山东大学 计算机科学与技术 学院

操作系统 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201705130120 | 姓名：苑宗鹤 | | 班级： 1班 |
| 实验题目： 进程调度算法实验 | | | |
| 实验学时：2 | | 实验日期：2019/12/22 | |
| 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：  pc | | | |
| 软件环境：  Win10 clion  Linux ubuntu | | | |
| 实验内容与设计：  设有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 SIGTSTP 信号时会自动将其优先数减 1。请编 程实现以上功能。    实验代码：  #include **<stdio.h>** #include**<sched.h>** #include **<unistd.h>** #include **<stdlib.h>** #include**<sys/time.h>** #include**<sys/resource.h>** #include**<iostream>** #include **<signal.h>** #include **<sys/types.h>  using namespace** std; **int** pid;  **typedef void** (\*sighandler\_t)(**int**); **struct** sched\_param p; **int** prio=5;  **void** sigint() {  **if**(pid>0){  prio++;  setpriority(**PRIO\_PROCESS**, pid, prio);  } } **void** sigtstp() {  **if**(pid>0){  prio--;  setpriority(**PRIO\_PROCESS**, pid, prio);  } } **int** main(**int** argc, **char** \*argv[]) {   **int** i, j, status;  **if** ((pid = fork()) > 0) {   *//主进程`* p.**sched\_priority** = 10;  sched\_setscheduler(pid,  **SCHED\_OTHER**,  &p);  setpriority(**PRIO\_PROCESS**, pid, prio);  }  signal(**SIGINT**, (sighandler\_t) sigint);  signal(**SIGTSTP** , (sighandler\_t) sigtstp);   sleep(1);  **for** (j = 0; j < 20; ++j) {  cout << **" pid="** << getpid() << **" pri="** << getpriority(**PRIO\_PROCESS**, 0) << endl;  sleep(1);  }    **return EXIT\_SUCCESS**; }  运行结果： | | | |
| 结论分析与体会：  通过pid判断父子进程  只在子进程中响应**SIGINT和SIGTSTP** | | | |