《计算机操作系统》实验报告

|  |  |  |
| --- | --- | --- |
| 班级： | 学号： | 姓名： |
| 指导老师姓名：钟鑫 | 实验日期：2023.10.20 | 成绩： |

|  |
| --- |
| 实验单元五 进程调度 |

# 实验目的和要求

|  |
| --- |
| 1、进程调度是处理机管理的核心内容  2、通过本实验加深对进程控制块、就绪队列等概念的理解  3、体会和了解优先权调度算法的具体实现方法 |

# 二、实验指导

|  |
| --- |
| 基于优先权的进程调度，把所有的进程按照优先权顺序插入就绪队列，调度的时候，从就绪队列中选择优先权最高的进程，调度运行。当一个进程运行的时候，动态修改其优先级系数和 CPU 时间，如果 CPU 时间增加到大于或等于所需的全部时间，则表示该进程运行完成，退出；否则，继续插入就绪队列，等待再一次被调度运行。  **步骤：**  1) 初始化工作：随机产生 5 个进程，设置好进程标志符 pid、优先级 pri、当前已获得的CPU时间统计cputime、运行完成总共需要的时间alltime、状态state（全部定义为1，即运行状态）；  2) 创建一个有序链表，作为就绪队列，并把上述 5 个进程按优先级从高到低的次序插入就绪队列；  3) 如果就绪队列不为空，从就绪队列中取下第一个进程（优先级最高的进程）模拟运行；就绪队列为空，退出运行；  4) 修改该进程的优先级数值 pri，增加 CPU 运行时间统计，即增加 cputime；  5) 判断，如果cputime>=alltime，该进程运行完毕退出；否则，该进程继续插入就绪队列，等待下一次调度运行；  6) 继续执行步骤 3)  算法流程：    1、数据结构说明  就绪队列是一个按照优先权排列的队列，所以我们采用有序链表结构（SortedList），在插入链表时，自动比较优先权，把尚未完成退出的进程插入到合适的位置。链表的首元素，  一定是优先权最高的进程。  有序链表（SortedList）函数说明：  Node\* InitSortedList( ) //初始化有序链表，返回一个链表的头结点  void insertSortedList(LinkList L,PROC proc) //按优先级把进程插入有序链表  PROC\* getHeadElement(LinkList L) //摘取链表的头结点，即优先权最高的进程  两个自定义结构：  typedef struct PROC{ //进程的 PCB 结构  int pid; //进程标志  int pri; //优先级  int cputime; //以用的 CPU 时间  int alltime; //运行完成所需的所有时间  int state; //状态  }PROC;  typedef struct Node{ //有序链表结点定义  struct PROC data;  struct Node \*next;  }Node; |

# 三、实验环境

|  |
| --- |
| 1.硬件环境：当前所有电脑硬件环境均支持。  2.软件环境：vm、Linux |

# 四、实验内容与结果（截图需要清晰可见结果）

|  |
| --- |
| **1.** 用 C 语言编写和调试一个基于优先权的进程调度程序。  2. 源代码截图  **3. 运行结果截图** |

**注：上机实验课作业文档命名方式为班级\_学号\_姓名；**

**如：信科1班202312345某某（一定要按规定格式上交报告否则按0分处理）**