

操作系统实验笔记

样题 1. 设周期性任务 P1, P2, P3 的周期 T1, T2, T3 分别为 100, 150, 350; 执行时间分别为 20,40,100. 试计算后回答是否可以用频率单调调度算法进行调度?

答: 频率单调调度算法是一种被广泛用于多周期性实时处理的调度算法。频率单调调度算法的基本原理是频率越低(周期越长)的任务的优先级越低。这时, 设任务周期为 T, 任务的执行时间为 C, 则使用频率单调调度算法的必要条件是 $C \leq T$ 。已经证明, 对于 $n(n \geq 1)$ 个周期的不同任务来说, 设每个周期为 T_i , 其相应任务的执行时间为 C_i , 则使用频率单调调度算法的充分

条件是: $\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_i}{T_i} + \dots + \frac{C_n}{T_n} \leq n \left(2^{\frac{1}{n}} - 1 \right)$

对于由 3 个周期组成的实时任务序列来说, 其执行时间与周期之比是:

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} \leq 3 \left(2^{\frac{1}{3}} - 1 \right) = 0.799$$

由于: $\frac{20}{100} + \frac{40}{150} + \frac{100}{350} = 0.752 < 0.799$

所以, 可以用频率单调调度算法进行调度.

样题 2. 假定某多道程序设计系统供用户使用的主存空间 100K, 磁带机 2 台, 打印机 1 台, 采用可变分区方式管理主存, 采用静态分配方式分配磁带机与打印机. 忽略用户作业 I/O 时间. 现有如下作业序列:

作业号	进入输入井时间	要求计算时间	主存需求量	磁带机需求	打印机需求
1	8:00	25 分钟	15K	1 台	1 台
2	8:20	10 分钟	30K	0 台	1 台
3	8:20	20 分钟	60K	1 台	0 台
4	8:30	20 分钟	20K	1 台	0 台
5	8:35	15 分钟	10K	1 台	1 台

作业调度策略先来先服务, 优先分配主存的低地址区域且不准移动已在主存的作业, 在主存中的各作业平分 CPU 时间, 问题如下:

(1) 作业调度选中各作业的次序是什么?

(2) 全部作业运行结束的时刻是什么?

(3) 如果把一个作业从进入输入井到运行结束的时间定义为周转时间, 在忽略系统开销时间条件下, 最大的作业周转时间是多少?

(4) 平均周转时间是多少?

答:

作业号	装入主存时间	开始执行时间	执行结束时间	周转时间
1	8:00	8:00	8:25	25 分钟
2	8:45	9:05	9:15	55 分钟
3	8:20	8:25	8:45	25 分钟
4	8:30	8:45	9:05	35 分钟
5	9:15	9:15	9:30	55 分钟

(1) 1, 3, 4, 2, 5

(2) 9:30

(3) 55 分钟

(4) $(25+55+25+35+55)/5=39$ (分钟)

样题 3. 设有 n 个缓冲区构成的循环缓冲区池, 每个缓冲区能容纳一个整数. 写进程 Writer 把整数逐个存入缓冲区池, 读进程 Reader 则逐个从缓冲区池中读出并打印输出, 要求打印的与输入的完全一样, 即个数, 次序, 数值一样. 试问:

(1) 写进程与读进程间具体的制约关系如何? (2) 用 PV 操作写出这两个进程的同步算法程序.

答：(1)写进程要领先于读进程，但当缓冲区全填满时，要等读进程取数；读进程要后于写进程，当缓冲去中无待读的整数时，要等待写进程写数。

(2)程序:

```
begin
  B: array[0...n-1] of integer
  W, R: integer;
SW, SR: Semaphore;
  W := 0; R := 0;
  SW := n; SR := 0;
cobegin
  PROCESS Writer
  begin
    L1: produce a integer into X; (或写生成一整数 X;)
      P( SW );
      B[W] := X;
      W := (W+1) mod n
      V( SR );
      goto L1;
  end;
  PROCESS Reader
  begin
    L2: P( SR )
      Y := B[R];
      R := (R+1) mod n
      V( SW );
      Print y;
      goto L2;
  end;
coend;
end;
```

样题 4. 某系统对主存采用页式管理，供用户使用的主存区域共 640K 字节，被分成 160 块，块号为 0,1,2.....159. 现有一作业的地址空间共占 4 页，其页号为 0,1,2,3，被分配到主存的第 2,4,1,5 块中，回答：

- (1) 作业每一页的长度为多少字节？ (2) 写出该作业被装入主存时，其对应的页表。
(3) 把该作业的每一页在主存中的起始地址(用 16 进制表示)填在下表中

页号	起始地址
0	002000
1	004000
2	001000
3	005000

答：(1) 每一页的长度为 $640K/160 = 4K$ 字节

(2) 页表

0	2	逻辑页号	主存块号
1	4	0	2
2	1	1	4
3	5	2	1
		3	5

样题 5. 作业的调度算法有几种？简述各自的优缺点。

答：1、先来先服务算法。

先来先服务算法是最简单的调度算法，它是按照作业进入“输入井”的先后次序来挑选作业，先进入的作业优先被挑选。

先来先服务算法具有一定的公平性，容易实现，但忽视了计算时间，可能使计算时间短的作业等待时间过长。

2、计算时间短的作业优先算法。

作业调度时依据在输入井中的作业提出的计算时间为标准，优先选择计算时间短且资源能得到满足的作业。

采用计算时间短的作业优先算法，能使平均周转时间最小，但它只考虑了用户估计的计算时间，可能使计算时间长的作业等待太久。

3、响应比最高者优先算法。

对“输入井”中的所有作业计算出它们的响应比，从资源能得到满足的作业中选择响应比高的作业优先装入主存储器。

响应比高者优先算法综合考虑作业的等待时间和计算时间。

4、优先数调度算法

为每个作业确定一个优先数，资源能满足且优先数高的作业先被选取，当几个作业有相同优先数时，对这些具有相同优先数的作业再按照先来先服务原则进行调度。

5、均衡调度法

根据作业对资源的要求进行分类，作业调度从各类作业中去挑选，尽可能地使得使用不同资源的作业同时执行。

这样不仅可以使系统的各种资源都在被使用，而且可以减少作业等待使用相同资源的时间，从而加快作业的执行。

样题 6. 若磁头的当前位置为 100 磁道，磁头正向磁道号增加的方向移动。现有一磁盘读写请求队列：22,374,202,127,13,54,182,389,19,4,7,28。若采用先来先服务、最短寻道时间优先和扫描算法，试计算出三种算法中，移动磁道总数和平均寻道长度各为多少？（列出寻道次序和移动道数）

答：1、先来先服务调度算法使移动臂移动的次序和移动的柱面数如下：

100 → 22 → 374 → 202 → 127 → 13 → 54 → 182 → 389 → 19 → 4 → 7 → 28

(78) (352) (172) (75) (114) (41) (128) (207) (370) (15) (3) (21)

移动磁道总数为：1576。

平均寻道长度为：131。

2、最短寻道时间优先调度算法使移动臂移动的次序和移动的柱面数如下：

100 → 127 → 182 → 202 → 54 → 28 → 22 → 19 → 13 → 7 → 4 → 374 → 389

(27) (55) (20) (148) (26) (6) (3) (6) (6) (3) (370) (15)

移动磁道总数为：685。

平均寻道长度为：57。

3、扫描调度算法使移动臂移动的次序和移动的柱面数如下：

100 → 127 → 182 → 202 → 374 → 389 → 54 → 28 → 22 → 19 → 13 → 7 → 4

(27) (55) (20) (172) (15) (335) (26) (6) (3) (6) (6) (3)

移动磁道总数为：674。

平均寻道长度为：56。

样题 7. 设系统有三种类型的资源，数量为(4,2,2)，系统中有进程 A,B,C 按如下顺序请求资源：

进程 A 申请(3,2,1)

进程 B 申请(1,0,1)

进程 A 申请(0,1,0)

进程 C 申请(2,0,0)

请给出一种防止死锁的资源剥夺分配策略，完成上述请求序列，并列出资源分配过程，指明哪些进程需要等待，哪些资源被剥夺。

答：可采用抢夺式的资源分配策略，使死锁的“不可抢夺资源”条件不成立。如果一个进程已经占有了某些资源而又要申请新资源，而新资源不能满足(已被其他进程占用)必须等待时，系统可以抢夺该进程已占有的资源。

以下是资源分配过程：

- (1) 首先是分配进程 A 所申请的资源 (3, 2, 1)；
- (2) 接着是进程 B 的申请(1,0,1)也能满足；
- (3) 进程 A 再次申请资源(0,1,0)时，由于不能满足资源需求，使进程 A 等待；
- (4) 进程 C 申请(2,0,0)时，资源已不能满足,这时可以剥夺进程 A 占用的资源(2,0,0)，进程 C 所需的资源得到满足。
- (5) 进程 A 申请的资源总数超出现有的资源总量，将会造成进程 A 的永远等待。

样题 8. 在请求页式系统中，一程序的页面走向(访问串或引用串)为 2,3,4,5,2,3,6,2,3,4,5,6,0. 设分配给该程序的存储块数为 M. 试分别计算 M=3 和 M=4 时 FIFO 和 LRU 两种方法的缺页数? 结果说明了什么?

答：1、FIFO 调度算法页面装入和调出的情况如下：

M=3 时:

2	3	4	5	2	3	6	2	3	4	5	6	0
2	2	<u>2</u>	5	5	<u>5</u>	6	6	6	6	<u>6</u>	<u>6</u>	0
	3	3	<u>3</u>	2	2	<u>2</u>	<u>2</u>	<u>2</u>	4	4	4	<u>4</u>
		4	4	<u>4</u>	3	3	3	3	<u>3</u>	5	5	5

缺页 缺页缺页缺页 缺页缺页缺页 缺页缺页 缺页

共产生 10 次缺页中断

M=4 时:

2	3	4	5	2	3	6	2	3	4	5	6	0
2	2	2	<u>2</u>	<u>2</u>	<u>2</u>	6	6	6	<u>6</u>	5	5	5
	3	3	3	3	3	<u>3</u>	2	2	2	<u>2</u>	6	6
		4	4	4	4	4	<u>4</u>	3	3	3	<u>3</u>	0
			5	5	5	5	5	<u>5</u>	4	4	4	<u>4</u>

缺页缺页缺页缺页 缺页缺页 缺页缺页缺页 缺页缺页

共产生 11 次缺页中断

2、LRU 调度算法页面装入和调出的情况如下：

M=3 时:

2	3	4	5	2	3	6	2	3	4	5	6	0
		4	5	2	3	6	2	3	4	5	6	0
	3	3	4	5	2	3	6	2	3	4	5	6
2	2	2	3	4	5	2	3	6	2	3	4	5

缺页 缺页缺页缺页 缺页缺页缺页 缺页缺页 缺页缺页

共产生 11 次缺页中断

M=4 时:

2	3	4	5	2	3	6	2	3	4	5	6	0
			5	2	3	6	2	3	4	5	6	0
			4	5	2	3	6	2	3	4	5	6
			3	4	5	2	3	6	2	3	4	5
			2	3	4	5	5	5	6	2	3	4

缺页 缺页缺页缺页 缺页 缺页缺页 缺页缺页

共产生 9 次缺页中断

结果说明: FIFO 不是堆栈算法, 增大 M 不一定降低缺页中断次数;

LRU 是堆栈算法, 增大 M 可以降低缺页中断次数.

样题 9. 当前磁盘读写位于柱面号 20, 此时有多个磁盘请求以下柱面号顺序送至磁盘驱动器:10,22,20,2,40,6,38. 寻道(track)时, 移动一个柱面需 6ms, 按下列三种算法计算所需寻道时间(柱面移动顺序及所需时间, 总寻道时间; 忽略最近指定柱面后所需寻道时间)

(1) 先到先服务 (2) 下一个最邻近柱面 (3) 电梯算法 (当前状态; 向上)

答: 1、先到先服务:

磁头移动顺序为: 20 → 10 → 22 → 20 → 2 → 40 → 6 → 38

(10) (12) (2) (18) (38) (34) (32)

所需时间: 60ms 72ms 12ms 108ms 228ms 204ms 192ms

磁头移动总量是: 146 柱面

总寻道时间是: 876ms.

2、下一个最邻近柱面:

磁头移动顺序为: 20 → 20 → 22 → 10 → 6 → 2 → 38 → 40

(0) (2) (12) (4) (4) (36) (2)

所需时间: 0ms 12ms 72ms 24ms 24ms 216ms 12ms

磁头移动总量是: 60 柱面,

总寻道时间是: 360ms.

3、电梯算法

磁头移动顺序为: 20 → 22 → 38 → 40 → 10 → 6 → 2

(2) (16) (2) (30) (4) (4)

所需时间: 12ms 96ms 12ms 180ms 24ms 24ms

磁头移动总量是: 58 柱面,

总寻道时间是: 348ms.

样题 10. 汽车司机与售票员之间必须协同工作, 一方面只有售票员把车门关好了司机才能开车, 因此, 售票员关好车门应通知司机开车. 另一方面, 只有当汽车已经停下, 售票员才能开门上下客, 故司机停车后应通知售票员, 汽车当前正在始发站停车上客, 试设必要的信号灯及赋初值, 写出他们的同步过程. (用管程或信号灯机制均可)

答:

SP: 表示是否可以开门上下客, 初值为"1", 表示正在始发站停车上客;

SD: 表示是否可以开车, 初值为"0", 表示还在始发站停车上客;

begin

SP, SD: Semaphore;

SP := 1; SD := 0;

cobegin

PROCESS Pay

begin

L1: P(SP);

{ 开门上客 };

{ 关车门 };

V(SD);

goto L1;

end;

PROCESS Drive

begin

```

L2: P( SD )
    { 开车 };
    { 停车 };
    V( SP );
    goto L2;
end;
coend;
end;

```

样题 11. 有三个进程 P1,P2 和 P3 并发工作. 进程 P1 需用资源 S3 和 S1; 进程 P2 需用资源 S1 和 S2; 进程 P3 需用资源 S2 和 S3. 回答:

(1) 若对资源分配不加限制, 会发生什么情况? 为什么?

(2) 为保证进程正确工作, 应采用怎样的资源分配策略? 为什么?

答: (1)可能会产生死锁. 因为当进程 P1、P2 和 P3 各得到了第 1 个资源 S3、S1 和 S2 后, 它们分别等待资源 S1、S2 和 S3, 于是形成了这样一个环路, 即: $P1 \rightarrow S1 \rightarrow P2 \rightarrow S2 \rightarrow P3 \rightarrow S3 \rightarrow P1$. 这条环路使 P1、P2 和 P3 循环等待资源的状态永远结束不了, 陷入了死锁。

(2)为保证进程能执行到结束, 可采用按序分配策略, 即进程 P1 应先申请资源 S1 再申请资源 S3, P2 应先申请资源 S1 再申请资源 S2, P3 应先申请资源 S2 再申请资源 S3,这样就可以保证每一个进程都先后得到所需的两个资源而执行到结束。