小组讨论报告1

**讨论时间：**2023.9.6

**地点：**博远楼办公室

**参加人员：**

**讨论时长：**1小时

**讨论内容分析：**

昨天开了操作系统课程设计的第一次线上会议，夏烨学长展示了如何实现多线程并发。我们对java如何实现多线程编程进行了讨论。在java中实现多线程编程主要有两种方法，一是继承Thread类，二是通过Runnable接口。

在本次课设中，需要实现至少三个线程并发执行，其中核心是时钟线程类，它是仿真中最重要的部分。我们认为想要确保程序无误，需要着重探讨时钟线程中如何对COUNTTIME这一变量进行更改。大家在自己的电脑上进行了编码实验。

**讨论心得：**

这是我开始操作系统课程设计的第一步——多线程并发，以此为基础，对线程操作有了更清晰的认知，在后续的阻塞唤醒线程写作和其他内容的撰写上更加得心应手。

**图片证明材料：**

小组讨论报告2

**讨论时间：**2023.9.16

**地点：**博远楼办公室

**参加人员：**

**讨论时长：**1.5小时

**讨论内容分析：**

距离开始课设已经过了一段时间，今天我们讨论了课程设计中硬件仿真和多个关键数据结构的实现问题。在课设中，我们主要要实现的核心硬件仿真就是CPU和内存，中断的实现在上次小组讨论时已经涉及到。关于数据结构，最重要的就是进程控制块PCB 的实现，这离不开作业控制块JCB，因为我们认为需要先从文件读入作业信息，再根据一定的条件将作业转换为进程。

我们分析了PCB和JCB中需要有哪些相同的部分，其中ID序列号和Instrucnum指令数目这两者是可以直接赋值的，此外，还需要有一个列表来保存每个作业中具体指令情况。基于上述分析，小组成员对程序中的PCB和JCB进行了更改。

**讨论心得：**

在本次讨论之前，我还处于对PCB与JCB分不清的状态，经过本次讨论，我能够实现PCB和JCB的正确使用。

**图片证明材料：**

小组讨论报告3

**讨论时间：**2023.9.20

**地点：**寝室、线上

**参加人员：**

**讨论时长：**1小时

**讨论内容分析：**

今天我们集中讨论了关于CheckJob()函数、作业后备队列和就绪队列的内容。这部分内容是进行进程调度的前提。

CheckJob()函数要求检测读入的作业队列中是否有已到达的作业（即其InTimes<=当前COUNTTIME），如果满足要求，那么将作业加入后备队列。在实现的过程中，我们发现有时会出现只能读入一次作业的情况，经过检查发现是if条件出错。

**讨论心得：**

一开始我们对CheckJob函数中具体的执行条件比较困惑，尝试撰写后总是会出现各种问题，经过讨论能够成功实现。

**图片证明材料：**

