



b.

	周转时间				等待时间			
	FCFS	SJF	消抢占	rr	FCFS	SJF	消抢占	rr
P ₁	2	3	3	3	0	1	1	1
P ₂	3	1	1	1	2	0	0	0
P ₃	11	20	20	20	3	12	12	12
P ₄	15	7	7	7	11	3	3	3
P ₅	20	12	12	12	15	7	7	7

d. 后三种平均等待时间最小

5.1 为什么区分CPU密集型程序和I/O密集型程序对调度程序是重要的

如果是I/O密集型任务，使用多线程，如果是CPU密集型任务，使用多进程。
前者指内存磁盘I/O使用率高，CPU使用率低；相反，后者指CPU使用率高，内存磁盘I/O使用率低。

二者侧重的重点不一致，通过区分二者来选择合适的调度算法可以更合理地使用计算机资源并加强计算效率

5.2

a. CPU利用率和响应时间

上下文切换会降低CPU利用率，但会减少响应时间

b. 平均周转时间和最大等待时间

通过最先执行最短作业优先调度（SJF）可以使平均周转时间最短。然而，这种调度策略可能会使需要长时间运行的任务永远得不到调度且会增加他们的等待时间。

c. I/O设备利用率和CPU利用率

CPU利用率的最大化可以通过长时间运行CPU限制的任务和同时不实行上下文切换。I/O设备利用率的最大化可以通过尽可能调度已经准备好的I/O限制的任务。因此，频繁的上下文切换会降低CPU利用率但是会提高I/O设备的利用率。

5.4

(1) 每个都有自己的运行队列：优点是不宜冲突，缺点是不便于管理

(2) 共享队列：优点是管理方便，缺点是同步麻烦，可能引起冲突

5.14

a. 当 $b > a > 0$ 时，是什么算法？

优先级调度，CPU执行时间越长，优先级越低

b. 当 $a < b < 0$ 时，是什么算法？

轮转算法，进入CPU之后优先级的消耗可以相当于时间片的消耗

5.21 在什么情况下，就进程截止限期而言，单调速率调度不如最早截止期限优先调度

首先阐述二者具体的区别：

单调速率算法采用抢占的，静态优先级的策略，调度周期性任务。

当较低优先级的进程正在运行并且较高优先级的进程可以运行时，较高优先级进程将会抢占低优先级。

在进入系统时，每个周期性任务会分配一个优先级，它与其周期成反比。周期越短，优先级越高，周期越长，优先级越低。

这种策略背后的理由是：更频繁地需要CPU的任务应分配更高的优先级，此外，单调速率调度假定：对于每次CPU执行，周期性进程的处理时间是相同的，也就是说，在每次进程获取CPU时，它的CPU执行长度是相同的。

最早截止时间优先(Earliest-Deadline-First, EDF)调度根据截止时间动态分配优先级。

截止时间越早，优先级越高；截止时间越晚，优先级越低。

根据PDF策略，当一个进程可运行时，它应向系统公布截止时间要求。优先级可能需要调整，以便反映新可运行进程的截止时间。

所以当进程的上下文切换和中断处理的代价小于抢占CPU的代价时，单调速率算法不如EDF算法