**作业三**

**操作系统主要管理计算机哪些事务：**

### **1、 控制和管理系统行为：**

操作系统执行基本的任务，比如：识别来自键盘的输入，将输出结果发送给监视器，管理存储设备上的文件和文件夹，控制箱磁盘驱动器和打印机这样的外部设备。操作系统还必须确保同事工作的不同程序和不同用户之间不会互相干扰。此外，还负责安全问题，确保未经授权的用户和程序不能访问这个系统。

### **调度和分配系统资源：**

操作系统负责决定一个程序需要哪些计算机资源（比如：CPU时间，内存空间，磁盘，输入和输出设备）以及调度和分配这些资源来运行这个程序。

### **调度操作：**

### 操作系统负责调度程序的各种行为以充分利用系统资源。现在很多程序都支持多程序设计、多线程以及多进程以提高系统的性能。

**涉及到的算法：**

# **1.先来先服务([FCFS](https://so.csdn.net/so/search?q=FCFS&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_42570192/article/details/_blank))**

**算法思想**：从公平的角度考虑，类似于排队购物、打饭等。  
**算法规则**：按照作业或者进程到达的先后顺序进行服务。  
**方法用于作业/进程调度**：用于作业调度时，考虑的是哪个**作业**先到达**后备队列**；用于进程调度时，考虑的是哪个**进程**先到达**就绪队列**。  
**进程调度的方式**：非抢占式算法。  
  
**优缺点**：优点是算法公平且实现简单；缺点是排队在长作业或长进程后面的短作业或短进程需要等待很长的时间，带权周转时间很大，对短作业或进程的用户体验不好。所以先来先服务算法对长作业有利，对短作业不利。

# **2**.**最短时间优先(**[SJF](https://so.csdn.net/so/search?q=SJF&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_42570192/article/details/_blank)**)**

最短时间优先(SJF, Shortest Job First)  
**算法思想**：追求最少的平均等待时间，最少的平均周转时间，最少的平均带权周转时间。  
**算法规则**：最短的作业或者进程优先得到服务，这里的最短指的是要求服务的时间最短。  
**方法用于作业/进程调度**：即可用于作业调度，也可以用于进程调度，用于进程调度时称为最短进程优先算法(SPF, Shortest Process First)。  
**进程调度的方式**：最短时间优先和最短进程优先是非抢占式算法，但也有抢占式的算法——最短剩余时间优先算法。  
最短剩余时间优先算法(SRTN, Shortest Remaining Time Next)是每当有进程加入就绪队列发生改变时就需要调度，如果新到达的进程剩余时间比当前运行的进程剩余时间更短，则由新进程抢占处理机，当前运行进程重新回到就绪队列，此外，当一个进程完成时也需要重新调度。

**优缺点**：优点是有较短的平均等待时间和平均周转时间；缺点是不太公平，对短作业有利，对长作业不利，可能产生饥饿现象，此外，作业或者进程的运行时间是由用户提供的，并不一定真实，所以不一定能做到真正的短作业优先。

# **3.优先级调度**

**算法思想**：随着计算机的发展，特别是实时操作系统系统的出现，越来越多的应用场景需要根据任务的紧急程度来决定处理顺序。  
**算法规则**：每个作业或进程有各自的优先级，调度时选择优先级最高的作业或进程。  
**方法用于作业/进程调度**：可用于作业调度，也可用于进程调度，还会用于I/O调度。  
**进程调度的方式**：抢占式和非抢占式都有，区别在于非抢占式只需要在进程主动放弃处理机时进行调度，而抢占式的还需要在就绪队列变化时检查是否会发生抢占。  
**合理地设置各类进程的优先级**：  
①系统进程优先级高于用户进程；  
②前台进程优先级高于后台进程；  
③操作系统更偏好 I/O 型进程(I/O 繁忙型进程)。  
**根据优先级是否可以动态地改变，可以将优先级分为静态优先级和动态优先级两种**。静态优先级在创建进程时就已经确定，之后一直不变；动态优先级创建进程时有一个初始值，之后会根据情况动态地调整优先级。  
**动态优先级的调整时机**：从追求公平、提升资源利用率等角度考虑。如果某进程在就绪队列中等待了很长时间，可以适当提升其优先级；如果某进程占用处理机运行了很长时间，可以适当降低其优先级；如果发现一个进程频繁地进行I/O操作，可以适当提升其优先级。  
**优缺点**：优点是用优先级区分紧急程度、重要程度，适用于实时操作系统，可灵活地调整对各种作业或进程的偏好程度；缺点是若源源不断地有高优先级进程到来，则可能导致饥饿。