

## 任务栏

在GUI当中实现了任务栏，包括开始按钮和系统时钟两个功能。

### 系统时钟

实现了系统的时钟功能，精确度达到秒级，可以显示当前BIOS的系统时间。

#### 实现方案

在实模式内部通过调用BIOS时钟中断获取系统启动时的时间，在系统进入保护模式后通过定时器对时钟进行维护。维护的精确度达到秒。方案思路符合目前主流操作系统对时钟采用的思路。

#### 相关文件

Bootpack.c - 用于维护时钟在系统启动后的运行和显示

Asmhead.nas - 用于读取和保存BIOS信息

### 开始按钮

通过对鼠标位置的监控和鼠标点击的操作的监控实现了开始按钮和开始菜单的功能。在本项目当中，才开始菜单内仅添加了一个命令行的图标以作演示之用。

#### 实现方案

在鼠标进行点击操作时，判断鼠标的位置信息。如果鼠标所处位置在于开始按钮之上，并且在其上不再有其他的图层覆盖，则判定开始按钮被选中，从而执行相关操作。

当开始菜单打开后，当鼠标进入开始菜单的内部时开始监控鼠标的运动行为，通过对于鼠标位置的计算获取其选中的项目。当鼠标点击时获取鼠标点击位置执行相应的程序。

整个的实现方案涉及对图层和鼠标的较为复杂的操作。

#### 相关文件

Bookpack.c - 更改对于鼠标操作的监控和响应

Sheet.c - 更改图层显示的相关细节

Window.c - 绘制相应的按钮和菜单

Graphic.c - 绘制相应的按钮和菜单

## 项目小结

这次项目让我受益匪浅，使我可以从一个系统开发者的角度重新认识操作系统，解开了多年以来对于操作系统的种种疑惑，认识到开发一个类似Windows这样的大型复杂功能强大的操作系统是多么的困难，使我对于微软的工程师们有了更加崇高的敬意。

更重要的是，通过这次项目，我更加深刻地了解了系统对于输入输出设备的操作。比如，界面的绘制就是直接写显存，而鼠标和键盘的操作则是通过PIC中断进行的。通过对于设备的管理、底层代码的编写，增进了我对于操作系统的理解，特别是引导、内存管理和进程调度，也增强了编码能力，还学习到了汇编和C语言的混合编译。

总之，本次项目令我收获颇丰。