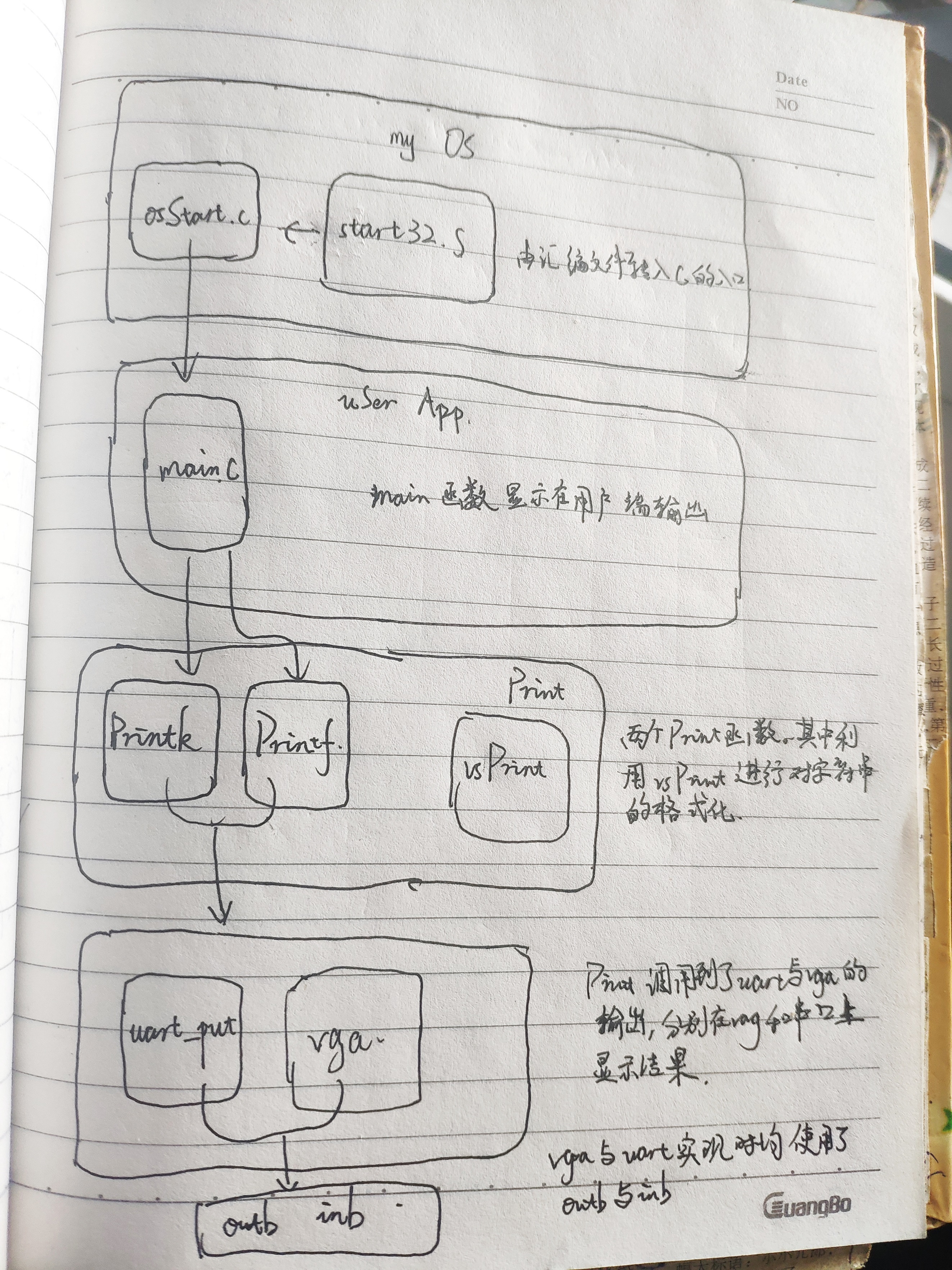
**操作系统实验2报告**

**软件框图**



**主流程及其实现**

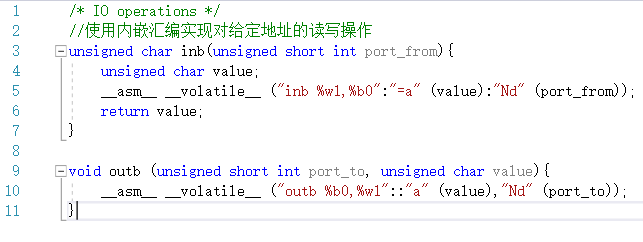


**主要功能模块及其实现**

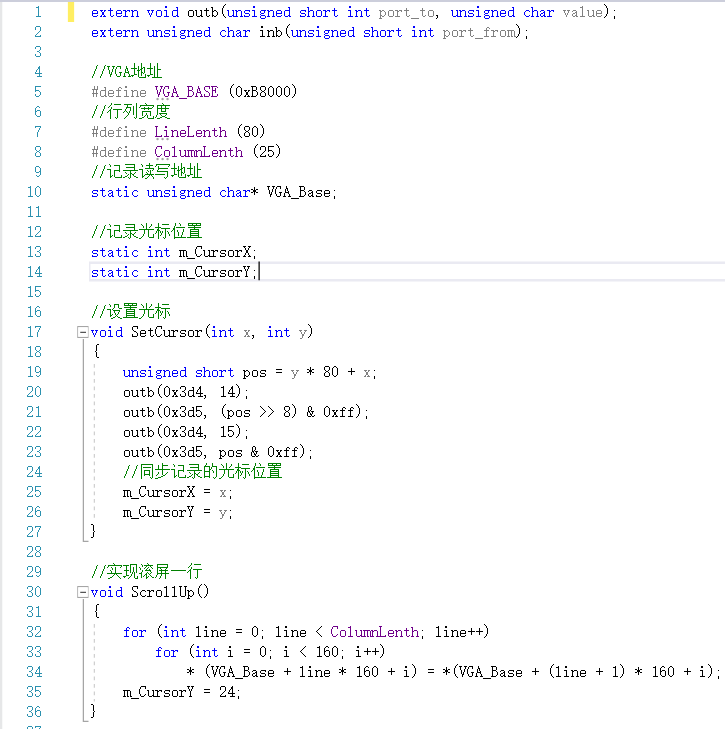


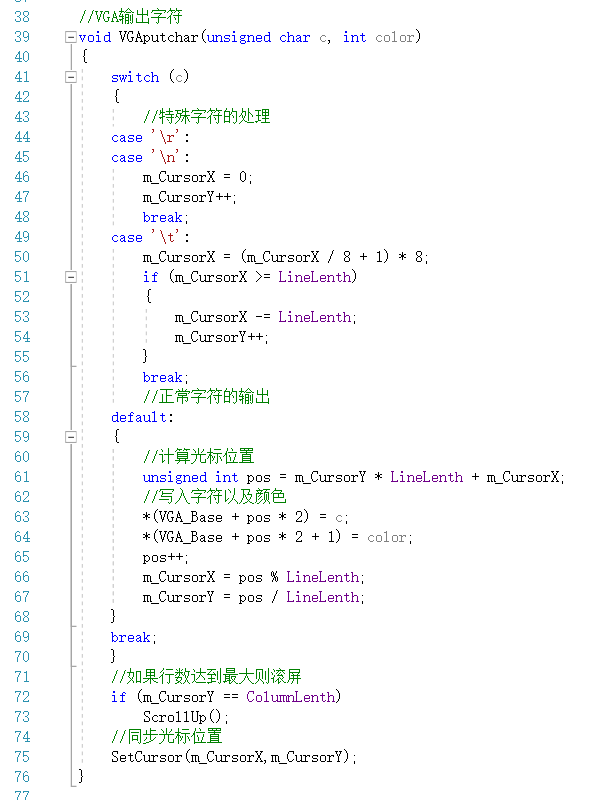
**源代码说明**

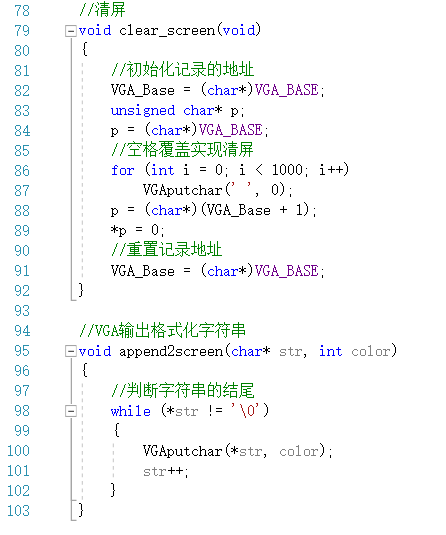
在本次实验的根目录下，除了配置了makefile与DS文件，还有4个文件夹：multibootheader、myOS、output、userApp 。multibootheader已经不陌生，其中放置了有关multibootheader协议的S文件。在output中，原本为空文件夹，用来输出我们编译后生成的文件。在该文件夹中的目录结构与根目录相似，主要是为了对每个文件输出时不产生混淆。在userApp中存放的是main.c文件与该文件夹下的makefile文件，Main文件是用来测试本次实验的功能的入口。在myOS文件夹中存放了本次实验的主要源文件。直接存放在此该目录下的文件有DS文件、该文件夹下的makefile文件、链接器文件、一个配置地址的S文件和一个包含了main函数的与multibootheader中的S文件相关联的C文件。这个文件也是从汇编到C的转变。在该目录下则有3个实现相关功能的文件夹：dev、i386、printk。每个文件夹下都有一个相应的makefile文件以及实现功能的C语言源文件。I386文件夹下是实现IO（inb、outb）的源文件，dev下存放的是在uart与vga上实现给定字符串并输出的功能的源文件，而在printk中则是实现printf与printk的源文件。每个子目录下的makefile文件的输出目录都是在output中的同名目录。相关源代码如图：



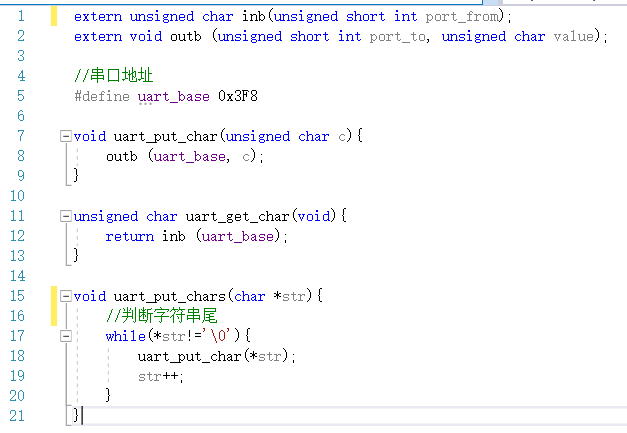
实现VGA功能的代码







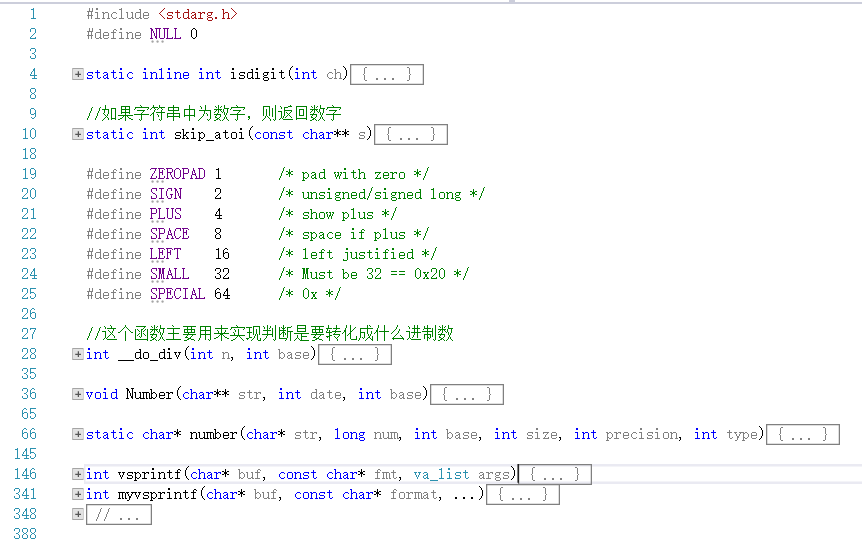
实现uart读写的代码



myprint



移植的字符串格式化代码的概览

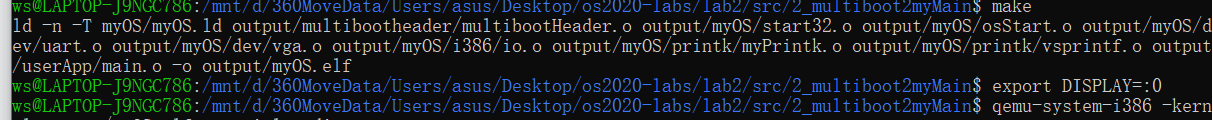


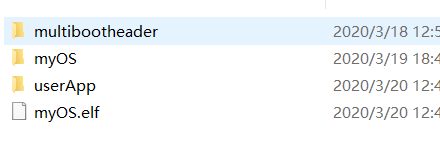
**代码布局说明**

所有的引导模块将按页（4KB）边界对齐，物理内存地址从1M处开始

**编译过程说明：**

在Ubuntu中先搜索到lab2的目录，然后通过指令make完成编译，可以看到在output目录中的对应目录中分别输出了与根目录下对应文件相同文件。

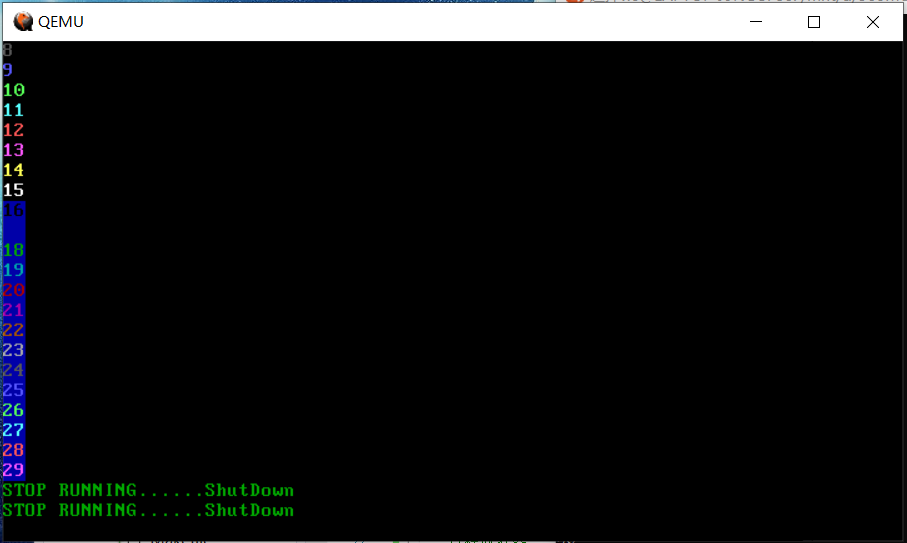




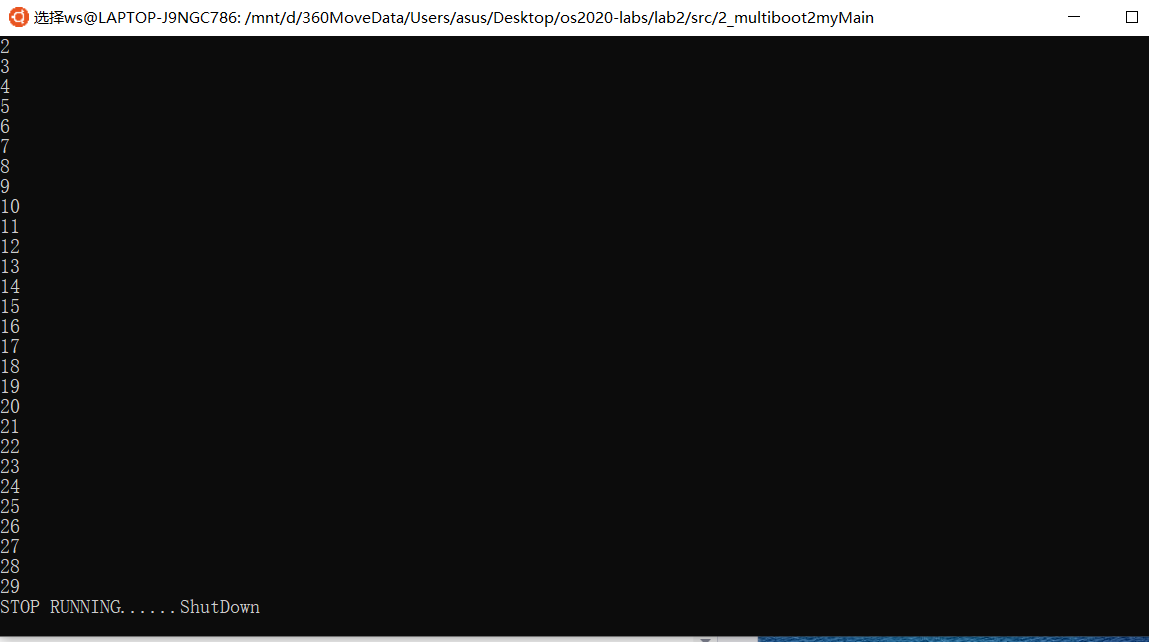
可以看到，make命令确保源代码目录下没有不正确的.o文件以及文件的互相依赖。它们分别链向源代码目录下的真正的i386所需要的真正的子目录。编译后产生的文件在图上可见，有multiboot、start32、osStart、uart、vga、io、myPrintk、vsprintf、main、myOS的输出文件以及标注了他们的输出目录。

**运行和运行结果说明：**

在Ubuntu中通过QEMU启动已经编译生成的bin文件，得到Linux的图形化界面运行结果，显示需要的输出。其中VGA输出结果是1-29个数字并以与该数字对应的编码的颜色渲染，同时在前后各有执行的开始与结束说明。



在linux命令窗口的显示结果则不带有颜色，但是输出内容与前者相同。



**遇到的问题和解决方案：**

1. 不会自己编写字符串的格式化函数。

通过网络工具移植。

1. 不了解如何输出到VGA上。

咨询助教并选择了直接用C语言的指针进行修改值

1. 不知道光标的实现方法。

参考老师给出的文件并且查阅网络资料完成。