

操作系统课程设计项目说明

2011 级软件工程
1152691 李志浩
1152730 李博一

项目名称：LOS

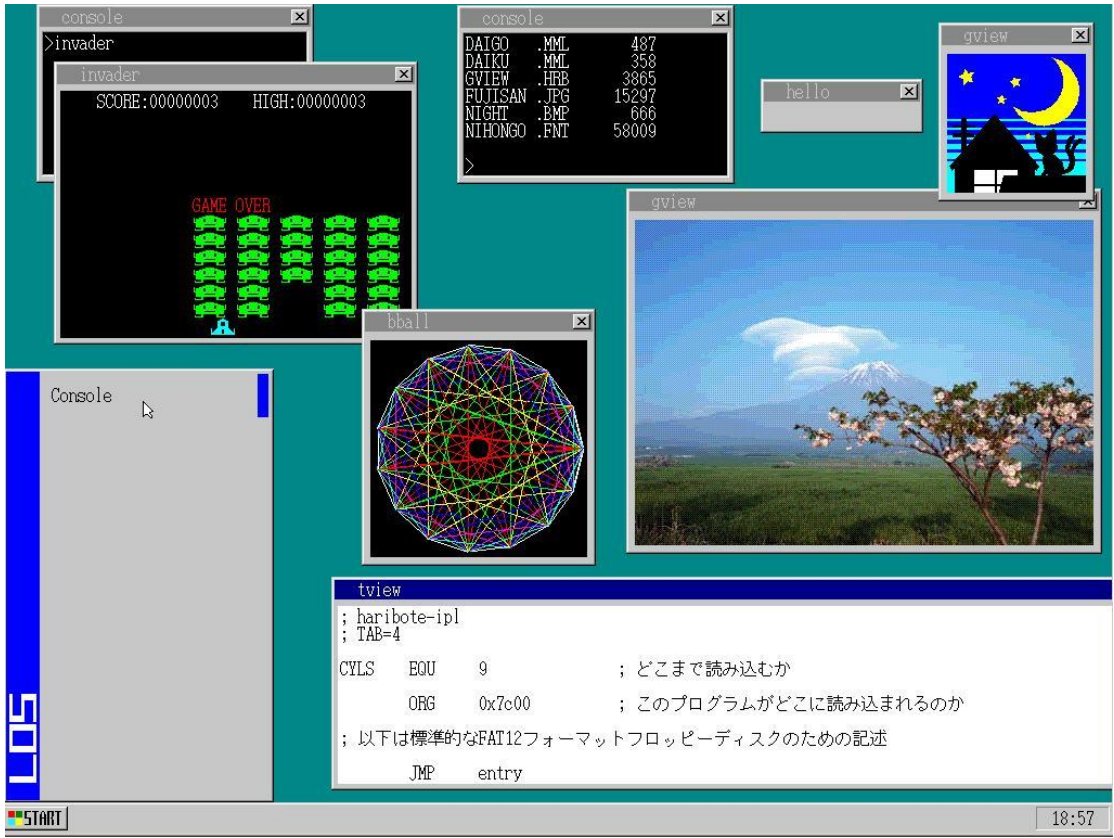
目录

- 操作系统课程设计项目说明1
- 项目名称：LOS.....1
- 开发环境及语言.....2
- 项目示例图2
- 项目添加功能描述3
 - 任务栏3
 - 系统时钟3
 - 实现方案3
 - 相关文件3
 - 开始按钮3
 - 实现方案3
 - 相关文件4
 - 多级反馈队列.....4
 - 实现方案4
 - 相关文件4
 - 命令行导航功能4
 - 实现方案4
 - 相关文件5
- 项目小结6

开发环境及语言

Windows XP/7/8 nask（改编自汇编语言） c 语言

项目示例图



项目添加功能描述

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

多级反馈队列

任务调度算法采用了多级反馈队列。同一层采用轮转法实现。

实现方案

当有新的进程加入时，加入到优先级最高的队列中，保证其优先执行。在进行上下文切换时获得将要被切换的任务，若任务未完成，则将其从当前队列中移出，然后对其所处队列进行调整，并对应更改其优先级，最后再加入到新的层中。每一次的调度采用轮转法实现。

相关文件

Bookpack.c	- 有新的进程加入时，默认添加到优先级最高的队列中
Mtask.c	- 更改进程切换算法
Naskfunc.nas	- 进程上下文切换调用

命令行导航功能

实现了命令行中上下键的导航，使得命令行更人性化。

实现方案

每个命令行是一个进程，对于每一个打开的命令行在内存中开辟一块空间用于存储用户的输入记录，通过队列的方式存储于自设的数据结构中。在用户键入

↑或↓时获得其扫描码并完成队列的存取，显示到命令行中。

相关文件

Bookpack.h	- 定义命令行记录的结构体，Struct CMD 和 CMD_BUF
Bookpack.c	- 完成对↑和↓键扫描码的读取和分析，进行处理
Memory.c	- 内存分配与释放
Console.c	- 实现对结构体中数据的读取和命令行显示的更改

项目小结

这次项目让我们受益匪浅，使我们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，使我们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，我们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，增进了我们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，本次项目令我们收获颇丰。