山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100300063 | 姓名：李彦浩 | | 班级：软件菁英班 |
| 实验编号：实验*2* | | | |
| 实验题目：进程调度算法实验 | | | |
| 实验学时：2 | | 实验日期：2023.05.1 | |
| 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：  ThinkPad 8核 Intel Core i5-8265U 1.60GHz 8G内存  山大云算力平台 devlinux 1核 1G内存 | | | |
| 软件环境：  Vmware workstation;Ubuntu 20.04 | | | |
| 实验步骤与内容：  独立实验要求开启两个并发执行的父子进程，不断循环输出各自进程号、优先数和调度策 略。 进程初始调度策略均为系统默认策略和默认优先级。当某个进程收到 SIGINT 信号时 会自动将其优先数加 1，收到 SIGCSTP 信号时会自动将其优先数减 1。  仿照示例程序给出的代码，建立两个并发的父子进程，赋予其系统默认的调度策略和优先数（根据实验指导书分别是0和10），随后设定两个绑定了键盘信号ctrl+c/z的函数，分别对当前优先数做自增和自减。        利用之前学习的fork()函数，建立并发进程。监测fork()函数的返回值pid,如果大于0，代表为父进程。父子进程都不断地汇报自己的pid,priority,scheduler。因此需要写死一个while(1)的循环在这两个进程中。循环中，每次根据进程号设置优先数（这个优先数或许已经经过了键盘信号更改），然后sleep3秒，这样做是为了防止printf的信息刷屏。    如果运行的是子进程，其实也和父进程的流程大同小异，函数里参数换了换而已。    在执行实验代码部分上，我采用了vmware workstation和Ubuntu20.04。将makefile文件和c文件放在了一个文件夹下。  在终端中gmake，对代码进行编译。    然后需要拿到root权限。因为提高进程优先数需要root用户的权限。    执行./psched，得到结果。  可以看到结果中，每过三秒会打印一次当前父子进程的各项信息（这个动态的过程在实验录屏中可以看到），随后如果按下ctrl+c会给优先数自增1，ctrl+z会给优先数自减1。        到这里整个独立实验要求的功能就都已经实现了。  本次实验实际投入学时2h,主要时间均用于代码的编写、调试与执行上  实验采用一种测试方式。执行./psched，按下ctrl+c/z，得到新的优先数。  实验中我创建了两个并发的进程，通过键盘信号修改其优先数，然后不断更新在终端中。 | | | |
| 结论分析与体会：  实验一开始时我将题目理解错误了，以为是要通过键盘信号控制两个进程的优先数分别自增1或自减1，即，父进程的自增自减由两个键盘信号控制，子进程的由另两个控制。但后来发现只有两个键盘信号，于是改成了当前代码中的形式。  实验结果在上一个部分中已经列出，与预测的结果完全一致，完全复现了设计目标。  代码功能上还有可以增加的地方。例如可以将控制信号改为4个，然后分别控制父子进程优先数的自增自减，不至于一个键盘信号按下去两个进程的优先数同时改变。  实验中我创建了两个并发的进程，通过键盘信号修改其优先数，然后不断更新在终端中。 | | | |