## 进程管理

### 需求分析

1. CPU：模拟中央处理器，用于解释执行“可执行文件”里面的指令。

x=? 给x赋值

x++ x加1

x-- x减1

!A? 占用A设备？时间。

寄存器模拟：

PSW，IR，PC，DR

中断模拟：

①进程结束②时间片到 ③I/O中断

1. PCB：

包括进程标识符、主要寄存器、进程状态、阻塞原因。三个队列：空白PCB队列，就绪队列和阻塞队列。

1. 进程调度：
   1. 采用时间片轮转法，时间片为5
   2. 将正在运行进程的现场（寄存器组）保存在该进程的 PCB 中；从就绪队列中选择一个进程；将这个进程的 PCB 中记录的各寄存器内容恢复到 CPU 各个寄存器内。（恢复现场）
   3. 闲逛进程
2. 进程控制：
   1. 创建进程
   2. 进程销毁
   3. 进程阻塞
   4. 进程唤醒

### 概要设计

CPU流程图



进程运行流程



### 详细设计

寄存器



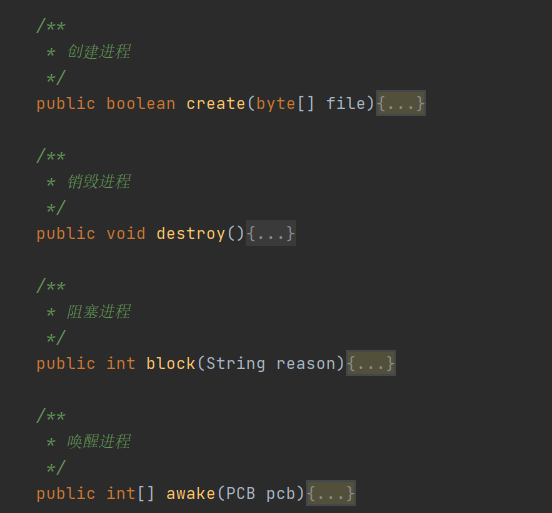
Cpu类





ProcessScheduling类

主要方法：



1. create()

输入值：file

输出值：boolean是否申请成功

操作变量：PCB进程及内存data

伪代码：申请内存空间

将进程的数据从磁盘读入内存

初始化pcb

将pcb保存入进程管理器的就绪队列

1. block()

输入值：设备号

输出值：无

操作变量：想要阻塞的PCB进程

伪代码：将进程从运行态转换成阻塞态

将进程pcb加入阻塞队列

保存现场

1. awake()

输入值：PCB

输出值：无

操作变量：想要唤醒的PCB进程

伪代码：将进程从阻塞态转换成就绪态

将进程的pcb加入就绪队列

1. destory()

输入值：无

输出值：无

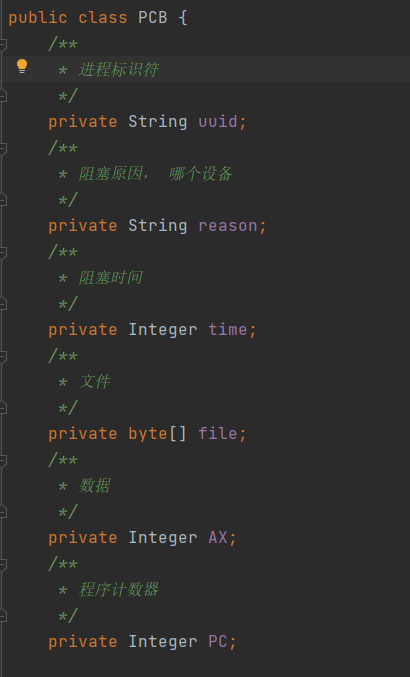
操作变量：想要销毁的PCB进程

伪代码：将进程从阻塞队列，就绪队列中删除

并且将pcb销毁

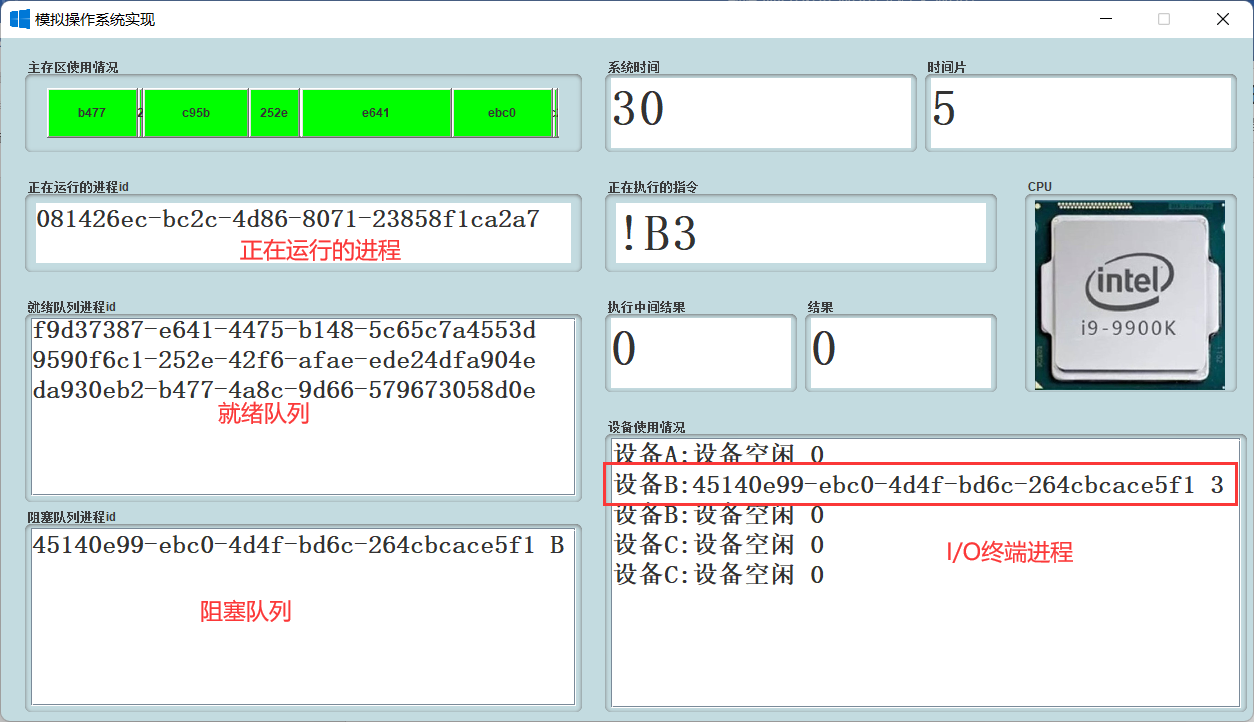
执行内存回收

PCB类



### 运行效果





### 个人感悟

在这一个学期的操作系统课程设计中我主要做cpu和进程管理的实现，收获了很多也学会了很多：

1. 代码能力，尝试了很多之前没有接触过的类和软件，极大提高了开发效率，也在其中学到了遇到问题解决问题的能力。
2. 对于操作系统有了更深层的了解和理解，在我遇到实现逻辑的问题时，回去翻课本和回顾老师之前所讲的内容，根据这些理论，很完美的解决我在实现逻辑上遇到的种种难题。
3. 其他课程与会应用其中，让我更深的理解了计算机及其架构。我在实现cpu时应用到了我在数据结构上学的知识还有计算机组成原理的知识，多门课相互融合，让我更加明白了计算机系统
4. 团队的重要性，当遇到难题解决不了时，团队提供了很多帮助，我们一起实现了分工完成，无缝连接，把课设分块，大家协力完成，比一个人完成效率高很多，也对以后团队合作提供了不少经验。

这是一次不错的课设体验。