????

名次解释、简答

- 操作系统_____
- 进程调度_____
- 磁盘调度_____

????

?????

- 1.???:??????????????????????????, ?????P?V ??????
- 2.??:
  ??
  ??????????????????????????????????
- 3.????:
  ??
  ??????
- 4.?????:
  ??
  ?,???????????,
  ??
  ??
  ??????????????????????
- 5.?????:????0S??????????????????0S??????????(
  ????????)???????:??????????????(????????)
  ???????????;????????????????????????????? I/0
  ???
- 6.??:?????????????
- 7.??????:
  ???CPU
  ??????????

- 8.?? :??????????????????????????????
- 9.?? ??????????????????????????????
- 10.??????????????????,???????????????, ??
- 11.??????????????????????????:???????????, ??????,???????
- 12.???:??????????????????????
- 13.????, ????????????????????????????????????(?????)????????????????????????????
- ??????????
  - ??????
  - ??????
  - ?????
  - ????
  - ???????
- ?????????:
  - 1.???????????????
  - 2.????????????????????????
  - 3.?????????????????????????????????
  - 4.???????????????

- 5.??????????????????????????

- ????????:
  - ????????
  - ????????
  - ??????????
  - ????????:
    - ???????
    - ???????-
    - ????????
    - ????????
- ???????:
  - ??????????????????????
  - ??????????????????????????
  - ????????????????????????????????
  - ??

14??????????????,??????????,
??????????????????????????????

??????

- ????????????????

- ??????????????????,????????
- ??,???????,?????????????
- ????,??????????????????????????
- ???????????????

??????????????????????????????

??????????????:

- 1.??????????????????
  - ??????????????????????????????????
  - ????????????????????????
  - ??????????????????????????????????????,????????
- 2.??????????????????????????????????
- 3.??????????????????????????????????

???5???????:

- ?????:?????????????CPU??????????
- ?????:??????????????????????
- ?????:??????????????I/0????????CPU?????????????????
- ?????:??????????????????I/0?????????????

15. ??????:??????????????????,
??????????????????????????????,
??????????????????????????

??????:??????????????????????????????????????,
????????????????

??:
???

?????????:

- 1.???????????????????????????
- 2.??
- 3.??
- 4.??

???????:???????????3??

?????????:

- 1.????????????????????
- 2.???????????????????????
- 3.??????????

16. ???????????:

- ??????????????
- ????????????????????????
- ????????????

CPU????????4??:

- 1. ??
????????????????????????????????????CPU????

- 2.????????????????????????????

- 3.??CPU
???????CPU??????????????????????????CPU????????????

- 4.??

?????????:

- 1.????????????????????????

- 2.???????????????????

- 3.?????????????????

- 4.????????????????

?????????:

- 1.???????????????

- 2.????CPU????

- 3.??????????

??????????

- 轮转法只能用在进程调度里面，轮转法的基本思路是让每个进程在就绪队列中的等待时间与享受服务的时间成正比，轮转法只能用来分配那些可以抢占的资源，将他们随时剥夺，再分配给别的进程。
- 由于作业调度室对除了 CPU 之外的所有系统硬件资源的分配，其中包括不可抢占资源，所以作业调度不使用轮转法。
- 在轮转法中时间片长度的选取非常重要，如果时间片长度过短，则调度程序剥夺 CPU 的次数增多，使得系统开销大，如果时间片长度过长，轮转法会变成先来先服务法。

17.????????????????????:????????????????

??????????:

- 1.????????????
- 2.??????????????????
- 3.??????????

????:
??
??????????????????????

?????????:
??
?????固定分区和动态分区有两种方法?

????????:

- ??:
  - 1.??????????????????????????????????????
  - 2.??????????????????????????
- ??:
  - 1.????????
  - 2.??????????????????
  - 3.??????????????

???????:

- ??????????????????
- ??????????????????

- ??????????????????????
- ??
  ????
- ????????????????????????????????
- ????????????????????????????????
- ??????????????????????????

????????:

- ??:
  - ??
  - ??????????????
- ??:
  - ???????????
  - ???????????? ???????????

????????:

- □□:
  - ????????????????????????????????
  - ??
  - ?????????
- □□:

- ??????????????
- ??????????????????????????????
- ????????????????????????????????????

??

??????????

- (1)????????????????????????????????????
- (2)??
- (3)?????????????
- (4)???????????
- (5)?????????????

?????????

- ????????????????????????????????
- ????????????????????????,??????????????????????
- ??????????????,????????????????????????
- ????????????????????,??????????????????????;
- ??,????????,??????????????,??????????????

18. ???????????:

- 1.????????????????????????????????
- 2.??????????????

- 3.??????????????????????

?????:

- 1.???????????:?????????????
- 2.???????????:?????????????
- 3.?????????:???????????????
- 4.????????????:??????????????????????

???????:

- 1.???????????????
- 2.??????????
- 3.??????

??????????????????????:

- 1.???????
- 2.??????
- 3.????
  - ??????????????????????????????????
  - ??????????????????????????????????
- 4.?????

?????????:

- ??
- ???????????
- ???????????
- ??
  ????
- ??
- ????????????????????????????????
- ??
- ????????????????????????????????????
- ??????????????????????

19. I/O?????????????:

- 1.??????????????????????????CPU?I/O??????????
- 2.???
- 3.DMA??????????
- 4.???

??:????????????????????????????????????CPU
??
???

????:CPU??????????????????????

????:CPU??????????????????????????

???????????:

- 1.???CPU???????????
- 2.??????????????
  - ???????????????
  - ??????????????????????
  - ????????
- 2.??????????????????????????
- 3.??????????????CPU??????????????????????
- 4 ??

??????:

- 1.?????????CPU?????????????
- 2.??CPU?????????????
- 3.????????
- 4.???????????????
- 5.?????????
- 6.?????????
- 7.???CPU???????????

???????:

- ????????????
  - ????

  - ????

  - ???

  - ????

- ?????I/O???CPU???????????I/O???CPU ?????????????????????????????CPU??????????

- ???????I/O???CPU ??

- ???

- ???????????????????????????????????

大题:

1. 设有 5个哲学家吃饭,。。。试解答以下问题:(1)描述一个保证不会出现两个邻座同时要求吃饭的通信算法。(2)描述一个既没有两邻座同时吃饭,又没有人饿死(永远拿不到筷子)的算法。

在什么情况下,5个哲学家全部吃不上饭?

答: (1)设信号量 c[0]~c[4],初始值均为 1,分别表示 i号筷子被拿 (i=0,1,2,3,4)。

```
send(i):第 i个哲学家要吃饭

begin

P(c[i]);

P(c[i+1 mod 5]);

eat;

V(c[i+1 mod 5]);

V（c[i]）;

end
```

该过程能保证两邻座不同时吃饭,但会出现 5个哲学家一人拿一支筷子,谁也吃不上饭的死锁情况。

(2)解决的思路如下:让奇数号的哲学家先取右手边的筷子,让偶数号的哲学家先取左手边的筷子。这样,任何一个哲学家拿到一支筷子以后,就已经阻止了他邻座的一个哲学家吃饭的企图,除非某个哲学家一直吃下去,否则不会有人会饿死。

```
Send(i):
begin
    if i mod 2==0
    then
    {
        P(c[i]);
        P(c[i+1] mod 5);
        eat;
        V(c[i]);
        V(c[i+1 mod 5]) }
    else
    {
        P(c[i+1 mod 5]);
        P(c[i]);
        eat;
        V(c[i+1 mod 5]);
        V(e[i]); }
end
```

2. 银行家算法是经典的死锁避免算法,请简述该算法

答: 银行家算法的基本思想是,进程在进入系统时,必须申明在运行过程中可能需要每种资源类型的最大单元数目,其数目不应超过系统拥有的资源总量。当进程请求一组资源时,系统必须首先确定是否有足够的资源分配给该进程。若有,再进一步计算在将这些资源分配给该进程后是否会使系统处于不安全状态。如果不会,才将资源分配给它,否则让该进程等待。

3.调度算法求周转时间:

表中写入作业号进入系统时间，运行时间，开始时间，结束时间，周转时间。

周转时间 ＝实际完成时间 -进入系统时间

平均周转时间 ＝周转时间相加 /作业数

带权周转时间 ＝周转时间 /运行时间

平均带权周转时间 ＝带权周转时间 /作业数

先来先服务（按照到达时间先后）。短作业优先（非抢战式按时间短的来）。

4.页面置换算法

计算缺页次数，页面置换次数和缺页率。

最佳置换算法 OPT:向后看（后面有的不置换）。

先进先出页面置换算法 FIFO:每次置换先来的。

最近最久未使用置换算法 LRU:向前看，前面有的不置换。

写了几列就是几个缺页次数，缺页次数 /总页数 ＝缺页率

置换了几次就是页面置换次数

命中率 ＝1-缺页率

5.分页式内存分配题

已知页号和块号，已知每页有多少个字节（页面大小），给出逻辑地址，求对应的物理地址。

逻辑地址 /页面大小 ＝p（页号） ....d（位移量）。

物理地址 ＝p对应的块号 ×页面大小 ＋位移量

6.分段式存储内存分配题

已知段号，段长，基址和状态位（ 0为主存， 1为主存）。

求逻辑地址（ 1， 55）对应的主存地址（也就是物理地址）。 1为段号， 55为段内地址。

答 ：

首先将段内地址与段长（逻辑地址的段号对应的段长）作比较，如果段内地址小于段长，没发生越界中断，物理地址 ＝段内地址 ＋基址。

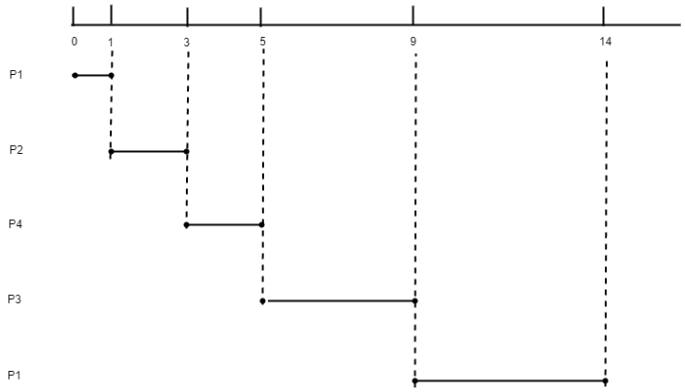
如果段内地址大于段长，发生越界，逻辑地址非法

????

1. 最短剩余时间优先调度算法
是一种抢占式的进程调度算法，其核心思想是根据进程的剩余执行时间动态选择最短者优先执行。

题 36 表

进程	到达时刻 t	运行时间
P1	0	6
P2	1	2
P3	1	4
P4	3	2



最短剩余时间算法
周转时间：执行时间-到达时间
P1: 14 -0 = 14

P2: 3 -1 = 2

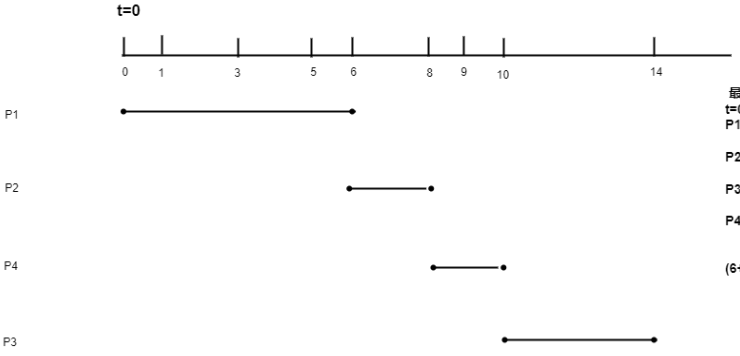
P3: 9 -1 = 8

P4: 5 -3 = 2

平均执行时间：（14+2+8+2）/ 4 = 6.50

1. 最短作业优先算法定义
SJF是一种非抢占式的作业调度算法，其核心思想是优先调度预计执行时间最短的作业，以减少平均周转时间和提高系统吞吐量。
若后来的短作业（进程）到达时，当前作业已开始执行，则不抢占，继续执行当前作业直到完成。

t=0



最短作业优先算法定义
t=0时，周转时间：执行时间-到达时间
P1: 6-0 = 6

P2: 8 -1 = 7

P3: 14 -1 = 13

P4: 10 -3 = 7

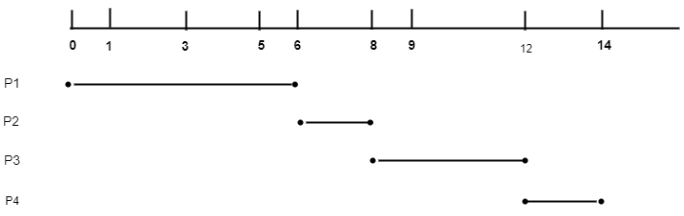
(6+7+13+7)/(6+2+4+2)=33/14

t=2时，根据此算法首先调度P2, 运行时间2+开始调度时间（2）=4
加权平均调度时间（4-1）/2=1.5

题 36 表

进程	到达时刻 t	运行时间
P1	0	6
P2	1	2
P3	1	4
P4	3	2

先来先服务调度（FCFS: First-Come, First-Served）非抢占式



先来先服务
t=0时，周转时间：执行时间-到达时间
P1: 6-0 = 6

P2: 8 -1 = 7

P3: 12 -1 = 11

P4: 14 -3 = 11

(6+7+11+11)/(4)=33/4=8.75

非抢占式多级反馈队列
系统维护多个就绪队列，每个队列对应不同的优先级。
通常队列优先级从高到低排列（如$Q_0 > Q_1 > Q_2$），高优先级队列分配较短的时间片，低优先级队列时间片较长。

点	到达时间	服务时间
P1	0	30
P2	10	60
P3	20	40
P4	30	50
P5	50	30

????

1	先来先服务	按顺序执行
2	最短寻找时间优先	找下一个距离最短的
3	电梯调度	按一个方向走完，然后反向走（与4不同是电梯调度是双向的）
4	单向扫描	按一个方向走完，然后回到最开头重新走

39. 假设对磁盘的请求串为柱面号 98、183、38、123、13、125、67、71，磁头的初始位置为 33，求在下列移臂调度[redacted]和移动臂需移动的距离。

(1) 先来先服务调度算法；

(2) 单向扫描调度算法（向柱面号增大的方向）。

(1) 33→98→183→38→123→13→125→67→71

65+85+145+85+110+112+58+4

=150+230+222+62

=380+284=664

(2) 33→38→67→71→98→123→125→183→13

5+29+4+27+25+2+58+183+13

=38+54+71+183

=92+254=346

等于64346啊

39. 设一移动头磁盘系统，共有 200 个柱面，编号为 0-199。磁盘请求以柱面号 10、100、191、31、20、150、32 的次序到达，当前磁头在 98 号柱面上。求在下列移臂调度算法下的服务次序和移动臂总共需移动的距离。

(1) 先来先服务调度算法；

(2) 移动臂由外向里移动（向柱面号增大的方向）的电梯调度算法。

(1) 98→10→100→191→31→20→150→32

88+90+91+160+11+130+118

=688

(2) 98→100→150→191→32→31→20→10

2+50+41+159+1+11+10

=274

这个就等于274啊这个题做完了

39. 假设磁盘的移动臂现在第 8 号柱面上，有 6 个访盘请求在等待，如题 39 表所示。

题 39 表

访盘请求序号	柱面号	磁头号	扇区号
①	9	6	3
②	7	5	6