

## 北京交通大学 2013 年硕士研究生入学考试试卷参考答案

科目代码: 923 科目名称: 操作系统原理

注意事项: 答案一律写在答题纸上, 写在试卷上的答案不予装订和评分

一、单项选择题。每道题有四个备选答案, 请从中选择一个最符合题目要求的答案填入空格中 (每题 3 分, 共 60 分)

1. 引入多道程序最重要的目的是 ( )

- A. 充分利用 CPU
- B. 提高实时响应速度
- C. 有利于代码共享
- D. 提高外部设备利用率

参考答案: 【A】

系统中只有一道程序时, CPU 不能得到充分的利用, I/O 的速度比 CPU 的速度低, CPU 处于空闲状态的时间比较多。引入多道程序最重要的目的就是提高 CPU 的利用率, 减少 CPU 的空闲时间。

2. 与命令介面相比, 图形用户介面可以提高操作系统的 ( )

- A. 方便性
- B. 有效性
- C. 可扩展性
- D. 开放性

参考答案: 【A】

方便性和有效性是设计操作系统时最重要的两个目标。在过去的很长一段时间内, 由于计算机系统非常昂贵, 因而其有效性显得比较重要。但是, 近十多年来, 随着硬件越来越便宜, 在设计配置在微机上的 OS 时, 人们似乎更重视如何使用户能更为方便地使用计算机, 故在微机操作系统中都配置了受到用户广泛欢迎的图形用户界面。

与命令介面相比, 图形用户界面极大地方便了用户使用计算机, 使中小學生都能很快地学会上机操作, 这无疑会更加推动计算机的迅速普及。

3. 与先来先服务、高响应比优先调度算法相比, 短作业优先的优点是 ( )

- A. 各类资源都能平衡利用
- B. 短作业不会长期等不到处理
- C. 吞吐量最大
- D. 响应时间最快

参考答案: 【C】

我们可以用排除法来解决这个问题。首先, 短作业优先、先来先服务、高响应比优先这几种算法,

都能保证短作业不会长期得不到执行，排除 B 答案。A、D 不是短作业优先优点的体现。

短作业优先(SJF)的调度算法是从后备队列中选择一个或若干个估计运行时间最短的作业，将它们调入内存运行。故而，短作业优先算法能够有效地降低作业的平均等待时间，提高系统吞吐量。

4. 将操作系统中的资源进行线性排队编号，进程对资源的请求未必按照资源的序号递增次序提出。这样可以通过摒弃（ ）条件预防死锁。

- A. 请求与保持
- B. 不剥夺
- C. 环路等待
- D. 互斥

参考答案：【C】

[1] 请求和保持条件：进程已经保持了至少一个资源，但又提出了新的资源请求，而该资源又已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。破坏该条件的方法是让进程一次性申请其需要的所有资源，都得到满足的时候，才分配给其资源，否则其必须等待。

[2] 不剥夺条件：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。当一个已经保持了某些资源的进程，再提出新的资源请求而不能立即得到满足时，必须释放它已经保持了的所有资源，待以后需要时再重新申请。

[3] 互斥条件：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。如果此时还有其它进程请求该资源，则请求者只能等待，直至占有该资源的进程用毕释放。值得注意的是，互斥条件不能去除掉，这是一类资源访问时必须具备的性质。

[4] 环路等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0, P1, P2, ..., Pn}中的 P0 正在等待一个 P1 占用的资源；P1 正在等待 P2 占用的资源，……，Pn 正在等待已被 P0 占用的资源。从而形成了资源的环路等待。破坏环路等待的条件是，系统将所有资源按类型进行线性排队，并赋予不同的序号，所有进程对资源的请求必须严格按照资源序号递增的次序提出，避免在所形成的资源分配图中出现环路。

5. 在避免死锁的方法中，引入了系统安全状态的概念，假设不对资源分配进行任何限制。下面几种说法，错误的是（ ）

- A. 系统安全状态可能转变为不安全状态
- B. 不安全状态时可能是已经死锁，可能还没有死锁
- C. 安全状态不可能已经死锁
- D. 不安全状态时资源分配图一定不可完全化

参考答案：【D】

首先，死锁了，系统一定处在不安全状态，处在不安全状态的系统不一定已经死锁。在使用银行家算法找安全序列的时候，我们说系统按照某个序列执行下去可能不安全，但是这个序列执行之初，还可以继续往下执行一段距离，并没有马上死锁。

6. 对任意 n 个作业，他们的平均带权周转时间最小值为（ ）

- A. 1
- B. 2
- C. 1/n
- D. n

参考答案：【A】

这个题目，我们可以在很多地方看到。做过王道操作系统的同学，也可以在对应的习题中遇到这个

题目。

7. 在程序运行前, 先将一个程序的所有模块以及所需要的库函数链接成一个完成的装配模块。这种链接方式称为 ( )

- A. 静态链接
- B. 装入时动态链接
- C. 可重定位链接
- D. 运行时动态链接

参考答案: 【A】

[1] 静态链接: 在程序运行之前, 先将各目标模块及它们所需的库函数, 链接成一个完整的装配模块, 以后不再拆开。我们把这种事先进行链接的方式称为静态链接方式。

[2] 装入时动态链接: 这是指将用户源程序编译后所得到的一组目标模块, 在装入内存时, 采用边装入边链接的链接方式。

[3] 运行时动态链接: 这是指对某些目标模块的链接, 是在程序执行中需要该(目标)模块时, 才对它进行的链接。

[4] 可重定位装入技术: 在多道程序环境下, 所得到的目标模块的起始地址通常是从 0 开始的, 程序中的其它地址也都是相对于起始地址计算的。此时应采用可重定位装入方式, 根据内存的当前情况, 将装入模块装入到内存的适当位置。

综上, 参考答案明显为 A。

8. 动态分区算法回收内存时, 如果回收区仅与空闲区链插入点前一个分区相邻接, 那么需要在空闲区链表中 ( )

- A. 增加一个新表项
- B. 修改前一个分区表项的大小
- C. 修改前一个分区表项的首地址
- D. 修改前一个分区表项的大小和首地址

参考答案: 【B】

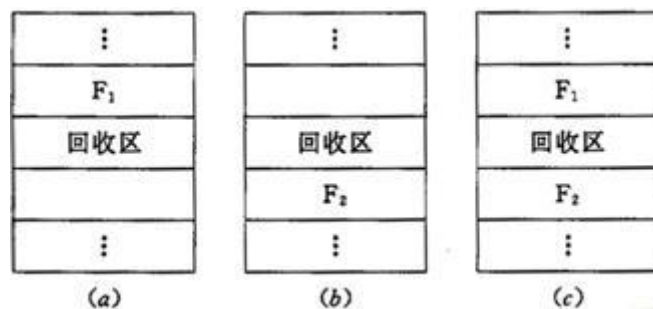
当进程运行完毕释放内存时, 系统根据回收区的首址, 从空闲区链中找到相应的插入点, 此时可能出现以下四种情况之一:

(1) 回收区与插入点的前一个分区 F1 相邻接, 这就是 B 选项, 只需要修改前一个分区表项的大小, 把回收区的大小加进去即可。见图(a)。

(2) 回收分区与插入点的后一分区 F2 相邻接, 见图(b)。此种方案我们认为, 也需要修改分区表项的首地址 (如果回收区处于低地址的话), 并修改表项的大小。

(3) 回收区同时与插入点的前、后两个分区邻接, 对应 D 答案, 见图(c)。

(4) 回收区既不与 F1 邻接, 也不与 F2 邻接。这需要新建表项, 对应于 A 答案。



9. 对换技术的主要作用是 ( )

- A. 将内存碎片合并为大的空闲空间
- B. 提高内存利用率
- C. 减少查找空闲分区的时间
- D. 提高外部设备利用率

参考答案:【B】

在多道程序环境下,一方面,在内存中的某些进程由于某事件尚未发生而被阻塞运行,但它却占用了大量的内存空间,甚至有时可能出现在内存中所有进程都被阻塞而迫使 CPU 停止下来等待的情况;另一方面,却又有着许多作业在外存上等待,因无内存而不能进入内存运行的情况。

显然这对系统资源是一种严重的浪费,且使系统吞吐量下降。为了解决这一问题,在系统中又增设了对换(也称交换)设施。所谓“对换”,是指把内存中暂时不能运行的进程或者暂时不用的程序和数据调出到外存上,以便腾出足够的内存空间,再把已具备运行条件的进程或进程所需要的程序和数据调入内存。对换是提高内存利用率的有效措施。

10. 关于基本分页的存储管理方式错误的说法 ( )

- A. 是一种离散的分配方式
- B. 物理块的大小应等于页面的大小
- C. 页面的大小应该是 2 的幂
- D. 不会产生内部碎片

参考答案:【D】

页的大小是固定的,所以有内部碎片。

11. 作为操作系统最重要的数据结构, ( ) 记录了操作系统所需的、用于描述进程当前情况及控制进程运行的全部信息。

- A. 索引结点
- B. 进程标示符
- C. 进程控制块
- D. 进程图

参考答案:【C】

12. ( ) 是指用于连接一个读进程和一个写进程以实现它们之间通信的一个共享文件。

- A. 共享存储器
- B. 管道
- C. 进程控制块

D. 索引结点

参考答案:【B】

13. 关于 I/O 控制方式, ( ) 控制方式使对 I/O 操作的组织和数据的传送最大限度地独立运行而无需处理机的干预。

A. I/O 通道

B. 中断驱动 I/O

C. 程序 I/O

D. 直接存储器访问 I/O

参考答案:【A】

14. 为了缓和 CPU 与 I/O 设备间速度不匹配的矛盾, 提高 CPU 和 I/O 设备的并行性, 现代操作系统关于 I/O 设备与处理机之间的数据交换几乎都用到了 ( )

A. 临界区

B. 缓冲区

C. 对换区

D. 工作集

参考答案:【B】

15. 为了提高操作系统自身的可适应性和可扩展性, 现代操作系统通过引入和实现 ( ) 的概念而实现了设备独立性。

A. 共享设备

B. 循环缓冲

C. 独占设备

D. 逻辑设备

参考答案:【D】

16. 关于文件的物理结构类型, ( ) 可以方便地支持和实现直接存取。

A. 顺序文件

B. 隐式链接文件

C. 显式链接文件

D. 目录文件

参考答案:【A】

17. 关于文件存储空间的管理方式, ( ) 需要使用空闲盘块号栈。

A. 空闲表法

B. 空闲链表法

C. 位示图法

D. 成组链接法

参考答案:【D】

18. 利用 ( ) 实现文件共享时, 只有文件主才拥有索引结点的指针, 而共享同一个文件的其他用户仅拥有对应文件的路径名。故而造成文件主删除共享文件后遗留悬空指针的问题。

A. 进程控制块

B. 共享存储器

- C. 索引结点  
D. 符号链

参考答案:【D】

19. 下面关于采用**抢占式**调度方式系统中系统调用完成时返回的描述, 不正确的是 ( )
- A. 只有当调用者进程仍具有最高优先级时, 才返回到调用者进程继续执行。  
B. 引发重新调度  
C. 无条件返回调用进程, 从调用位置的下一条指令处继续执行  
D. 有可能把调度者进程放入就绪队列而执行别的进程

参考答案:【C】

20. 设置和实现了内核支持线程的系统中处理机以 ( ) 为单位进行调度。
- A. 线程  
B. 程序  
C. 进程  
D. 函数

参考答案:【A】

## 二、计算题 (共 35 分)

### 1. 银行家算法 (12 分)

系统有三类资源{A,B,C},最大资源数分别是 10,5,7.当前 T0 时刻有 5 个进程 P0-P4.资源分配情况如下表:

进程与资源情况	最大需求			已获得的资源数			need		
	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	7	4	3
P1	3	2	2	3	0	2	0	2	0
P2	9	0	2	3	0	2	6	0	0
P3	2	2	2	2	1	1	0	1	1
P4	4	3	3	0	0	2	4	3	1

请问:

- (1). 当前时刻是否安全?

系统当前剩余的资源为  $work=(10,5,7)-(8,2,7)=(2,3,0)$ 。故而, 调度 P1 开始执行。

解析:

C P	Work			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P1	2	3	0	0	2	0	3	0	2	5	3	2	true
P3	5	3	2	0	1	1	2	1	1	7	4	3	true
P0	7	4	3	7	4	3	0	1	1	7	5	3	true
P2	7	5	3	6	0	0	3	0	2	10	5	5	true
P4	10	5	5	4	3	1	0	0	2	10	5	7	true

所以, 存在一个序列 P1、P3、P0、P2、P4 使得进程可以执行完毕。故而系统是安全的。【还有其他正确

序列，找出来即可】

(2). T1 时刻, P0 请求资源 (0,2,0)。请问按照银行家算法是否应该同意 P0 请求并分配资源? 我们先分配看看。若分配了 (0, 2, 0) 给 P0, 此时系统剩余的资源是 (2, 1, 0)。

进程与资源情况	最大需求			已获得的资源数			need		
	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	7	2	3
P1	3	2	2	3	0	2	0	2	0
P2	9	0	2	3	0	2	6	0	0
P3	2	2	2	2	1	1	0	1	1
P4	4	3	3	0	0	2	4	3	1

很显然，系统已经挂了。所以，不能给 P0 分配 (0,2,0) 的申请资源。

## 2. 计算物理地址 (8 分)

一个操作系统基于基本分段的存储管理方式, 支持的最大段长度是 64KB。一个进程的段表如下(十进制)。

段号	段长	段基址
0	512	80K
1	20K	50K
2	12K	81K
3	3K	95K
4	32K	10K

请问:

(1). 逻辑地址 0x47FD5 对应的物理地址是多少?

解析:

0x47FD5, 低 4 个 16 进制数表示段内地址, 高 1 个 16 进制数 4 刚好对应段号。可以看到, 该地址对应段号 4。该段的最大长度是 32K。即 0100 0000 0000 0000。划线部分就是最大段长。而该地址的段内地址是 0111 1111 1101 0101。很明显, 访问的地址大于段号, 所以越界了。

(2). 逻辑地址 0x003FF 对应的物理地址是多少?

解析: 段号是 0, 段内的 16 位 2 进制数是 0000 0011 1111 1111。段号 0 的段长只有 512。即 16 位段内地址最大是 0000 0000 1111 1111。(注意, 段内地址也是从 0 开始的。比如说, 段长是 1KB, 段内地址从 0~1023)。显然, 段内越界了。

## 3. 计算混合索引文件支持文件最大长度 (10 分)

某系统针对磁盘空间采用混合索引分配方式, 其文件索引结点设置为 12 个地址项, 即 iaddr(0)~iaddr(11)。其中, iaddr(0)~iaddr(9)存放直接地址即存放文件前 10 个索引盘块的盘块号。iaddr(10)存放一级间址即存放文件一级索引分配的盘块号; iaddr(11)存放二级间址即存放文件二级索引分配的索引盘块的盘块号。假

定系统盘块的大小是 8KB，且每一个盘块号占用 4B，问该系统支持的最大文件长度是多少字节？

解析：这个题目，无力吐槽了，出现过太多次了，好几年的期末题目上有，好几年的真题上也有类似的。而且算法特别固定。要是不会算这个题目，或者不太熟练的，加强一下啊！师弟师妹们！这是常考的知识点。

系统的盘块大小是 8KB，一个盘块号是 4B。那么一个盘块可以存放 2K 个项。

[1] iaddr(0)~iaddr(9)这 10 个直接地址支持文件的大小是 80KB

[2] iaddr(10)是一级间址，支持的文件大小是  $2K \times 8KB = 16MB$

[3] iaddr(11)是二级，间址，支持的文件大小是  $2K \times 2K \times 8KB = 32GB$

故而，支持的最大文件长度是  $32GB + 16MB + 80KB$

#### 4. 计算 FAT 表空间（5 分）

对于 200GB 的硬盘，若采用 FAT 文件系统格式且系统盘块大小设定为 4KB，问：

(1). FAT 表项长度应当选用 16 位还是 32 位？（采用 2 进制表示）

解析：FAT，即我们常说的文件分配表。文件存放在外存！内存是临时用来存放文件的。该硬盘共有盘块

号  $N = \frac{200GB}{4KB} = 50M$ 。而  $32M < 50M < 64M$ 。所以，表项的长度至少有 26 位。为了便于存取，取 32 位。

(2). FAT 表共占用多少字节的空间？

解析：文件分配表用 32 位表示，即 4B。因为有 50M 表项，所以 FAT 表用占用  $50M \times 4B = 200MB$  的空间。

### 三、论述题（25 分）

1. 由于在过去很长一段时间里，计算机系统非常昂贵，因而在设计操作系统时，有效性比方便性更重要。但是，近 20 年来随着微机的普及，所配置的操作系统则更加重视方便性。上述说法你是否赞同？请分别列举三中操作系统实现有效性的方法和三种实现方便性的方法，并根据对操作系统经典理论和操作系统发展趋势的认识，结合你使用操作系统的实际经验对比进行论述（不超过 200 字）（10 分）

答题思路：

[1] 虚拟处理机

[2] 虚拟存储器

[3] 操作系统提供了良好的用户接口，用户按需要输入命令，操作系统按命令去控制程序的执行；用户也可以在程序中调用操作系统的功能模块完成相应服务，而不必了解硬件的物理特性。

操作系统的有效性包含两个方面的含义，即提高系统资源利用率，提高系统吞吐量。

操作系统实现有效性的方法，随便举一些例子，呵呵...：

[1] 有效地管理和分配硬件、软件资源，提高系统工作效率。

[2] 操作系统扩充硬件的功能，使硬件的功能发挥得更好。

[3] 操作系统使用户合理共享资源，防止各用户间的相互干扰。

[4] 操作系统以文件形式管理软件资源，保证信息的安全和快速存取。

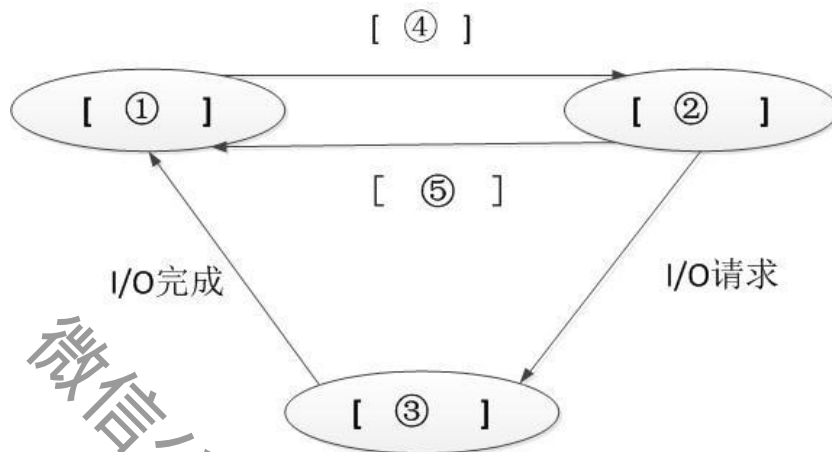
[5] 操作系统合理地组织计算机的工作流程，使用户程序能顺利完成。

当然，当中【2】、【4】、【5】是一个不错的组合，你觉得呢？



关于本题，有同学讨论了很长时间没有结果。我们认为，答案就在书中，但不拘泥于课本。当然，要是你还有更好的答案，请告诉我们。谢谢！

2. 分析图中给出的不完整的进程基本状态及其转换信息，在 5 个空格内填上对应的状态名称或者相应的转换条件。（10 分）



解析：I/O 的请求和完成时突破口。显然，状态 3 是阻塞。I/O 完成之后，阻塞进程被唤醒，进入就绪队列等待调度执行。所以，状态 1 是就绪。而正在执行的进程请求 I/O 操作的时候，才阻塞起来。状态 2 是运行。

4 显然是进程调度。5 可以理解成时间片用完。关于本题，也可以参考我们编写的《北京交通大学期末考试题目和解析》一书第二章进程管理的算法题和解答题的 16 题的解析。

3. 简述利用假脱机技术实现打印机共享的基本处理过程。（5 分）

解析：

当用户进程请求打印输出时，SPOOLing 系统同意为它打印输出，但并不真正立即把打印机分配给该用户进程，而只为它做两件事：① 由输出进程在输出井中为之申请一个空闲磁盘块区，并将要打印的数据送入其中；② 输出进程再为用户进程申请一张空白的用户请求打印表，并将用户的打印要求填入其中，再将该表挂到请求打印队列上。如果还有进程要求打印输出，系统仍可接受该请求，也同样为该进程做上述两件事。

如果打印机空闲，输出进程将从请求打印队列的队首取出一张请求打印表，根据表中的要求将要打印的数据，从输出井传送到内存缓冲区，再由打印机进行打印。打印完后，输出进程再查看请求打印队列中是否还有等待打印的请求表。若有，又取出队列中的第一张表，并根据其中的要求进行打印，如此下去，直至请求打印队列为空，输出进程才将自己阻塞起来。仅当下次再有打印请求时，输出进程才被唤醒。

【补充】SPOOLing 技术的特点：

- (1)提高了 I/O 速度。从对低速 I/O 设备进行的 I/O 操作变为对输入井或输出井的操作,如同脱机操作一样,提高了 I/O 速度,缓和了 CPU 与低速 I/O 设备速度不匹配的矛盾。
- (2)设备并没有分配给任何进程。在输入井或输出井中,分配给进程的是一存储区和建立一张 I/O 请求表。
- (3)实现了虚拟设备功能.多个进程同时使用一独享设备,而对每一进程而言,都认为自己独占这一设备。不过,

该设备是逻辑上的设备。

#### 4. 应用题（共 30 分）

##### 1. 基本多级反馈队列调度算法过程如下：

- a) 设置多个就绪队列，第一个队列优先级最高，其他就绪队列优先级依次降低。优先级越高的队列中进程执行时分配到的时间片就越小。
- b) 当新进程到达时首先放入第一队列末尾，按 FCFS 调度。轮到它执行时如果不能在一个时间片中执行完毕就放入第二队列尾。如此下去，直到降低到最后一个队列采用时间片轮转方式调度。
- c) 每次调度时，都选择优先级最高的非空队列进行调度。当正在执行一个进程时，如果有新进程到达一个优先级高的队列，则采用抢占式调度，立即执行新进程。

请回答如下问题：（15 分）

(1). 为什么说该算法能既满足短作业的需求，又兼顾长作业的需求？

- [1] 对于短作业，开始时像终端型作业一样，如果仅在第一队列中执行一个时间片即可完成，便可获得与终端型作业一样的响应时间。对于稍长的作业，通常也只需在第二队列和第三队列各执行一个时间片即可完成，其周转时间仍然较短。
- [2] 长批处理作业用户。对于长作业，它将依次在第 1, 2, ..., n 个队列中运行，然后再按轮转方式运行，用户不必担心其作业长期得不到处理。

(2). 在什么情况下长作业可能实现饿死现象？请问你如何改进？

解析：系统短作业或者比该长作业的优先级高的作业不断到来，长作业就会一直等不到处理机，产生饿死现象。这里是时间片+优先级，改成时间片+优先级+响应比。本题比较活，你也可以用自己的方法来做。

(3). 交互式作业往往只需 CPU 执行很短的时间就会进行 I/O 操作。对这类作业基本多级反馈队列调度算法存在什么问题？如何改进？

解答：作业很短，通常在一个时间片内就能执行完毕，算法像是回退到了

2. 在某商场开放入口自由进入，同时在唯一出口处设有收银台、由唯一的一个收银员负责收款业务。顾客在商场里自由购物后都须在出口收银台前排队和完成缴款手续后方可离开。请用记录型信号量机制描述顾客在商场中购物的全过程。

（提示：收银员亦须给出描述）（15 分）

解析：这里没有说商场最多能容纳多少人。若不考虑容量的问题，那么就简单了。

可以有这种写法：

顾客进入商场购物；

P(mutex);

结账;  
v(mutex);  
走人;

呵呵...但是这个太简单了。15 分呢，要是这么写，扣分估计有点严重。这也太短小精悍了。对吧？其实，若是告知最多可以容纳 N 个人，那么写法如下：

semaphore counter, mutex=1;用来设置互斥结账的信号量

int empty=N;

int guest=0;//初始假设超市没有人

收银员进程：

P(guest); //有人结账

收银

V(counter)//结账完毕了，顾客走了，后面排队的顾客可以结账了。

顾客进程:

P(empty);

/\*

来之前看看超市是不是能装得下人，若可以装下，就进入超市购物，若装不下，在超市外面排队等待。

\*/

进入店内购物;

V(guest);//通知收银员，我购物完毕了，打算结账走了，可以收银了。

P(counter);//结完帐，放掉收银员的信号量，让其他人结账

V(empty);//离开超市了，超市的顾客容量应该+1 了。

若是不作要求，来多少装多少，那么吧 empty 相关的操作删除即可。

【PV 操作到底怎么写？】关于这个问题，其实没有一个标准答案的。或许同样的题目，你会看很多的写法。到底哪一种是对的？其实只要是合理的，都是对的。

计算机/软件工程专业  
每个学校的  
考研真题/复试资料/考研经验  
考研资讯/报录比/分数线  
免费分享



微信 扫一扫  
关注微信公众号  
计算机与软件考研