

## 第一章

1. 有一台计算机, 具有 1MB 内存, 操作系统占用 200KB, 各个用户进程分别占用 200KB。假设用户进程等待及执行 I/O 操作的时间平均占比 80%, 若增加 1MB 内存空间, 则 CPU 的利用率能够提高多少?

答: 设每个进程等待及执行 I/O 的时间占比为  $P$ , 则  $n$  个进程同时处于等待或执行 I/O 状态的概率是  $P^n$ , 此时 CPU 是空闲的, 故 CPU 的利用率为  $1-P^n$ 。由题意可知, 除去操作系统, 内存还能容纳 4 个用户进程, 由于每个用户进程等待及执行 I/O 操作时间为 80%, 故:

$$\text{CPU 利用率} = 1 - (80\%)^4 = 0.59$$

若再增加 1MB 内存, 系统中可同时运行 9 个用户进程, 此时:

$$\text{CPU 利用率} = 1 - (80\%)^9 = 0.87$$

故增加 1MB 内存使 CPU 的利用率提高了 47%, 即:

$$87\% - 59\% = 147\%$$

$$147\% - 100\% = 47\%$$

3. 某计算机拥有单 CPU 和两台 I/O (I1、I2) 设备, 支持多道程序设计, 同时投入 3 个程序运行。其执行轨迹如下:

P1: I2 (30ms), CPU (10ms), I1 (30ms), CPU (10ms), I2 (20ms)

P2: I1 (20ms), CPU (20ms), I2 (40ms)

P3: CPU (30ms), I1 (20ms), CPU (10ms), I1 (10ms)

如果 CPU、I1 和 I2 并行工作, 优先级从高到低依次为 P1、P2 和 P3, CPU 可根据优先级抢占, 但设备 I1 和 I2 不可抢占。试求: ①每个运行程序从投入到完成分别所需要的时间; ②CPU 的利用率; ③I/O 设备利用率。

答: ①P1 从投入到运行完成需 110ms, P2 从投入到运行完成需 90ms, P3 从投入到运行完成需 110ms。

②CPU 空闲时间段为: 60ms 至 70ms, 80ms 至 90ms, 100ms 至 110ms。所以 CPU 利用率为  $(110-30)/110=72.7\%$ 。

③设备 I1 空闲时间段为: 20ms 至 40ms, 90ms 至 100ms, 故 I1 的利用率为  $(110-30)/110=72.7\%$ 。设备 I2 空闲时间段为: 30ms 至 50ms, 故 I2 的利用率为  $(110-20)/110=81.8\%$ 。

4. 某计算机系统拥有单 CPU 和两台 I/O (I1、I2) 设备, 支持多道程序设计, 同时投入 3 个程序运行。其执行轨迹如下:

P1 : I2 (30ms), CPU (10ms), I1 (30ms), CPU (10ms)

P2 : I1 (20ms), CPU (20ms), I2 (40ms)

P3 : CPU (30ms), I1 (20ms)

如果 CPU、I1 和 I2 并行工作, 优先级从高到低依次为 P1、P2 和 P3, CPU 可根据优先级抢占, 但设备 I1 和 I2 不可抢占。试求: ①每个运行程序从投入到完成分别所需要的时间; ②CPU 的利用率; ③I/O 设备利用率。

答: ①P1 从投入到运行完成需 80ms, P2 从投入到运行完成需 90ms, P3 从投入到运行完成需 90ms。

②CPU 空闲时间段为: 60ms 至 70ms, 80ms 至 90ms。所以 CPU 利用率为  $(90-20)/90=77.78\%$ 。

③设备 I1 空闲时间段为: 20ms 至 40ms, 故 I1 的利用率为  $(90-20)/90=77.78\%$ 。设备 I2 空闲时间段为: 30ms 至 50ms, 故 I2 的利用率为  $(90-20)/90=77.78\%$ 。

5. 若内存中有 3 道程序 A、B、C，按照 A、B、C 的优先次序运行。各程序的计算轨迹为：

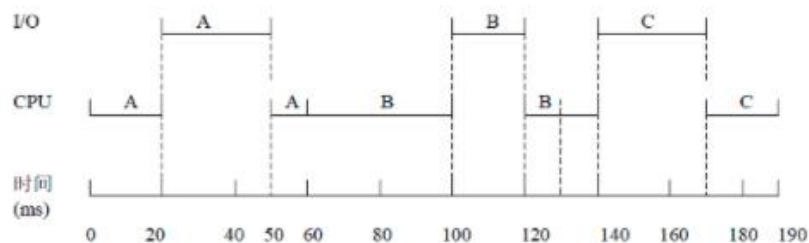
A：计算（20ms），I/O（30ms），计算（10ms）

B：计算（40ms），I/O（20ms），计算（10ms）

C：计算（10ms），I/O（30ms），计算（20ms）

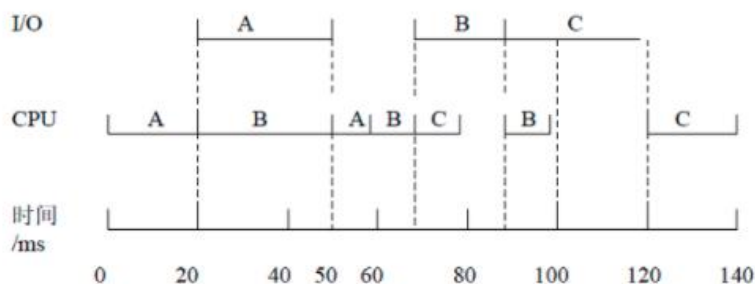
如果 3 道程序都使用相同的设备进行 I/O 操作（即程序以串行方式使用设备，调度开销忽略不计），试分别画出单道和多道运行的时间关系图。在两种情况下，CPU 的平均利用率各是多少？

答：单道运行时间关系图：



单道总运行时间为 190ms。CPU 利用率为  $(190-80)/190=57.9\%$ 。

多道运行时间关系图：



多道总运行时间为 140ms。CPU 利用率为  $(140-30)/140 \approx 78.6\%$ 。

8. 有两个程序，程序 A 按顺序使用：（CPU）10s、（设备甲）5s、（CPU）5s、（设备乙）10s、（CPU）10s。程序 B 按顺序使用：（设备甲）10s、（CPU）10s、（设备乙）5s、（CPU）5s、（设备乙）10s。若先执行程序 A，再执行程序 B，求总的 CPU 利用率是多少？

答：程序 A 执行了 40 秒，其中 CPU 用了 25 秒。程序 B 执行了 40 秒，其中 CPU 用了 15 秒。两个程序共用了 80 秒，CPU 花了 40 秒。故 CPU 利用率为  $40/80=50\%$ 。

9. 试写出命令解释程序的程序处理逻辑。

答：操作系统首先启动命令解释程序，输出命令提示符，等待键盘中断输入命令，或者是批处理作业当中的行结束符，或者是鼠标双击一个图标，鼠标选择一个菜单项，或者是多通道的模式识别等。

一旦这个事件发生后，操作系统就从命令缓冲区读入用户要求系统执行的那一条命令，然后进行响应，解释分析命令，执行处理代码。

命令解释处理完毕后，接收下一条命令。在操作控制计算机时，有前台命令和后台命令的区别。所谓前台就是命令之间必须严格按序执行，操作系统必须等待当前命令执行完毕后才能接收下一条命令；所谓后台就是启动命令之后，不管这个命令执行是否结束，操作系统马上就可以接收下一条命令。

10. 试述系统调用的实现原理，并写出系统调用的处理逻辑。

答：实现原理：编写系统调用服务例程；设计系统调用入口地址表，每个入口地址都指向一个系统调用的服务例程，有些还包含系统调用自带参数的个数；需要开辟现场保护区，以保存发生系统调用时应用程序的处理器现场。

处理逻辑：应用程序执行系统调用，产生中断转向内核态，进入陷阱处理程序，它将按功能号来查询入口地址表，并转至对应服务例程执行，完成后退出中断，返回应用程序断点继续运行。

## 第二章

1. 下列指令中，哪些只能在内核态运行？①读时钟日期；②陷入指令；③设置时钟日期；④加载 PSW；⑤置特殊寄存器；⑥启动 I/O 指令。

答：③④⑤⑥

15. 假定执行作业  $Job_1 \sim Job_5$ ，作业号的数字下标为其到达顺序，即依次按照序号 1、2、3、4、5 进入单处理器系统，各作业的执行时间和优先数如下表所示。①分别给出先来先服务调度算法、时间片轮转算法（时间片长度为 1ms）、短作业优先算法及非抢占式的优先数调度算法（优先数越小则优先级越高）下各作业的执行次序；②计算每种情况下作业的平均周转时间。

作业号	执行时间	优先权
$Job_1$	10	3
$Job_2$	1	1
$Job_3$	2	3
$Job_4$	1	4
$Job_5$	5	2

答：

采用 FCFS 算法调度作业，运作情况：

执行次序	执行时间	等待时间	开始时刻	完成时间	周转时间
Job1	10	0	0	10	10
Job2	1	10	10	11	11
Job3	2	11	11	13	13
Job4	1	13	13	14	14
Job5	5	14	14	19	19
作业平均周转时间 $T = (10+11+13+14+19) / 5 = 13.4$					

采用 RR 算法调度作业，若令时间片长=1，各作业执行情况为：1、2、3、4、5、1、

3、5、1、5、1、5、1、5、1、1、1、1。

作业	执行时间	提交时间	完成时间	周转时间
Job1	10	0	19	19
Job2	1	0	2	2
Job3	2	0	7	7
Job4	1	0	4	4
Job5	5	0	14	14
作业平均周转时间 $T = (19+2+7+4+14) / 5 = 9.2$				

采用 SJF 算法调度作业，运作情况：

执行次序	执行时间	等待时间	开始时刻	完成时间	周转时间
Job2	1	0	0	1	1
Job4	1	1	1	2	2
Job3	2	2	2	4	4
Job5	5	4	4	9	9
Job1	10	9	9	19	19
作业平均周转时间 $T = (1+2+4+9+19) / 5 = 7$					

采用非剥夺优先权算法调度作业，运作情况：

执行次序	优先数	执行时间	等待时间	周转时间
Job2	1	1	0	1
Job5	2	5	1	6
Job1	3	10	6	16
Job3	3	2	16	18
Job4	4	1	18	19
作业平均周转时间 $T = (1+6+16+18+19) / 5 = 12$				

17. 现有 5 个批处理作业 A~E 均已到达一台按单道方式执行的处理器，其运行时间分别为 2min、4min、6min、8min 和 10min，各自的优先级分别规定为 1、2、3、4 和 5，其中 5 是最高级。对于时间片轮转调度算法（时间片长度为 2min）、优先数调度算法、短作业优先调度算法、先来先服务调度算法（按作业到达次序 C、D、B、E、A），在忽略进程切换时间的前提下，计算出平均作业周转时间。

答：(1)FCFS 调度算法

执行次序	执行时间	等待时间	周转时间
C	6	0	6
D	8	6	14
B	4	14	18
E	10	18	28
A	2	28	30
作业平均周转时间 $T = (6+14+18+28+30) / 5 = 19.2$			

(2)优先数调度算法

执行次序	执行时间	等待时间	周转时间
E	10	0	10
D	8	10	18
C	6	18	24
B	4	24	28
A	2	28	30
作业平均周转时间 $T = (10+18+24+28+30) / 5 = 22$			

(3)时间片轮转法

按次序 A B C D E B C D E C D E D E E 轮转执行。

作业	执行时间	等待时间	周转时间
A	2	0	2
B	4	8	12
C	6	14	20
D	8	18	26
E	10	10	30
作业平均周转时间 $T = (2+12+20+26+30) / 5 = 18$			

(4)SJF 调度算法

作业	执行时间	等待时间	周转时间
A	2	0	2
B	4	2	6
C	6	6	12
D	8	12	20
E	10	20	30
作业平均周转时间 $T = (2+6+12+20+30) / 5 = 14$			

18. 在一个只支持三道程序同时运行的多道程序系统中，作业调度采用短作业优先调度算法，进程调度采用以优先数为基础的抢占式调度算法。在下表所示的作业序列中，优先数即为进程优先数，优先数越小则优先级越高。

作业名	到达时刻	估计运行时间/min	优先数
A	10:00	40	5
B	10:20	30	3
C	10:30	60	4
D	10:50	20	6
E	11:00	20	4
F	11:10	10	4

试填充下表：

作业名	进入内存时刻	运行结束时间	作业周转时间/min
A			
B			
C			
D			
E			
F			

平均作业周转时间  $T =$

答：

作业名	进入内存时刻	运行结束时间	作业周转时间/min
A	10:00	12:40	160
B	10:20	10:50	30
C	10:30	11:50	80
D	10:50	13:00	130
E	12:00	12:20	80
F	11:50	12:00	50
平均作业周转时间 $T=88.3$			

19. 在一个只支持四道程序同时运行的多道程序系统中，若在一段时间内先后到达 6 个作业，其提交时刻和估计运行时间由下表给出。

作业	提交时刻	估计运行时间/min
1	8:00	60
2	8:20	35
3	8:25	20
4	8:30	25
5	8:35	5
6	8:40	10

系统采用 SRTF 调度算法，作业被调度进入系统后中途不会退出，但作业运行时可被剩余时间更短的作业所抢占。①分别给出 6 个作业的开始执行时间、作业完成时间、作业周转时间。②计算平均作业周转时间。

答：

作业号	提交时间	需运行时间	开始运行时间	被抢占还需运行时间	完成时间	周转时间
1	8:00	60	8:00	40	10:35	155
2	8:20	35	8:20	30	9:55	95
3	8:25	20	8:25		8:45	20
4	8:30	25	9:00	25	9:25	55
5	8:35	5	8:45		8:50	15
6	8:40	10	8:50		9:00	20

说明：

J2 到达时抢占 J1；J3 到达时抢占 J2。

但 J4 到达时，因不满足 SRTF，故 J4 不能被运行，J3 继续执行 5 分钟。

由于是 4 道的作业系统，故后面作业不能进入内存而在后备队列等待，直到有作业结束。

根据进程调度可抢占原则，J3 第一个做完。而这时 J5、J6 均已进入后备队列，而 J5 可进入内存。

因 J5 最短，故它第二个完成。这时 J6 方可进入内存。因 J6 最短，故它第三个完成。

然后是：J4、J2 和 J1。

平均作业周转时间  $T=(155+95+20+55+15+20)/6=60$ 。

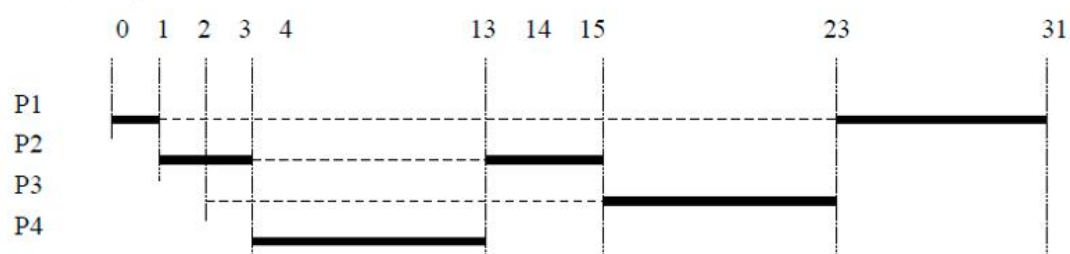
21. 设有 4 个进程  $P_1$ 、 $P_2$ 、 $P_3$ 、 $P_4$ ，它们到达就绪队列的时刻、运行时间及优先级如下表：

进程	到达就绪队列的时刻	运行时间/ms	优先级
$P_1$	0	9	1
$P_2$	1	4	2
$P_3$	2	8	3
$P_4$	3	10	4

①若采用抢占式优先数调度算法，试给出各个进程的调度次序以及进程的平均周转时间和平均等待时间。

②若采用时间片轮换调度算法，且时间片长度取 2ms，试给出各个进程的调度次序以及进程的平均周转时间和平均等待时间。

答：①

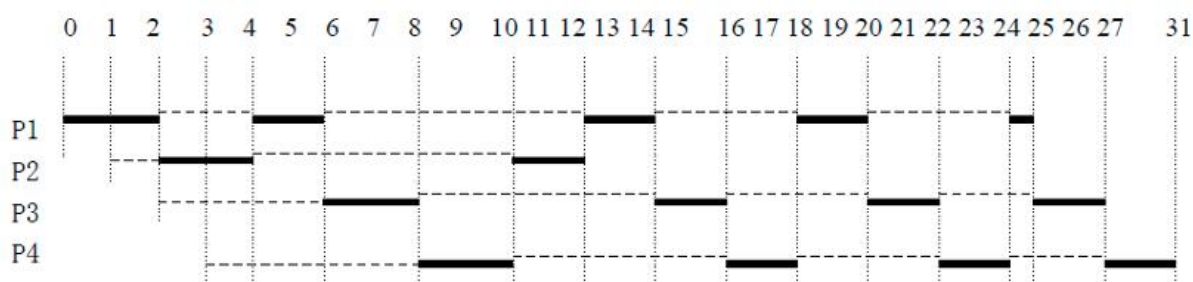


故调度次序为： $P_1$ 、 $P_2$ 、 $P_4$ 、 $P_2$ 、 $P_3$ 、 $P_1$ ；

平均周转时间= $(31+14+21+10)/4=19$ ；

平均等待时间= $(22+10+13+0)/4=11.25$ 。

②采用时间片轮换调度算法，调度次序如下：



平均周转时间= $(25+11+25+28)/4=22.25$ ；

平均等待时间= $(16+7+17+18)/4=14.5$ 。



### 第三章

4. 在可变分区存储管理下, 按地址排列的内存空闲区大小依次为: 10KB、4KB、20KB、18KB、7KB、9KB、12KB 和 15KB。依次考虑以下两个连续存储区的请求序列: ①12KB、10KB、9KB; ②12KB、10KB、15KB、18KB。请回答: 分别使用最先适配算法、最优适配算法、最坏适配算法和下次适配算法, 哪些空闲区将被使用?

答: (1) 对于序列 12KB、10KB、9KB;

1) 最先适配算法: 12KB 选中分区 3, 这时分区 3 还剩 8KB。10KB 选中分区 1, 恰好分配, 故应删去分区 1。9KB 选中分区 4, 这时分区 4 还剩 9KB。

2) 最优适配算法: 12KB 选中分区 7, 恰好分配, 故应删去分区 7。10KB 选中分区 1, 恰好分配, 故应删去分区 1。9KB 选中分区 6, 恰好分配, 故应删去分区 6。

3) 最坏适配算法: 12KB 选中分区 3, 这时分区 3 还剩 8KB。10KB 选中分区 4, 这时分区 4 还剩 8KB。9KB 选中分区 8, 这时分区 8 还剩 6KB。

4) 下次适配算法: 12KB 选中分区 3, 这时分区 3 还剩 8KB。10KB 选中分区 4, 这时分区 4 还剩 8KB。9KB 选中分区 6, 恰好分配, 故应删去分区 6。

(2) 对于序列: 12KB、10KB、15KB、18KB;

1) 最先适配算法: 12KB 选中分区 3, 这时分区 3 还剩 8KB。10KB 选中分区 1, 恰好分配, 故应删去分区 1。15KB 选中分区 4, 这时分区 4 还剩 3KB。最后无法满足 18KB 的申请, 应该等待。

2) 最优适配算法: 12KB 选中分区 7, 恰好分配, 故应删去分区 7。10KB 选中分区 1, 恰好分配, 故应删去分区 1。15KB 选中分区 8, 恰好分配, 故应删去分区 8。18KB 选中分区 4, 恰好分配, 故应删去分区 4。

3) 最坏适配算法: 12KB 选中分区 3, 这时分区 3 还剩 8KB。10KB 选中分区 4, 这时分区 4 还剩 8KB。15KB 选中分区 8, 恰好分配, 故应删去分区 8。最后无法满足 18KB 的申请, 应该等待。

4) 下次适配算法: 12KB 选中分区 3, 这时分区 3 还剩 8KB。10KB 选中分区 4, 这时分区 4 还剩 8KB。15KB 选中分区 8, 恰好分配, 故应删去分区 8。最后无法满足 18KB 的申请, 应该等待。

13. 在一页式存储管理系统中, 逻辑地址长度为 16 位, 页面大小为 4096B, 已知第 0、1、2 页依次存放在第 10、12、14 号物理块中, 现有逻辑地址 2F6AH, 请问其相应的物理地址是多少? (地址以十六进制表示)

答: 因为逻辑地址长度为 16 位, 而页面大小为 4096 字节, 所以, 前面的 4 位表示页号。把 2F6AH 转换成二进制为: 0010 1111 0110 1010, 可知页号为 2, 故放在 14 号物理块中, 写成十六进制为: EF6AH。

15. 在一个页式虚拟存储管理系统中, 用户编程空间为 32 个页, 页长为 1KB, 内存空间为 16KB。如果应用程序有 10 页长, 若已知页号为 0、1、2、3 的页已依次分得页框号为 4、7、8、10 的页框, 试把逻辑地址 0AC5H 和 1AC5H 转换成对应的物理地址 (以十六进制表示)。



答：页面大小为 1KB，所以低 10 位为页内偏移地址；用户编程空间为 32 个页面，即逻辑地址高 5 位为虚页号；主存为 16 个页面，即物理地址高 4 位为物理块号。

0AC5H 转换为二进制为 000101011000101B，虚页号为 2(00010B)，映射至物理块号 4，故系统访问物理地址 12C5H(01001011000101B)。

而执行逻辑地址 1AC5H 的转换时会发现虚页号为 6，不在页面映射表中，继而发生缺页中断，由系统另行分配页框。

21. 在一个页式虚拟存储管理系统中，进程访问地址的序列为：10-11-104-170-73-305-180-240-244-445-467-366。请回答：（1）如果页面大小为 100B，给出页面访问序列；（2）若进程分得 3 个页框，采用 FIFO 和 LRU 算法，计算缺页中断率。

答：（1）页面访问序列为 1, 1, 2, 2, 1, 4, 2, 3, 3, 5, 5, 4。

（2）FIFO 为 5 次，缺页中断率为  $5/12=41.6\%$ 。LRU 为 6 次，缺页中断率为  $6/12=50\%$ 。LRU 反比 FIFO 缺页中断率高。

FIFO: 缺页中断率= $5/12=41.6\%$

0	0	1	1	0	3	1	2	2	4	4	3
0	0	0	0	0	0	0	2	2	2	2	2
		1	1	1	1	1	1	1	4	4	4
					3	3	3	3	3	3	3
*		*			*		*		*		

LRU:缺页中断率= $6/12=50\%$

0	0	1	1	0	3	1	2	2	4	4	3
0	0	0	0	0	0	0	2	2	2	2	2
		1	1	1	1	1	1	1	1	1	3
					3	3	3	3	4	4	4
*		*			*		*		*		*

25. 给定段表如下：

段号	段首址	段长
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

试计算以下二维逻辑地址（<段号，段内位移>）对应的物理地址：（1）<0, 430>；（2）<3, 400>；（3）<1, 1>；（4）<2, 500>；（5）<4, 42>。

答：

- （1）因为  $0 < 5$ ， $430 < 600$ ，所以物理地址= $219+430=649$
- （2）因为  $3 < 5$ ， $400 < 580$ ，所以物理地址= $1327+400=1727$
- （3）因为  $1 < 5$ ， $1 < 14$ ，所以物理地址= $2300+1=2301$
- （4）因为  $2 < 5$ ， $500 > 100$ ，所以地址越界
- （5）因为  $4 < 5$ ， $42 < 96$ ，所以物理地址= $1952+42=1994$

## 第四章

12. 假定磁盘有 200 个柱面，编号 0~199，当前移动臂位于 143 号柱面上，并刚刚完成 125 号柱面的服务请求。如果请求队列的先后顺序是：86，147，91，177，94，150，102，175，130；试问：为了完成上述请求，下列算法移动臂所移动的总量分别是多少？并给出移动臂移动的顺序。①先来先服务算法；②最短查找时间优先算法；③双向扫描算法；④电梯调度算法。

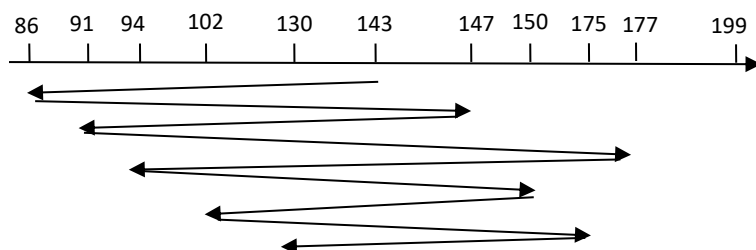
答：①先来先服务算法 FIFO 为 565，依次为 143-86-147-91-177-94-150-102-175-130。

②最短查找时间优先算法 SSTF 为 162，依次为 143-147-150-130-102-94-91-86-175-177。

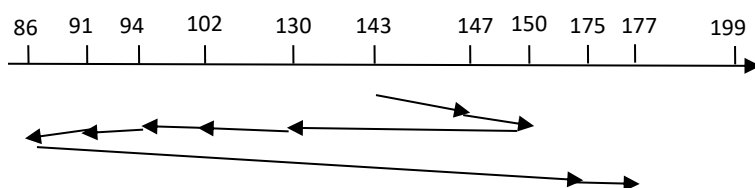
③扫描算法 SCAN 为 169，依次为 143-147-150-175-177-199-130-102-94-91-86。

④电梯调度为 125(先向地址大的方向)，依次为 143-147-150-175-177-130-102-94-91-86。

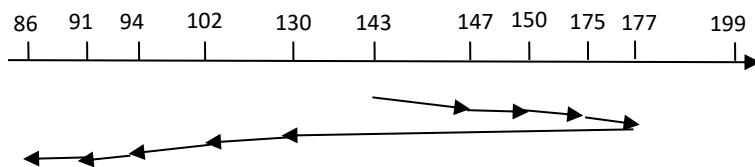
(1) 先来先服务



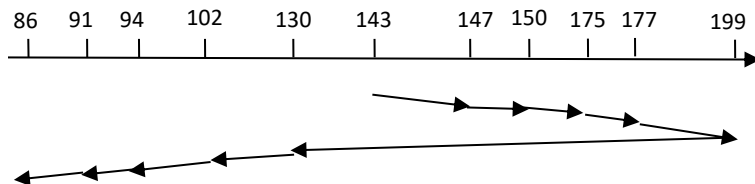
(2) 最短查找时间优先



(3) 电梯算法



(4) 双向扫描



14. 若磁头当前位于第 100 号柱面, 且正向磁道号减小的方向移动。现有磁盘读写请求队列, 柱面号依次为: 190, 10, 160, 80, 90, 125, 30, 20, 29, 140, 25。若分别采用最短查找时间优先算法和电梯调度算法, 试计算出各种算法中移臂所经过的柱面数?

答: 采用 SSTF 处理次序为: 100-90-80-125-140-160-190-30-29-25-20-10, 总柱面数为: 310。采用电梯调度处理次序为: 100-90-80-30-29-25-20-10-125-140-160-190, 总柱面数为: 270。

15. 若磁头当前位于第 100 号柱面, 且正向柱面号增加的方向移动。现有磁盘读写请求队列, 柱面号依次为: 23, 376, 205, 132, 19, 61, 190, 398, 29, 4, 18, 40。若采用先来先服务算法、最短查找时间优先算法和双向扫描算法, 试计算出各种算法中移动臂所经过的柱面数?

答: 采用先来先服务处理次序为: 100-23-376-205-132-19-61-190-398-29-4-18-40, 总柱面数为: 1596。

采用 SSTF 处理次序为: 100-132-190-205-61-40-29-23-19-18-4-376-398, 总柱面数为: 700。

采用 SCAN 处理次序为: 100-132-190-205-376-398-61-40-29-23-19-18-4, 总柱面数为: 692。

图略。

## 第六章

8. 设有  $n$  个进程共享一个互斥段, 如果: ①每次只允许一个进程进入互斥段; ②每次最多允许  $m$  个进程 ( $m \leq n$ ) 同时进入互斥段。试问: 以上两种情况下所采用的信号量初值是否相同? 试给出信号量值的变化范围。

答: 所采用的互斥信号量初值不同。

①互斥信号量初值为 1, 变化范围为  $[-n+1, 1]$ 。

当没有进程进入互斥段时, 信号量值为 1; 当有 1 个进程进入互斥段但没有进程等待进入互斥段时, 信号量值为 0; 当有 1 个进程进入互斥段且有一个进程等待进入互斥段时, 信号量值为 -1; 最多可能有  $n-1$  个进程等待进入互斥段, 故此时信号量的值应为  $-(n-1)$  也就是  $-n+1$ 。

②互斥信号量初值为  $m$ , 变化范围为  $[-n+m, m]$ 。

当没有进程进入互斥段时, 信号量值为  $m$ ; 当有 1 个进程进入互斥段但没有进程等待进入互斥段时, 信号量值为  $m-1$ ; 当有  $m$  个进程进入互斥段且没有一个进程等待进入互斥段时, 信号量值为 0; 当有  $m$  个进程进入互斥段且有一个进程等待进入互斥段时, 信号量值为 -1; 最多可能有  $n-m$  个进程等待进入互斥段, 故此时信号量的值应为  $-(n-m)$  也就是  $-n+m$ 。

11. 设在公共汽车上, 司机和售票员的活动分别如下: ①司机的活动: 启动车辆; 正常行车; 到站停车; ②售票员的活动: 关车门; 售票; 开车门。在汽车不断到站、停车、行驶的过程中, 司机与售票员的活动之间存在哪些同步关系? 试用信号量和 PV 操作实现其同步。

应设置两个信号量：s1、s2；s1 表示是否允许司机启动汽车(其初值为 0)；s2 表示是否允许售票员开门(其初值为 0)。用 P、V 原语描述如下：

```
semaphore s1, s2;
s1=0;s2=0;
cobegin
    driver ( );
    busman ( );
coend
process driver ( ) {
    while(true) {
        P(s1);
        {启动车辆};
        {正常行车};
        {到站停车};
        V(s2);
    }
}

process busman ( ) {
    while(true) {
        {关车门};
        V(s1);
        {售票};
        P(s2);
        {开车门};
        {上下乘客};
    }
}
```

20. 一组生产者进程和一组消费者进程共享 9 个缓冲区，每个缓冲区可以存放一个整数。生产者进程每次一次性地向 3 个缓冲区中写入整数，消费者进程每次从缓冲区取出一个整数。请用：①信号量和 PV 操作；②管程，写出能够正确执行的程序。

答：①信号量和 P、V 操作。

```
var int buf[9];
int count,getptr,putptr;
count=0;getptr=0;putptr=0;
semaphoreS1,S2,SPUT,SGET;
S1=1;S2=1;SPUT=1;SGET=0;

process producer-i( ) {
    while(true) {
        {生产 3 个整数};
        P(SPUT);
        P(S1);
        buf[putptr]=整数 1;
        putptr=(putptr+1) % 9;
        buf[putptr]=整数 2;
        putptr=(putptr+1) % 9;
        buf[putptr]=整数 3;
        putptr=(putptr+1) % 9;
        V(SGET);
        V(SGET);
        V(SGET);
        V(S1);
    }
}

process consumer-j( ) {
    int y;
    while(true) {
        P(SGET);
        P(S2);
        y=buf[getptr];
        getptr=(getptr+1) % 9;
        count++;
        if (count==3)
            {count=0;V(SPUT);}
        V(S2);
        {consume the 整数 y};
    }
}
```

22. 现有三个生产者 P1、P2、P3，他们都要生产桔子水，每个生产者都已分别购得两种不同原料，待购得第三种原料后就可配制成桔子水，装瓶出售。有一供应商能源源不断地供应糖、水、桔子精，但每次只拿出一种原料放入容器中供给生产者。当容器中有原料时需要该原料的生产者可将其取走，当容器为空时供应商又可放入一种原料。假定：①生产者 P1 已购得糖和水；②生产者 P2 已购得水和桔子精；③生产者 P3 已购得糖和桔子精；试用：①管程；②信号量与 PV 操作，写出供应商和三个生产者之间能正确同步的程序。

②信号量与 P、V 操作

```
semaphore S,S1,S2,S3;
```

```
S=1;S1=S2=S3=0;
```

```
enum container{糖,水,桔子精};
```

```
cobegin
```

```
process 供应商() {
```

```
while(true) {
```

```
P(S);
```

```
{take material into container};
```

```
if(container==桔子精) V(S1);
```

```
else if(container==糖) V(S2);
```

```
else V(S3);
```

```
}
```

```
}
```

```
process P2() {
```

```
while(true) {
```

```
P(S2);
```

```
{take 糖 from container};
```

```
V(S);
```

```
{做桔子水};
```

```
}
```

```
}
```

```
coend.
```

```
process P1() {
```

```
while(true) {
```

```
P(S1);
```

```
{take 桔子精 from container};
```

```
V(S);
```

```
{做桔子水};
```

```
}
```

```
}
```

```
process P3() {
```

```
while(true) {
```

```
P(S3);
```

```
{take 水 from container};
```

```
V(S);
```

```
{做桔子水};
```

```
}
```

```
}
```

27. 设当前的系统状态如下，此时 Available=（1，1，2）。试回答下列问题：①计算各个进程还需要的资源数 C<sub>ki</sub>-A<sub>ki</sub>。②此时系统是否处于安全状态，为什么？③进程 P2 发出请求向量 request<sub>2</sub>（1，0，1），系统能把资源分配给它吗？④若在进程 P2 申请资源后，P1 发出请求向量 request<sub>1</sub>（1，0，1），系统能把资源分配给它吗？⑤若在进程 P1 申请资源后，P3 发出请求向量 request<sub>3</sub>（0，0，1），系统能把资源分配给它吗？

进程	Claim			Allocation		
	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
P <sub>1</sub>	3	2	2	1	0	0
P <sub>2</sub>	6	1	3	5	1	1
P <sub>3</sub>	3	1	4	2	1	1
P <sub>4</sub>	4	2	2	0	0	2

答：①P1, P2, P3, P4 的 Cki-Aki 分别为：(2, 2, 2)、(1, 0, 2)、(1, 0, 3)、(4, 2, 0)。

②系统处于安全状态，存在安全序：P2, P1, P3, P4。

进 程	Claim R1 R2 R3	Allocation R1 R2 R3	Need R1 R2 R3	Work R1 R2 R3	finish
P1	3 2 2	1 0 0	2 2 2	9 3 6	<del>F</del> T
P2	6 1 3	5 1 1	1 0 2	6 2 3	<del>F</del> T
P3	3 1 4	2 1 1	1 0 3	8 3 4	<del>F</del> T
P4	4 2 2	0 0 2	4 2 0	8 3 6	<del>F</del> T

③可以分配，存在安全序列：P2, P1, P3, P4。

④不可以分配，资源不足。

⑤不可以分配，不安全状态。

28. 将死锁检测算法用于下面的数据，试回答以下问题：①系统此时处于安全状态吗？②若第二个进程提出资源请求 request2 (0, 0, 1, 0)，系统能分配资源给它吗？③若第五个进程提出资源请求 request5 (0, 0, 1, 0)，系统能分配资源给它吗？

$$Need = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 \\ 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 \end{bmatrix} \quad Allocation = \begin{bmatrix} 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad Available = (1, 0, 2, 0)$$

答：①此时可以找出进程安全序列：P4, P1, P5, P2, P3。故系统处于安全状态。

②可以分配，存在安全序列：P4, P1, P5, P2, P3。

③不可分配，系统进入不安全状态。

进 程	Claim R1 R2 R3 R4	Allocation R1 R2 R3 R4	Need R1 R2 R3 R4	Work R1 R2 R3 R4	finish
P1	4 1 1 1	3 0 1 1	1 1 0 0	5 1 3 2	<del>F</del> T
P2	0 2 1 2	0 1 0 0	0 1 1 2	5 2 3 2	<del>F</del> T
P3	4 2 1 0	1 1 1 0	3 1 0 0	6 3 4 2	<del>F</del> T
P4	1 1 1 1	1 1 0 1	0 0 1 0	2 1 2 1	<del>F</del> T
P5	2 1 1 0	0 0 0 0	2 1 1 0	5 1 3 2	<del>F</del> T