山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200300092 | 姓名：郑睿杰 | | 班级：大数据2班 |
| 实验编号：实验3 | | | |
| 实验题目：进程调度算法实验 | | | |
| 实验学时：4 | | 实验日期：2024.5.1 | |
| 实验目的：  加深对进程调度概念的理解，体验进程调度机制的功能，了解 Linux 系统中进程  调度策略的使用方法。 练习进程调度算法的编程和调试技术。 | | | |
| 硬件环境：  机型：ThinkBook-13x-ITG  CPU：11th Gen Intel(R) Core(TM) i7-1160G7 @ 1.20GHz  内存：16150988 kB | | | |
| 软件环境：  Ubuntu 9.4.0-1 ubuntu1~20.04.2 + vscode2022 | | | |
| 实验步骤与内容：  【实验模型描述】  本实验使用C语言编写了一个简单的父子进程模型。在该模型中，父进程和子进程分别在无限循环中调用系统调用函数，用于设置各自的进程优先级，并通过信号处理函数实现动态调整。父进程和子进程的优先级由一个全局变量 flag 控制，通过 Ctrl+C 和 Ctrl+Z 信号的捕获，动态增减 flag 的值，从而调整进程的优先级。  【主要算法代码分析】    1.使用 signal 函数设置 SIGINT 和 SIGTSTP 信号的处理函数为 add 和 minus，用于2.捕获 Ctrl+C 和 Ctrl+Z 信号，并对全局变量 flag 进行增减操作。  3.使用 fork 函数创建子进程，父子进程各自根据 flag 的值调用 setpriority 函数设置自身的进程优先级。  4.使用 getpriority 函数获取当前进程的优先级，并通过 printf 函数输出相关信息。  父子进程均进入无限循环，周期性输出进程ID、优先级和调度器信息。  【运行结果】    【实验结果的分析综合】  通过实验结果可以观察到，父子进程在不同的优先级下交替运行，并根据 Ctrl+C 和 Ctrl+Z 信号的触发，动态调整各自的优先级。调度器根据进程优先级的变化，合理地调度进程的执行顺序，从而实现了进程的动态调度。 | | | |
| 结论分析与体会：  【实验中遇到的问题和解决方法】   1. 在程序中出现了链接错误，提示找不到 add 和 minus 函数的定义。将inline去掉后即可正常运行 2. 按下 Ctrl+C 后出现了终端错误的情况，研究后得知需要将终端切换到最高权限才能正常操作   【实验结果达到设计目标的程度】  实验结果基本达到了设计目标，成功实现了父子进程的优先级动态调整，并观察到调度器在不同优先级下的表现。  【还可以进行哪些改进】  可以进一步扩展实验，尝试使用其他进程通信的方式，如共享内存、消息队列等，以及其他调度算法，如多级反馈队列调度算法等，从而深入探究进程间通信和调度的机制。  在实验中加入更多的调度策略和优先级变化的情况，以便更全面地观察调度器在不同情况下的表现，从而加深对操作系统调度机制的理解。  【实验得到的收获和启发】  通过本次实验，我不仅学会了如何使用信号处理、进程优先级控制和进程调度相关函数，还深入理解了进程通信和调度的原理和实现方式。这为我进一步学习和应用操作系统提供了坚实的基础，也提升了我的编程能力和问题解决能力 | | | |