CPU 调度算法实验

1. 实验目的

加深对操作系统CPU调度以及调度算法的理解

2. 实验内容

- 单处理器环境下，针对最短作业优先算法(SJF)和优先级调度算法(Priority)，分别模拟实现抢占调度和非抢占调度的调度程序
- 设计使用三个队列，分别为就绪队列(readyQueue)、运行队列(runningQueue)、等待队列(waitingQueue)
- 进程状态三种，分别为就绪状态：0、运行状态：1、等待状态：2
- 输入：task.txt 文件和指定调度算法
  - task.txt 文件为需要调度的进程集
  - 非抢占式最短作业优先调度算法: sjf
  - 抢占式最短作业优先调度算法: psjf
  - 非抢占式优先级调度算法: pprio
  - 抢占式优先级调度算法: prio
- 输出：按照所指定的调度算法，显示调度结果
  - 比如，P2:2、P1:2、P3:1、P4:5

3. 部分源码

task.txt 文件格式

格式：进程ID， 到达时间， 优先级， 运行时间

```
1 0 10 3
2 2 1 2
...
```

进程结构

```
typedef struct node {
    int procID;             /* 进程ID */
    int releaseTime;        /* 到达时间 */
    int priority;           /* 优先级 */
    int cpuTime;            /* 运行时间 */
    int executedTime;       /* 已运行时间 */
    int state;              /* 进程状态 */
    struct node * next;     /* 指向下一个进程指针 */
} PCB
```