**第一次小组讨论（**9月6号**）：**

在这个课程设计中，我们小组着重研究了处理机管理和内存管理两个关键模块，并通过实践和仿真来验证和优化算法的性能。

在处理机管理方面，我们设计了一个基于轮转调度算法的处理机调度仿真模型。通过模拟进程的抢占和释放，我们成功实现了多进程的轮转调度，并且通过调整时间片大小和优先级等参数，对其进行了性能优化。我们还通过可视化界面展示了每个进程在不同时间片内的运行情况，以及调度算法对于系统整体性能的影响。

而在连续动态内存管理方面，我们采用了首次适应算法和最佳适应算法来实现内存分配与回收的仿真。通过模拟进程请求内存和释放内存的操作，我们实现了动态分区的内存管理，并且根据实验结果对比和分析了两种算法的性能差异。我们还结合内存碎片的情况，对算法进行了改进和优化，提高了内存的利用率和系统的整体性能。

通过这次课程设计，我们小组不仅加深了对处理机管理和内存管理的理论理解，还通过实践学会了如何应用仿真工具进行系统模拟，并进行性能优化。同时，我们也体验到了团队合作和沟通的重要性，通过分工合作，我们在规定时间内完成了任务，同时也培养了团队意识和协作能力。

此外，我们小组还认识到了操作系统设计与实现对于计算机系统的重要性。一个高效、稳定的操作系统能够提高计算机的性能，并且为用户提供良好的使用体验。因此，我们在未来的学习和工作中将更加注重操作系统相关技术的学习和应用，为我们的技术能力提升打下坚实的基础。



**第二次小组讨论（9月15号）**

在这个课程设计中，我们小组着重研究了磁盘调度算法和文件系统的设计与实现，并通过仿真来验证和优化算法性能。

首先，在磁盘调度算法方面，我们设计了一个基于扫描算法的磁盘调度仿真模型。通过模拟读写请求的到达顺序和磁盘磁头的移动情况，我们成功实现了磁盘的扫描调度，并通过调整起点位置和方向等参数，对其进行了性能优化。我们还通过可视化界面展示了不同请求的服务时间和等待时间，以及调度算法对磁盘性能的影响。

其次，在文件系统设计方面，我们实现了一个简单的文件系统模拟。我们设计了文件分配表、目录结构和文件读写接口等，通过模拟文件的创建、删除和读写操作，成功构建了一个完整的文件系统。我们还实现了基本的文件权限控制和目录管理功能，并通过对文件的组织和存储方式进行优化，提高了文件系统的性能。

通过这次课程设计，我们小组不仅加深了对磁盘调度算法和文件系统设计的理论理解，还通过实践学会了如何进行仿真和优化。同时，我们也体会到了团队合作和沟通的重要性，通过分工合作，我们在规定时间内完成了任务，并培养了团队合作精神。

在未来，我们将继续深入学习和探索操作系统相关技术，为我们的技术能力提升打下稳固基础。通过不断学习和实践，我们相信我们可以为计算机系统的稳定性和性能提供更好的支持和贡献。



**第三次小组讨论（9月27号）**

首先，在进程同步方面，我们设计了一个基于信号量机制的进程同步仿真模型。通过模拟进程的创建和运行，并设置合适的同步点和信号量操作，我们成功实现了多个进程之间的同步。通过调整信号量值和优先级等参数，对同步算法进行性能优化。我们还通过可视化界面展示了进程在不同同步点上的执行情况，以及同步算法对整体系统性能的影响。

其次，在互斥机制方面，我们设计了一个基于互斥锁的进程互斥仿真模型。通过设置互斥锁的申请和释放操作，我们成功实现了多个进程之间的互斥访问共享资源。通过调整互斥锁的优先级和超时机制等参数，对互斥算法进行性能优化。我们还通过可视化界面展示了不同进程对共享资源的访问情况，以及互斥算法对系统性能的影响。

通过这次课程设计，我们小组不仅加深了对并发进程同步与互斥机制的理论理解，还通过实践学会了如何应用仿真工具进行系统模拟，并进行性能优化。同时，我们也体验到了团队合作和沟通的重要性，通过分工合作，我们在规定时间内完成了任务，并培养了团队合作精神。

