**操作系统实验5报告**

**软件框图**

**主流程及其实现**

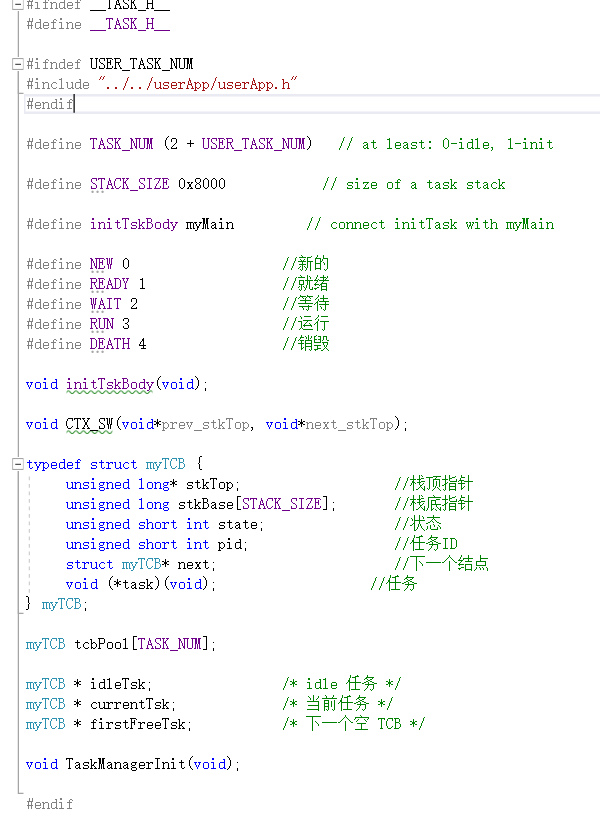
在上次实验的基础上，增加下列内容。

**主要功能模块及其实现**

本次实验中要实现的功能模块分别按五个部分实现

**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文件、memTestCase文件、userTasks文件与该文件夹下的makefile文件，本次实验中的userTasks是提供的测试Task相关功能文件。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6个实现相关功能的文件夹：dev、i386、printk、kernel、lib、include。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。本次实验中在lib中添加了task.h文件，主要功能时任务池的空间声明与TCB数据结构的定义，特别的，我在lib中自行添加了颜色宏的头文件，方便用宏定义写入myPrint的颜色参数。kernel文件夹中增加task.c源代码文件以实现task的相关功能。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。下面展示相关源代码。



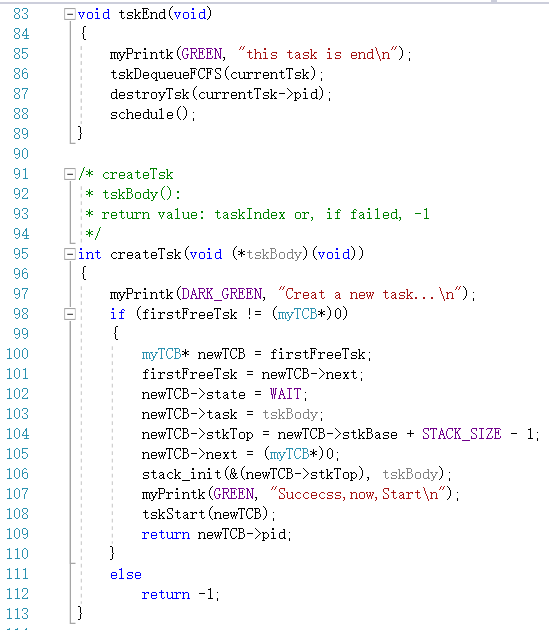
task.h代码，主要是自行定义了TCB数据结构的内容，同时还定义了若干task状态宏



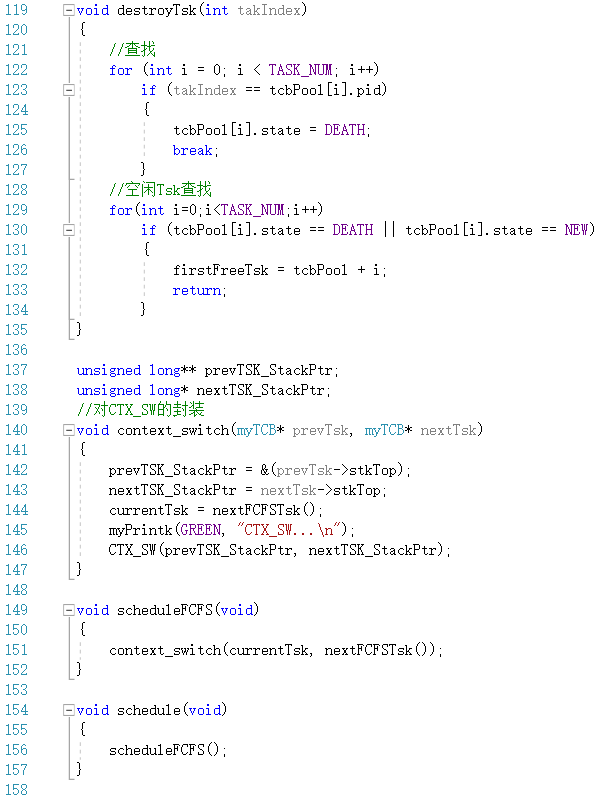
task.c代码部分1，主要参考了教师展示的代码，实现了判断队空、求下一task、初始化等功能。



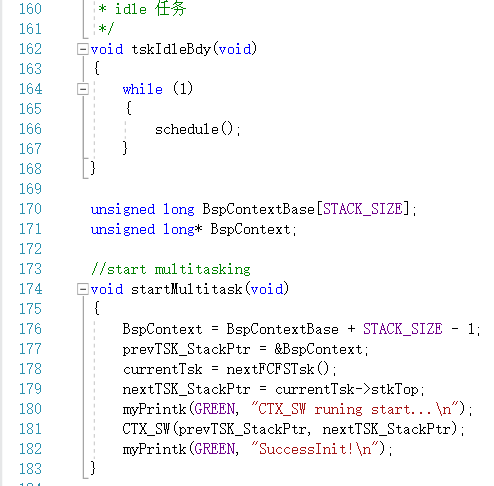
task.c代码部分2，实现出入队以及栈的初始化、任务的开始



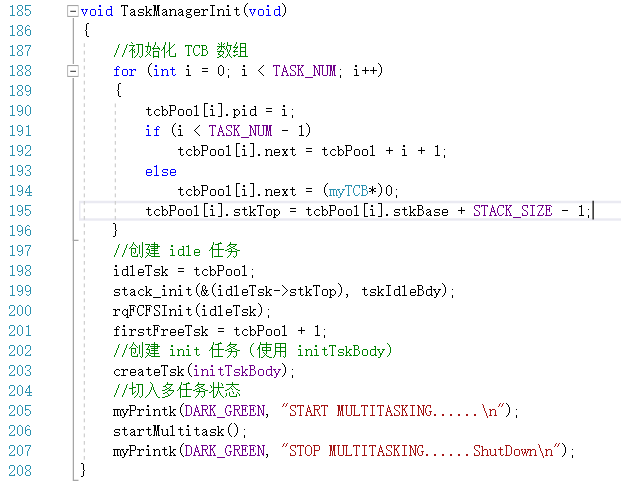
task.c代码部分3，主要实现任务的结束与创建



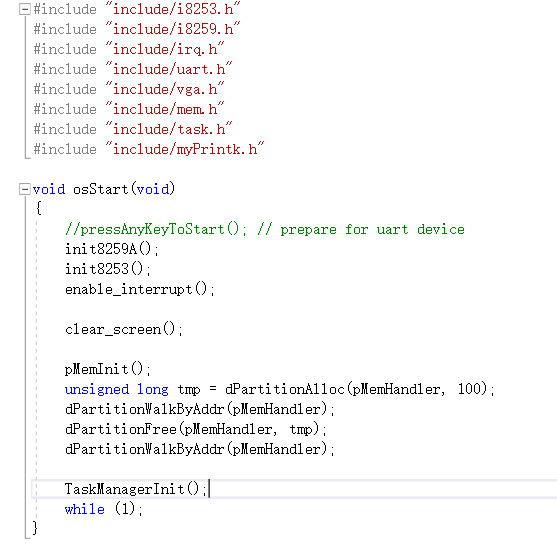
task.c代码部分4，主要实现任务的销毁、封装实现的上下文切换与调度



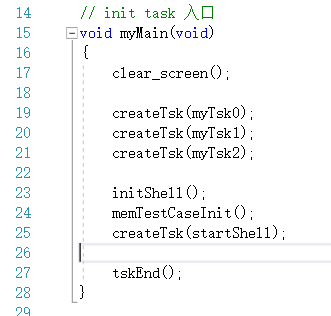
task.c代码部分5，主要实现了idle任务和多任务模式



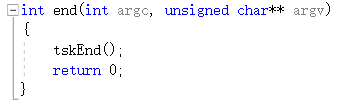
task.c代码部分6，主要实现任务管理的初始化



osStart代码部分



main代码部分，将Shell作为一个Task



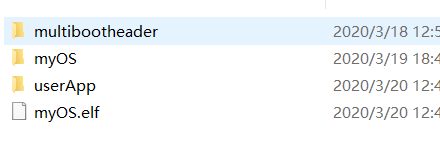
除了讲义要求的必须部分，我还实现了一个end指令，其作用即在Shell中执行tskEnd()

**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。本次实验中在封装上下文切换的函数时修改了current\_task和prev/next\_task的信息。

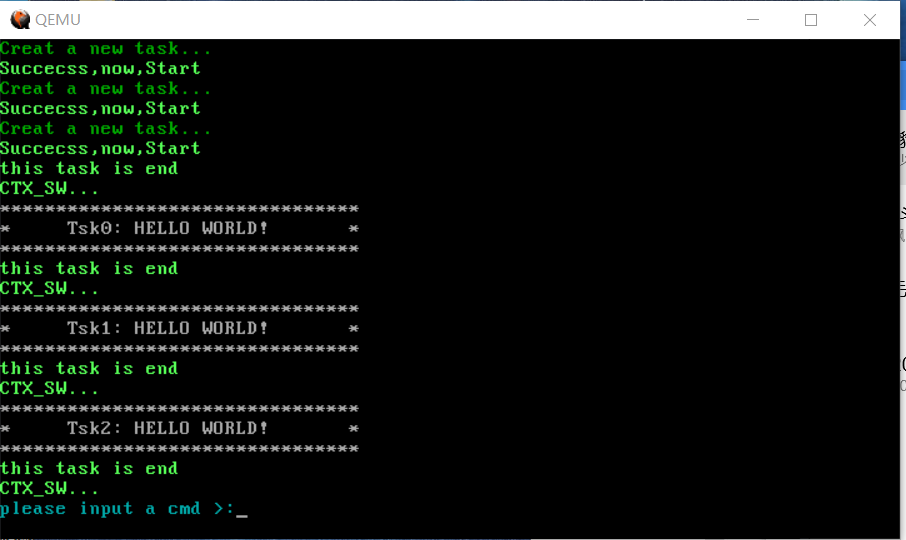
**编译过程说明：**

在Ubuntu中先搜索到lab4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。

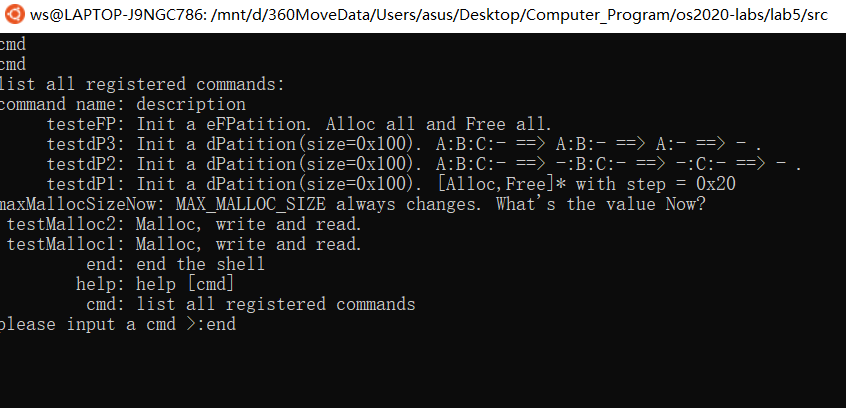


**运行和运行结果说明：**

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，然后再通过Ubuntu启动一个交互界面，用于输入与输出。



刚运行内核时的QEMU界面。



在运行了end命令后，QEMU界面退出。

**遇到的问题和解决方案：**

1. 不能理解任务管理的具体实现方式

通过网络工具查询和了解不，然后通过借鉴与咨询助教、参考教师所给示例实现。

1. 无法正常对上栈

咨询助教后了解到是没有给栈指针赋给空间，遂将指针成员变量改为数组形式实体空间的自动分配。