

操作系统第3次书面作业

- 数据科学与计算机学院
- 软工三班
- 米家龙
- 18342075

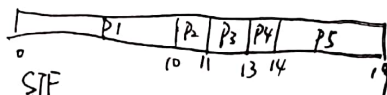
米家龙 18342075
软工三班

5.2

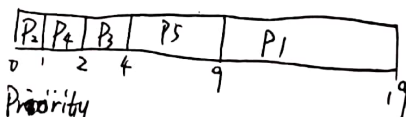
- ① CPU利用率和响应时间: 通过减小上下文切换来降低经常性的上下文切换, 但可能会导致进程的响应时间增加
- ② 平均周转时间和最大等待时间: 最先执行最短任务可以使平均周转时间最短, 但可能会使长时间运行任务永远无法被调度
- ③ I/O设备利用率和CPU利用率:
CPU利用率的最大化可以通过长时间运行CPU限制的任务且不执行上下文切换来实现。
I/O设备利用率最大化可通过尽可能调度已准备好的I/O限制任务, 但会导致上下文切换。
上述两种利用率难以同时最大化。

5.4

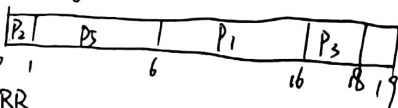
a. FCFS



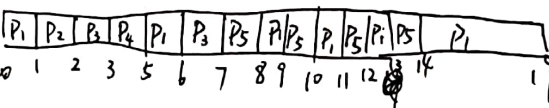
SJF



Priority



RR



b. 周转时间

进程	FCFS	RR	SJF	Priority
P1	10	19	19	16
P2	11	2	1	1
P3	13	7	4	18
P4	14	4	2	19
P5	19	14	9	6

c. 等待时间 (表同上)

进程	FCFS	RR	SJF	Priority
P1	0	9	9	6
P2	10	1	0	0
P3	11	5	2	16
P4	13	3	1	18
P5	14	9	4	1

d. SJF

5.5 so SJF和优先级标准, 这两种会引起饥饿

5.10

FCFS: 区别短任务, 因为在长任务到达后到来的短任务会有很长的等待时间

RR: 所有任务被分配的CPU时间区间都一样, 短任务如果能优先完成, 那么会很快离开

多级反馈队列: 与RR类似, 不会优先选择完成短任务。

9.2 对每个访问操作, 都需要查表及检查对页是否驻留, 并且检查程序是否有访问该页或写入的权限。上述检查在硬件中执行, TLB可用作缓存来提高查找效率。

9.3 ①将相应的页以写保护方式映射到这2个程序的虚拟地址空间

②发生写操作时, 必须进行复制, 从而避免程序访问不同的副本, 避免相互干扰

③硬件支持:

每次访问内存, 查询页表并检查是否有写保护, 如果有, 则发生陷阱, 交给OS操作。

9.4 二进制形式为

页表项: 0001 0001 0001 0010 0011 0100 0101 0110
页大小为 2^{12} , 则页表大小为 $2^{32-12} = 2^{20}$, 低位表示

9.5

$$\text{已知 } (1-P) \times 0.1 + 0.3P \times 8 \times 10^3 + 0.7P \times 20 \times 10^3 = 0.2$$

$$\text{解得 } P = \frac{1}{164000} \mu s$$

9.9

a. 发生页错误时, 如果该页在自由快池中不存在, 那么该页会被驱逐到磁盘, 作为常驻页移动到快池, 而已访问页会被移动到常驻集中。

b. 当页错误发生, 如果该页位于快池中, 则该页会被移入常驻页集, 而另一个常驻页会被移到快池。

c. 当常驻页上限为1时, 系统会退化使用自由快池中的LRU页面替换算法。

d. 当常驻页上限为0, 即无常驻页时, 系统会退化使用FIFO页面替换算法。

杨龙 18342075 软工三班

9.10 系统将在大量时间进行分页,说明内存分配过多。
如果降低程序的优先级,那么写进进程将减少页错误频率,从而提高CPU利用率。
另一种方法是更多的物理内存或更快地磁盘。

因此 a, b, c 不能达到期望效果。

而 d, e 可以

f 会提高读取和分页速度,可以改善情况。

g, 可让CPU更快获取数据,可行

h, 该操作可以减少页错误和提高利用率。

9.13.

a. ① 初始值为0

② 有新页与帧相关联时,计数器+1

③ 页面与其关联的帧失去联系时,计数器-1

④ 找到计数最小的帧进行替换可以使用FIFO算法。

b. 14

c. 11

9.14 $0.8 \times 1 + 0.1 \times 2 + 0.1 \times 5002$
 $= 501.2 \mu s = 0.5012 ms$

9.15 抖动是进程所需最小页帧数分配不足导致的,从而引起出现连续页错误。
② 系统可通过评估CPU利用率来检测抖动。
③ 可通过降低多道程序的级别来消除