

■ 第一部分

同步和互斥问题

July 25 J

间接 第一定派一部

- 1. 进程之间存在着哪几种制约关系? 各是什么原因引起的? 下列活动分别属于哪种制约关系?
 - (1) 若干同学去图书馆借书; 外
 - (2) 两队举行篮球比赛; 为一
 - (3) 流水线生产的各道工序; 10 7
 - (4) 商品生产和社会消费。

解: 研制 图参写作

- 进程之间存在两种制约关系,即间接相互制约关系(互 斥)和直接相互制约关系(同步)。互斥是由于并发进 程之间竞争系统的临界资源引起的;同步是由于进程间 的相互合作而引起的。
- (1)是间接制约(互斥)关系, 书是临界资源。
 - (2)是间接制约(互斥)关系,篮球是临界资源。
 - (3)是直接互制约(同步)关系,一个工序完成后才能开始下一个工序。
 - (4)是直接互制约(同步)关系,商品生产出来后才能被消费,商品被消费后才需要再生产。

② 2. 桌上有个能盛得下五个水果的空盘子。爸爸不停地向盘中放苹果或橘子,儿子不停地从盘中取出橘子享用,女儿不停地从盘中取出苹果享用。规定三人不能同时从盘中取放水果。试用信号量实现爸爸、儿子和女儿这三个循环进程之间的同步。

Wait - 申请资源 Signal - 解放~

解

```
semaphore empty=5, apple=0, orange=0, mutex=1;
Dad {
    while (true) {
        wait(empty);
        wait(mutex);
        将水果放入盘中;
        signal(mutex);
        if(放入的是橘子) signal(orange);
        else signal(apple);
```

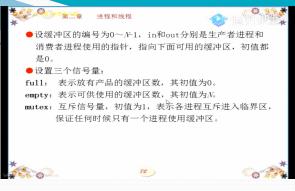
```
son {
                             daughter {
 while (true) {
                                  while (true) {
        wait(orange);
                                     wait(apple);
        wait(mutex);
                                     wait(mutex);
        从盘中取一个橘子:
                                     从盘中取一个苹果;
        signal(mutex);
                                     signal(mutex);
        signal(empty);
                                     signal(empty);
        享用橘子;
                                     享用苹果;
```

- 3. 有一个计算进程和打印进程,他们共享一个单缓冲区,计算进程不断地计算出一个整型结果并将它放入单缓冲区中,打印进程则负责从单缓冲区中取出每一个结果进程打印,请用信号量编程来实现它们的同步关系。
- 分析: 是一个n=1的生产者-消费者问题。



解

```
semphore full=0,empty=1;
int buffer;
cp(){
  int nexte;
  while(1){
  compute the next number in
nextc;
  wait(empty);
  buffer=nextc;
  signal(full);
```



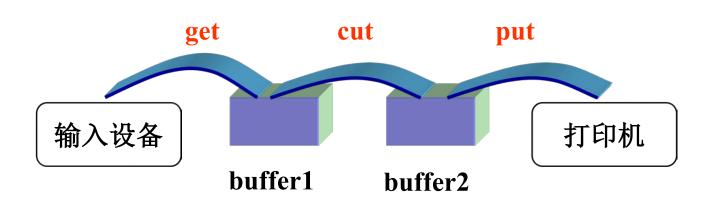
```
pp(){
  int nextp;
  while(1){
  wait(full);
  nextp=buffer;
  signal(empty);
  print the number in
nextp;
```

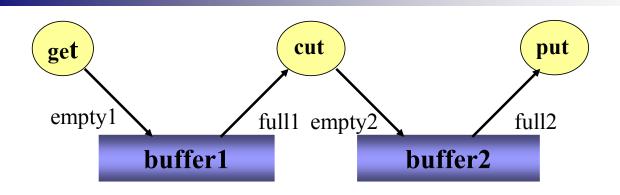
- 4. 请用信号量解决以下的"过独木桥"问题:同一方向的行人可连续过桥,当某一方向有人过桥时,另一方向的行人必须等待;当某一方向无人过桥时,另一方向的行人可以过桥。
- 分析: 类似读者-写者问题的一个变型。

解:

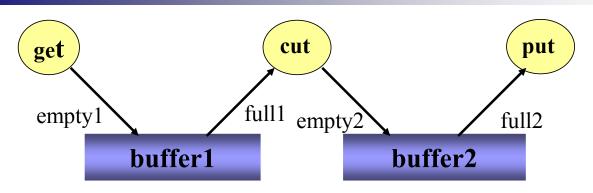
```
int countA=0, countB=0; //表示A、B方向上已在独木桥上行人数 semphore mutexA=mutexB=1; //分别用来实现对countA和countB的互斥
访问
semphore mutex=1; //实现两个方向的行人对独木桥的互斥使用
A方向的行人:
PA(){
  wait(mutexA);
  if(countA==0) wait(mutex);
  countA++;
  signal(mutexA);
  通过独木桥:
  wait(mutexA);
  countA--:
  if(countA==0) signal(mutex);
  signal(mutexA);
B方向行人的算法与上述算法类似,只需将其中的mutexA改成mutexB,countA改为countB即可。
```

■ 5.有三个进程,进程get从输入设备上不断读数据,并存入buffer1;进程cut不断将buffer1的内容剪切到缓冲区buffer2,进程put则不断将buffer2的内容在打印机上输出。三个进程并发执行,协调工作。写出该三个进程并发执行的同步模型。





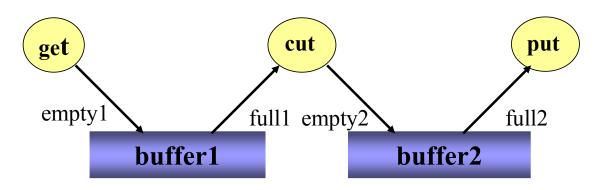
- 分析存在如下同步关系:
- (1) 只有buffer1为空,get才能工作,并使buffer1为满。
- (2)要求buffer1为满,同时buffer2为空,cut才能工作,工作结果使buffer1为空,buffer2为满。
- (3) 只有buffer2为满,put才能工作,并使buffer2为空。



■ 解答:

(1) 设置信号量

- 设信号量empty1表示缓冲区buffer1为空,初值为1
- 设信号量full1表示缓冲区buffer1为满,初值为0
- 设信号量empty2表示缓冲区buffer2为空,初值为1
- 设信号量full2表示缓冲区buffer2为满,初值为0



(2) 同步算法描述如下:

```
cut()
get()
                                              put()
{ while(true)
                      { while(true)
                                             { while(true)
 { wait(empty1)
                         wait(full1)
                                                 wait(full2)
   get操作
                                                 put操作
                         wait(empty2)
                         cut操作
   signal(full1)
                                                 signal(empty2)
                         signal(empty1)
                         signal(full2)
```

```
semaphore empty1=1,full1=0,empty2=1,full2=0;
get()
{ while(true)
 { wait(empty1);
   get操作;
   signal(full1);
cut()
{ while(true)
  wait(full1);
   wait(empty2);
   cut操作;
   signal(empty1);
   signal(full2);
```

```
put()
{ while(true)
   wait(full2);
    put操作;
    signal(empty2);
main()
{ parbegin(get,cut,put);
```

■ 6.用PV操作解决司机和售票员的问题。

- 分析存在如下同步关系:
- (1) 售票员关门后,司机才能开车。
- (2)司机到站停车后,售票员才能开门。
- 设信号量close表示关门开车,初值为0;
- 设信号量stop表示到站开门,初值为0



```
semaphore close=0, stop=0;
driver()
{ while(true)
 { wait(close)
   启动车辆:
   正常行驶:
   到站停车;
   signal(stop)
```

```
seller()
{ while(true)
{ 关门
   signal(close)
  售票
   wait(stop)
  开门
main()
{ parbegin(driver, seller);
```

■ 7. 有一个阅览室,共有100个座位,读者进入时必须 先在一张登记表上登记,该表为每一位列一表目,包 括座号和读者姓名等,读者离开时要消掉登记的信息。 试用PV操作描述读者进程之间的同步关系。

- 分析:读者填表进入阅览室,这时要考虑阅览室里是 否有座位;同时还要考虑登记表的互斥使用。
- 设信号量seats=100, mutex=1。前者用于约束只能有100个进程共享阅览室,后者用来约束登记表的互斥使用。

semaphore seats=100

```
reader()
{ while (true)
             wait (seats);
             wait (mutex);
             填写登记表;
             signal (mutex);
             选书读书;
             wait (mutex);
             消掉登记;
             signal (mutex);
             signal (seats);
```

■ 8. 某小型超级市场,可容纳50个人同时购物。入口处 备有篮子,每个购物者可拿一只篮子入内购物。出口 处结帐,并归还篮子(出、入口禁止多人同时通过)。

试用PV操作写出购物者的同步算法。

- 1
 - 设信号量empty=50, mutex1=1, mutex2=1。empty用于约束 只能有50个人共享超市, mutex1用来约束入口的互斥使用, mutex2用来约束出口的互斥使用。
 - 算法描述如下:

```
Consumer(int i)
  wait(empty);
  wait(mutex1);
  从入口进入超市并拿篮子:
  signal(mutex1);
  选购商品;
  wait(mutex2);
  到出口结账并归还篮子:
  signal(mutex2);
  signal(empty);
```

■ 第二部分

处理机调度

■ 1. 系统 有4个进程: P1,P2, P3,P4。它们的到达时间以及估计 运行的时间如下图所示:

进程	到达时间(ms)	估计运行时间(ms)
P1	0	20
P2	5	15
Р3	10	5
P4	15	10

■ 请计算使用下述调度算法时,进程的执行顺序、周转时间、 带权周转时间以及进程流的平均周转时间和平均带权周转 时间。

(1)FCFS) (2)SJF (3)HRRN

(1) 采用FCFS算法时,先到达系统的作业先调度执行工士 所以调度顺序为P1、P2、P3、P4。

进程	到达 时间	估计运 行时间	开始执 行时间	完成时间	周转时间	带权周转时间
P1	0	20	0	20	20	20/20=1
P2	5	15	20	35	30	30/15=2
P3	10	5	35	40	30	30/5=6
P4	15	10	40	50	35	35/10=3.5

■ 平均周转时间为(20+30+30+35)/4=28.75ms 平均带权周转时间为W=(1+2+6+3.5)/4=3.125ms (2) 采用SJF算法时,优先调度到达的短作业,所以调度顺序为P1、P3、P4、P2。

进程	到达 时间	估计运 行时间	开始执 行时间	完成时间	周转时 间	带权周转时 间
P1	0	20	0	20	20	20/20=1
P2	5	15	35	50	45	45/15=3
P3	10	5	20	25	15	15/5=3
P4	15	10	25	35	20	20/10=2

■ 平均周转时间为(20+15+20+45)/4=25ms

平均带权周转时间为: (1+3+2+3)/4=2.25ms

R2=(15+15)/15=2, R3=(10+5)/5=3, R4=(5+10)/10=1.5 所以再调度P3。

P3的周转时间为10+5=15ms; 带权周转时间为: 15/5=3

R2=(20+15)/15=2.33, R4=(10+10)/10=2

所以调度P2执行。

P2的周转时间为20+15=35ms; 带权周转时间为: 35/15=2.33 最后调度P4执行。

P4的周转时间为25+10=35ms; 带权周转时间为: 35/10=3.5 综上, 调度顺序为P1、P3、P2、P4。

平均周转时间为(20+15+35+35)/4=26.25ms

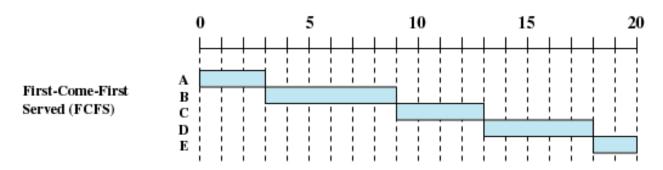
平均带权周转时间为 (1+3+2.33+3.5)/4=2.46ms

■ 2. 系统 有5个进程: A, B, C, D, E。它们的到达时间以及估计运行的时间如下图所示:

进程	到达时间(ms)	估计运行时间(ms)
A	0	3
В	2	6
C	4	4
D	6	5
E	8	2

- 请计算使用下述调度算法时,进程的周转时间和进程流的 平均周转时间。
- (1)FCFS (2) SJF (3) HRRN (4) RR(q=1)

(1) FCFS算法:按照先来先服务原则,调度顺序为A,B,C,D,E,详细情况如下图所示



因此,A的周转时间为3ms;B的周转时间为9-2=7ms;

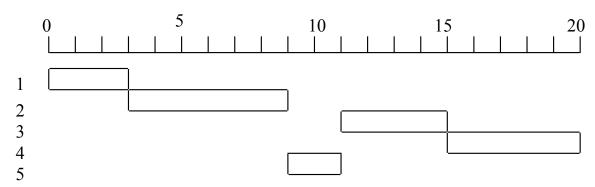
C的周转时间为13-4=9ms;

D的周转时间为18-6=12ms;

E的周转时间为20-8=12ms;

平均周转时间为(3+7+9+12+12)/5=8.6ms

(2) SJF算法:按照短进程优先原则,调度顺序为A,B,E,C,D,详细情况如下图所示



因此,A的周转时间为3ms;B的周转时间为9-2=7ms;

C的周转时间为15-4=11ms;

D的周转时间为20-6=14ms;

E的周转时间为11-8=3ms;

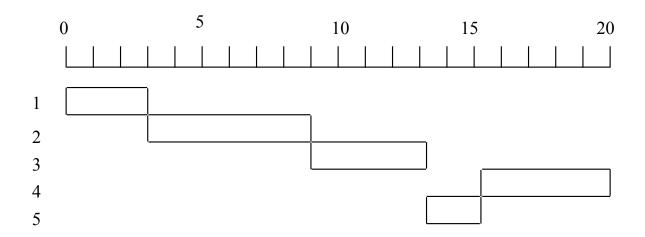
平均周转时间为(3+7+11+14+3)/5=7.6ms

.

(3) HRRN算法:

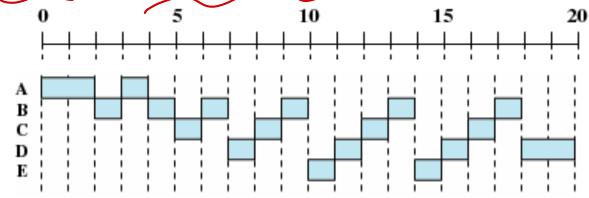
- 初始时刻只有A,因此先调度A执行。 A的周转时间为3ms。
- B执行后, C,D,E均就绪,按照最高响应比调度原则, Rc=(9+4-4)/4=2.25, Rd=(9+5-6)/5=1.6, Re=(9+2-8)/2=1.5 因此再调度C执行, C的周转时间为13-4=9ms;
- C执行后, Rd=(13+5-6)/5=2.4, Re=(13+2-8)/2=3.5, 因此调度 E执行, E的周转时间为15-8=7ms;
- 最后调度D, D的周转时间为20-6=14ms

■即,调度过程如图所示



■ 平均周转时间为(3+7+9+7+14)/5=8ms

(4)RR (q=1)算法:按照时间片轮转法原则,调度情况如下 图所示 0 5 10 15



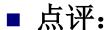
因此,A的周转时间为4ms; B的周转时间为18-2=16ms;

C的周转时间为17-4=13ms;

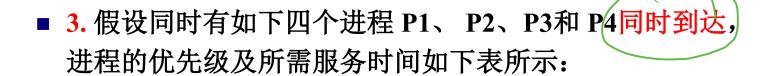
D的周转时间为20-6=14ms;

E的周转时间为15-8=7ms;

平均周转时间为(4+16+13+14+7)/5=10.8ms



□各种调度算法下,进程的调度顺序及调度执行 的详细过程需要以图或文字的形式进行说明。 仅给出最后的结果是不完整的。



进程	优先级	所需服务时间(ms)
P1	2	4
P2	5	3
Р3	4	5
P4	3	2

- 考虑如下调度算法,各进程的调度顺序及周转时间。
- (1)最高优先级HRS (2)轮转法RR (q=1, q=2)

(1) 采用最高优先级HRS算法时,调度顺序为:

P2->P3->P4->P1。

P2的周转时间为3ms;

P3的周转时间为3+5=8ms;

P4的周转时间为3+5+2=10ms;

P1的周转时间为3+5+2+4=14ms;

平均周转时间为(3+8+10+14)/4=8.75ms

(2) 采用RR轮转法时,假定时间片q=1ms

- 第一轮调度: P1P2P3P4 (4)
- 第二轮调度: P1P2P3P4 (4) P4执行结束 P4周转时间为4+4=8ms
- 第三轮调度: P1P2P3 (3) P2执行结束 P2周转时间为8+2=10ms
- 第四轮调度: P1P3 (2) P1执行结束 P1周转时间为11+1=12ms
- 第五轮调度: P3 (1) P3执行结束 P3周转时间为13+1=14ms
- 平均周转时间为(8+10+12+14)/4=11ms

(2) 采用RR轮转法时,假定时间片q=2ms

■ 第一轮调度: P1P2P3P4 (8) P4执行结束

2 2 2 P4周转时间为8ms

■ 第二轮调度: P1P2P3 (5) P1、P2执行结束

2 1 2 P1周转时间为8+2=10ms

P2周转时间为8+2+1=11ms

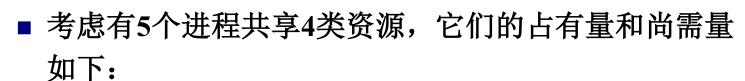
■ 第三轮调度: P3 (1) P3执行结束

1 P3周转时间为13+1=14ms

■ 平均周转时间为(8+10+11+14)/4=10.75ms

■ 第三部分

死 锁



进程	已分配	尚需	可用
P0	0023	0 0 1 2	1623>
P1	1000	2000	
P2	0 3 3 0	0140	91.2,0
Р3	1010	0813	
P4	0014	3000	

- (1) 当前状态安全吗?
- (2) 如果进程2提出资源请求 (0,1,2,0), 按照银行家算法, 能否满足要求? ✓

1503

(1)利用安全性算法对上述状态进行分析,如下表所示,

		1	- Phila	
	Work	Need	Allocation	Finish
	ABCD	ABCD	A B C D	
P0	1 6 2 3	0 0 1 2	0 0 2 3	true
P2	1 6 4 6	0 1 4 0	0 3 3 0	true
Р3	1 9 7 6	0 8 1 3	1 0 1 0	true
P1	2 9 8 6	2 0 0 0	1 0 0 0	true
P4	3 9 8 6	3 0 0 0	0 0 1 4	true

因为存在安全序列 {P0, P2, P3, P1, P4}, 所以系统是安全的。

- (2) P2发出请求(0,1,2,0)后,系统按银行家算法进行检查:
 - \bigcirc Request(0,1,2,0)<=Need₂(0,1,4,0);
 - ②Request(0,1,2,0)<=Available(1,6,2,3);</pre>
 - ③试探为P2分配资源,并修改数据:

Available=(1,5,0,3); Allocation₂=(0,4,5,0)Need₂=(0,0,2,0)°.

- ④进行安全性检查,发现Available(1,5,0,3)<Need_i,即Available不能满足任何进程的请求,故系统进入不安全状态。
- 因此,P2提出的上述请求不能满足,系统不能分配资源给它。

■ 第四部分

存储管理



■ 1. 若在一分页存储管理系统中,某作业的页表如下所示。已知页面大小为1KB,试将逻辑地址1021、4012、5014分别转化为相应的物理地址。要求写出计算过程。

页号。	物理块号。
0.0	20
1_{ℓ}	3 _{\varphi}
2.	1.0
3₽	60

- ■解:本题中,页面大小L=1KB=1024。设页号为P,页内位移为D,则:
- (1) 对于逻辑地址1021 P=INT(1021/1024)=0, D=1021 mod 1024=1021, 查页表第0页得到其在内存中的物理块号为2, 所以物理地址为: 2*1024+1021=3069。
- (2) 对于逻辑地址4012, P=INT(4012/1024)=3, D=4012 mod 1024=940, 查页表第3页的带其在内存中的物理块号为6, 所以物理地址为: 6*1024+940=7084。
- (3) 对于逻辑地址5014, P=INT(5014/1024)=4, D=5014 mod 1024=918, 因页号4超过页表长度,产生越界中断,故该逻辑地址非法。

- 2. 某系统采用页式存储管理策略,拥有逻辑空间32页,每页2KB,拥有物理空间1MB。
 - (1) 写出逻辑地址的格式;
- (2) 若不考虑访问权限,进程的页表有多少项?每项至少有多少位?
- (3) 若物理空间减少一半,页表结构应相应做怎样的改变?

7年

■ 答:

(1)由于该系统拥有32个逻辑页面,所以页号必须用5位表示;而每页2K,页内地址需要11位来描述。故逻辑地址格式为: 15 11 10 0

页号 页内地址

- (2) 进程的页表项最多为32项;若不考虑访问权限,则每个页表项中只需给出对应的物理块号。1MB的物理空间可分为29个内存块,故每个页表项至少有9位。
- (3)如果物理空间减半,则页表中页表项数不变,但每项的长度可减少1位。)

3. 对应不表所示的段表,请将逻辑地址(0,137), (1,4000), (2,3600), (5,230)转换成物理地址。

段号	内存始址	段长
0	50K	10K
1	60K	3K
2	70K	5K
3	120K	8K
4	150K	4K

■ 答:

(1) 段号0、段内地址137均合法;

对应的物理地址为: 50K+137==50*1024+137=51337.

- (2) 段号1合法;段内地址4000超过段长3K,因此产生越界中断。
- (3) 段号2、段内地址3600均合法;

对应的物理地址为: 70K+3600=75280.

(4) 段号5不合法,产生越界中断。(注:段表的长度为5,段号为:0~4,所以段号5不合法.)

- 4. 设作业的虚拟地址为16位,其中高6位为段号,低 10位为段内相对地址。试问:
- (1)一个作业最多可以有多少个段?每段的最大长度是多少字节?
- (2) 某作业的请求段表如下,试计算[0,230]、[1,50]、[2,130]、[3,70]的主存地址。其中方括号内的前一元素为段号,后一元素为段内地址。

段号	段长	主存起始地址	是否在主存
0	400	1200	是
1	80	_	否
2	100	3100	是
3	120	4000	是

解: (1)由于高6位为段号,低10位为段内相对地址,所以作业最多允许有: (26=64个段;每段的最大长度是: 210=1024个字节。

←2) 虚拟地址 [0,230]:

查请求段表可知: 0段的主存起始地址为1200,段长为400,故其物理地址为: 1200+230=1430;

[1,50]: 由于该虚拟地址的段号为1,查请求段表可知,1段不在内存中,故发生缺段中断;

[2,130]:由于2段的段长为100,而130>100,故会发生越界中断。

[3,70]的物理地址为: 4000+70=4070.

.

■ 5. 在请求分页存储管理中,若某个作业的页面访问串为"4、3、2、3、5、3、5、4、3、2、4、5",分别采用OPT、FIFO、LRU页面置换算法,当分配给该作业的物理块数为3时,试计算在访问过程中所发生的页面置换次数和缺页次数,并计算缺页率是多少?

(1) 采用OPT置换算法,共发生2次页面置换, 5次缺页;缺页率=5/12=42%.

4	3	2	3	5	3	5	4	3	2	4	5
4	4	4		4					4		
	3	3		3					2		
		2		5					5		
x	×	×		x					×		

(2) 采用FIFO置换算法,共发生5次页面置换, 8次缺页;缺页率=8/12=67%.

4	3	2	3	5	3	5	4	3	2	4	5
4	4	4		5			5	5	2		2
	3	3		3			4	4	4		5
		2		2			2	3	3		3
*	x	x		x			×	×	x		x

(3) 采用LRU置换算法, 共发生4次页面置换, 7次缺页; 缺页率=7/12= 58%.

4	3	2	3	5	3	5	4	3	2	4	5
4	4	4		5			5		2		2
	3	3		3			3		3		5
		2		2			4		4		4
×	*	×		×			sc		*		*