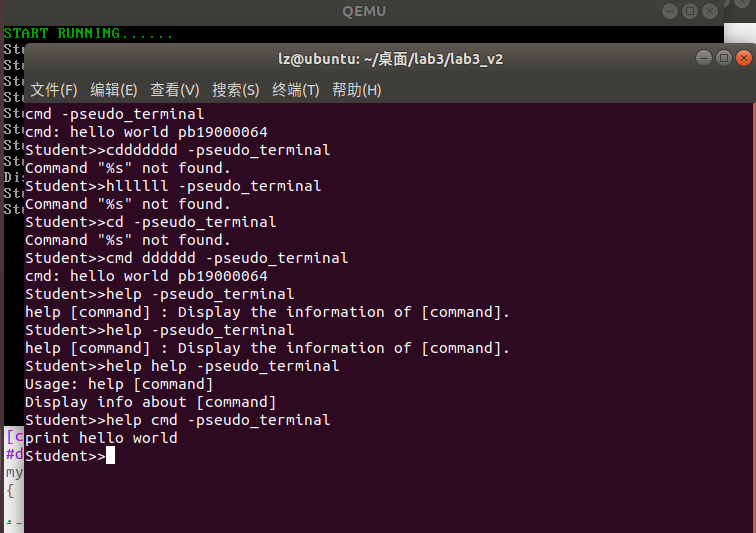
**编译过程说明：**chmod 777 source2run.sh

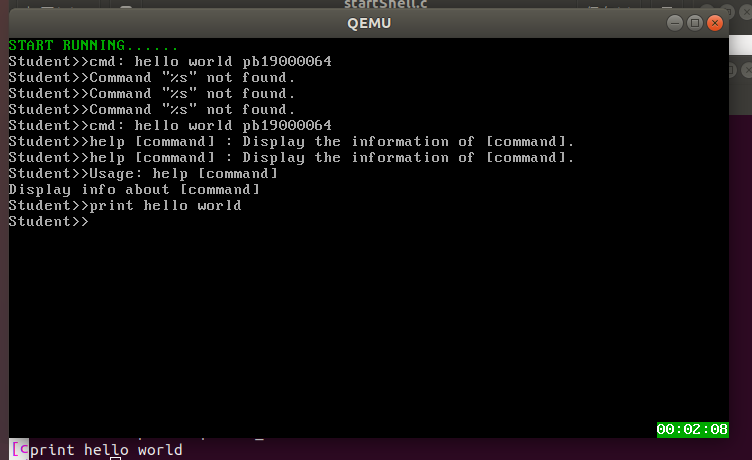
./source2run.sh

sudo screen /dev/pts/1

**输出结果：**

助教给出的myprintk没办法格式化输出， 所以输出的字符串中存在’%s’

****

****