

**实 验（实训）报 告**

**项 目 名 称**  进程控制

**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**  2024/10/30

**班 级**  22软件2班

**学 号**  220110900402

**姓 名**  柴思敏

**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】**  深入理解进程控制的基本概念和技术，涵盖进程的创建、终止、同步、通信等方面。通过实践编程练习，深刻地理解操作系统中的进程管理机制。具体要求包括：  1.学习如何在Linux环境下使用GCC编译器编译C程序。  2.掌握使用fork()函数创建子进程的方法。  3.理解进程间通信（IPC）的基本概念，尤其是管道（pipe）的使用。要求能够编写简单的C程序来实现进程的创建和进程间的通信。  4.用C语言实现对N个进程采用动态优先权算法的调度。  **【基本原理】**  **进程：**操作系统进行资源分配和调度的基本单位，每个进程都有自己的地址空间、内存、数据栈以及执行代码。  **进程控制：**涉及进程的生命周期管理，如创建、调度、同步、通信及终止。  **进程间通信（IPC）：**不同的进程之间需要一种机制来进行数据交换或协调工作，常见的IPC机制有管道、消息队列、共享内存等。  **动态优先权调度算法：**根据进程的动态特性调整其优先级的调度策略，以达到优化系统性能的目的。  **【实施环境】**  操作系统：Ubuntu操作系统  编译器：GCC  开发工具：终端、nano编辑器 |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）**  1.通过包管理器apt来安装GCC：    2.完成hello.c  创建一个名为my\_c\_programs的目录，再创建一个C源文件：    打开一个空白的编辑窗口，输入C代码：    使用gcc编译器编译文件，生成一个可执行文件，然后运行程序，显示结果：    3.完成proceses.c      4.完成communication.c    16c72a029f389f1d45dd8b6486d3611  5.用C语言实现对N个进程采用动态优先权算法的调度      **【结论与讨论】（结果、分析）**  结果：  成功安装并配置了GCC编译器，能够在Linux环境下编译和运行C语言程序。利用fork()函数创建了子进程，并通过观察父进程与子进程之间的行为差异，加深了对进程创建机制的理解。通过编写程序实现了简单的进程间通信功能，使用管道(pipe)作为进程间的数据传输通道，验证了不同进程间可以有效且安全地交换数据。对N个进程进行了动态优先权算法的调度实现，通过对进程优先级的动态调整，提高了系统的响应速度和资源利用率。  分析：  在实验过程中，遇到了一些常见的问题，如编译错误、程序逻辑错误等，通过仔细检查代码和查阅相关资料，这些问题得到了解决。在使用管道时，忘记关闭不必要的读写端会导致数据传输失败，通过关闭正确的端口解决了这一问题。在实现进程间通信时，遇到了数据同步的问题，即如何确保发送方已经完成数据写入后接收方才开始读取。通过查阅资料和多次尝试，最终使用了信号量来解决这一问题。  动态优先权算法的实现较为复杂，需要考虑多种因素，如进程的CPU占用率、等待时间等，合理设置优先级调整策略对于提高系统效率至关重要。  通过这次实验，不仅巩固了理论知识，还提高了实际动手能力，对于后续学习操作系统更高级的功能具有重要意义。 |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：**  **批阅日期：** |