

**实 验（实训）报 告**

**项 目 名 称**  第二次实验报告

**所属课程名称**  操作系统

**项 目 类 型**  验证/设计型

**实验(实训)日期**  2024/11/10

**班 级**  软件1班

**学 号**  220110900701

**姓 名**  蔡喆彦

**指导教师**  陈伟锋

浙江财经大学教务处制

|  |
| --- |
| **一、实验（实训）概述：** |
| **【目的及要求】**  安装gcc编译器。  完成hello.c、fork.c、processes.c、communication.c这四个程序的编写和编译。  通过编写C程序来模拟操作系统中的进程调度过程，特别是使用动态优先权算法来调度多个进程，实现对N个进程采用动态优先权算法的调度。  **【基本原理】**  动态优先权算法是一种进程调度算法，它根据进程的等待时间和服务时间来调整进程的优先级。这种算法通常用于操作系统的进程调度中，以确保系统资源得到有效利用，避免进程饥饿，提高系统吞吐量。  定义一个进程控制块（PCB）来存储每个进程的信息。  实现用调度器来管理PCB。  **【实施环境】**  Ubuntu、VMware、c语言 |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）**  **实验5**  **1、安装gcc**  使用以下命令安装gcc    使用该命令查看是否安装成功    **2、完成hello.c**  创建一个名为hello.c的文件，并在文本编辑器中编辑代码      通过以下命令可以运行该代码     1. **完成fork.c**   创建文件并编辑代码    运行代码    4、完成proceses.c  编写代码    运行代码可以看出父进程和子进程进入了无限循环    5、完成communication.c  编辑代码    运行显示结果    **实验6**  编写代码实现对N个进程采用动态优先权算法的调度  部分代码如下：    运行后显示：    输出结果：      **【结论与讨论】（结果、分析）**  使用C语言在虚拟机中实现简单的模拟。  通过调整进程的优先级，系统能够更有效地分配CPU时间片，减少进程等待时间，并提高整体的系统吞吐量。如果没有适当的调度策略，某些进程可能会长时间得不到执行，导致饥饿。  合理的进程调度策略可以显著提高系统性能，包括更快的响应时间和更高的资源利用率。  动态优先权算法考虑了进程的等待时间，有助于实现CPU资源的公平分配。  可以深入理解进程调度的基本概念和动态优先权算法的工作原理，以及进程间通信的重要性。 |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：**  **批阅日期：** |