2021 天勤计算机考研八套模拟卷 • 卷六

操作系统篇选择题答案解析

1．A。

这里的“账号”等价于“用户”。

I 正确很容易理解，支持多用户，肯定就是支持同时登录。

III 正确，多个客户端通过同一账号登录，这些个客户端其实运行的只是一个进程。Linux 支持多任务的系统，所以肯定是可以的。

知识点扩展：

Linux 是一个多用户、多任务的操作系统；考生应该了解单用户多任务和多用户多任务的概念。

（1）Linux 的单用户、多任务。

单用户、多任务；比如以 beinan 登录系统，进入系统后，要打开 gedit 来写文档，但在写文档的过程中，感觉少点音乐，所以又打开 xmms 来点音乐；当然听点音乐还不行，MSN 还得打开，想知道几个朋友现在正在做什么。这样，在用 beinan 用户登录时，执行了 gedit、xmms 以及 msn 等，当然还有输入法 fcitx；这样说来就有点简单了，一个 beinan 用户，为了完成工作，执行了几个任务；当然 beinan 这个用户，其他的人还能以远程登录过来，也能做其他的工作。

（2）Linux 的多用户、多任务。

有时可能是很多用户同时用同一个系统，但并非所有的用户都一定要做同一件事，所以这就有多用户、多任务。

举个例子，比如 LinuxSir.org 服务器，上面有 FTP 用户、系统管理员、web 用户、常规普通用户等，在同一时刻，可能有人正在访问论坛；有人可能在上传软件包管理子站，比如 luma 或 Yuking 在管理他们的主页系统和 FTP；在与此同时，可能还会有系统管理员在维护系统；浏览主页的用的是 nobody 用户，大家都用同一个，而上传软件包用的是 FTP 用户；管理员的对系统的维护或查看，可能用的是普通账号或超级权限 root 账号；不同用户所具有的权限也不同，要完成不同的任务就需要不同的用户，也可以说不同的用户，可能完成的工作也不一样。

值得注意的是，多用户、多任务并不是大家同时挤到一起在一台机器的键盘和显示器前来操作机器，多用户可能通过远程登录来进行，比如对服务器的远程控制，只要有用户权限任何人都是可以上去操作或访问的。

2．D。

Ⅰ错误，当发送方发送一个较小的数据包时，发送方将数据复制至消息队列，然后接收方从消息队列中拷走，这称为复制发送；如果数据包较大，发送方只是把指向数据包的指针和数据包大小发送给接收者，接收者通过指针访问数据包，这称为引用发送。显然引用发送比复制发送更复杂，但不需要复制数据，所以引用发送效率高。

Ⅱ错误，进程调度信息属于进程管理的内容，并非进程通信内容，这里还缺少一个实现消息队列互斥访问的互斥信号量。

Ⅲ错误，各个进程有自己的内存空间、数据栈等，所以只能使用进程间通信（Inter Process Communications，IPC），而不能直接共享信息。需要注意的是，这里的内存空间和进程通信中的共享的缓冲区是不一样。

Ⅳ正确，并发进程之间进行通信时，必定存在资源共享问题。进程通信归结为三大类：

（1）共享存储器系统，很明显共享了存储器资源。

（2）消息传递系统，共享了消息文件。

（3）管道通信，共享了管道文件。

3．D。

非抢占式（见下表）：

非抢占式进程调度的时间

进程名	到达时间	运行时间	开始时间	结束时间	周转时间
P1	0.0	9	0.0	9.0	9

P2	0.4	4	12.0	16.0	15.6
P3	1.0	1	9.0	10.0	9
P4	5.5	4	16.0	20.0	14.5
P5	7	2	10.0	12.0	5

平均周转时间为（9+15.6+9+14.5+5）/5=10.62。

抢占式（见下表）：

抢占式进程调度的时间

进程名	到达时间	运行时间	开始时间	结束时间	周转时间
P1	0.0	9	0.0	20.0	20
P2	0.4	4	0.4	5.4	5
P3	1.0	1	1.0	2.0	1
P4	5.5	4	5.5	11.5	6
P5	7	2	7.0	9.0	2

平均周转时间为（20+5+1+6+2）/5=6.8。

知识点回顾：

周转时间=结束时间-到达时间=等待时间+运行时间

区分：进程调度方式和进程调度算法。

进程调度方式指的是：

✧ 抢占方式。

✧ 非抢占方式。

进程调度算法指的是：

✧ FCFS。

✧ SJF。

✧ ……

4．D。

同步（直接相互制约关系）：一个进程到达了某些点后，除非另一个进程已经完成了某些操作，否则就不得不停下来等待这些操作的结束，这就是进程的同步，有了同步后进程间就可以相互合作了。用 P、V 操作实现进程同步，信号量的初值应根据具体情况来确定。若期望的消息尚未产生，则对应的初值应设为 0；若期望的消息已经存在，则信号量应设为一个非 0 的正整数。

互斥（间接相互制约关系）：多个进程都想使用一个临界资源，但是不能同时使用，于是只好一个进程用完了才能给其他进程用，这就是进程互斥。从某种意义上说，互斥是同步的一种特殊情况。一般互斥信号量的初始值都设置为 1，P 操作成功则将其改成 0，V 操作成功将其改成 1，所以互斥信号量的初值为 1。

综上所述，本题选 D。

5．B。

在 5 题图 a 中，系统中共有 R_1 类资源 2 个，R_2 类资源 3 个，在当前状态下仅有一个 R_2 类资源空闲。进程 P_2 占有一个 R_1 类资源及 1 个 R_2 类资源，并申请 1 个 R_2 类资源；进程 P_1 占有 1 个 R_1 类资源及 1 个 R_2 类资源，并申请 1 个 R_1 类资源及 1 个 R_2 类资源。因此，进程 P_2 是一个既不孤立又非阻塞的进程，消去进程 P_2 的资源请求边和资源分配边，便形成了下图 a 所示情况。

当进程 P_2 释放资源后，系统中有 2 个 R_2 类空闲资源，1 个 R_1 类空闲资源。因此，系统能满足进程 P_1 的资源申请，使得进程 P_1 成为一个既不孤立又非阻塞的进程，消去进程 P_1 的资源请求边和资源分配边，便形成了下图 b 所示情况。由死锁定理可知，5 题图 a 中的进程-资源图不会产生死锁。

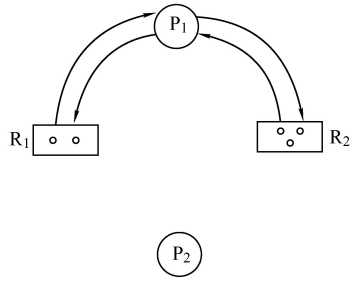


图 a　消去进程 P_2 的资源请求边和资源分配边

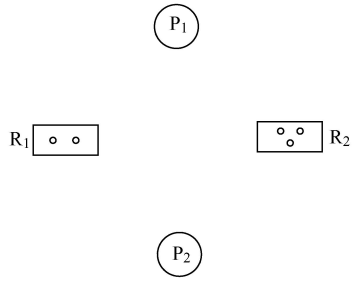


图 b 消去进程 P_1 的资源请求边和资源分配边

在 5 题图 b 中，系统中共有 R_1 类资源 1 个、R_2 类资源 2 个、R_3 类资源 2 个、R_4 类资源 1 个。在当前状态下仅有 1 个 R_3 资源空闲。进程 P_1 占有 1 个 R_2 资源，并申请 1 个 R_1 资源；进程 P_2 占有 1 个 R_1 资源及 1 个 R_3 资源，并申请 1 个 R_4 资源；进程 P_3 占有 1 个 R_4 资源及 1 个 R_2 类资源，并申请 1 个 R_3 类资源及 1 个 R_2 类资源。因此，该资源分配图中没有既不孤立又不阻塞的进程结点，即系统中的 3 个进程均无法向前推进，由死锁定理可知，5 题图 b 的进程-资源图会产生死锁。

6．B。

在段页式存储系统中，为了获取一条指令或数据，必须三次访问内存，第一次访问内存中的段表，从中取得页表地址；第二次访问内存中的页表，从中取得该页所在的物理块号，并将该块号与页内地址一起形成指令或数据的物理地址；第三次访问根据第二次访问所得的地址，真正取出指令或数据（但这是在访问地址正确的情况下）。

为了防止越界，在段页式存储系统中，配置了一个段表寄存器，其中存放段表始址和段表长度。在进行地址变换时，首先利用段号，将它与段长进行比较。若段号超出段表长度，表示越界了。

同样段表中的，页表大小这项也是为了预防地址越界的情况。

一般情况下，段号页号都是从 0 开始编址，这点从题目所给的图中也可得知。

该用户访问的 3 个地址中，地址一，段号检查通过，页号越界，3 不在页号 0～2 范围内，所以只访问内存 1 次。

地址二，段号检查未通过，8 不在段号 0～7 范围内，越界，所以访问内存 0 次。

地址三，段号检查通过，且页号也无越界，成功访问到数据，所以访问内存 3 次。

知识点回顾：

首先必须知道，系统为每个进程建立一张段表，而每个分段有一张页表。

段表项中至少包括段号、页表长度和页表起始地址，页表项中至少包括页号和块号。

段页式系统中从逻辑地址到物理地址的地址变换过程如下：

（1）从逻辑地址中取出前几位得到段号 S，中间几位得到页号 P，后几位得到页内偏移量 W。

（2）比较段号 S 和段表长度 M，若 S≥M，则产生越界中断，否则转至下一步。

（3）段表中对应段表项地址=段表起始地址 F+段号 S×段表项大小，取出该段表项内容的前几位得到对应页表的页表长度 C，后几位得到对应的页表的起始地址 d，若页号 P≥C，则产生越界中断，否则转至下一步。

（4）对应页表项地址=d+P×页表项大小，从该页表项内容得到物理块号 b。

（5）计算 E=b×页表大小 L+W。

（6）用得到的物理地址 E 去访问内存。

7．C。

CLOCK 页面淘汰算法的缺页情况（见下表）。

CLOCK 页面淘汰算法的缺页情况

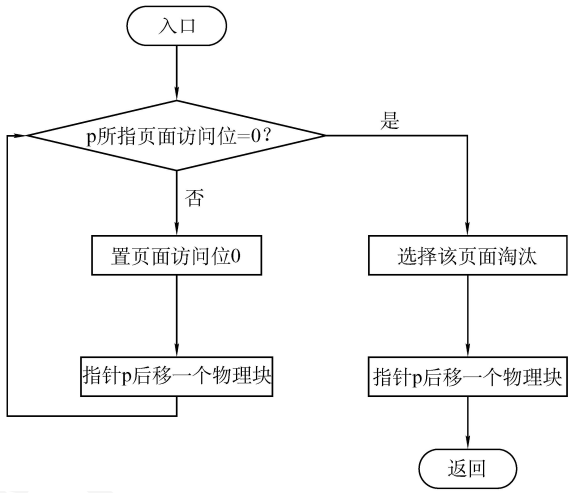
访问页面	物理块 0	物理块 1	物理块 2	说　明	缺页否
初始状态	→∧	∧	∧	p 指向块	
	0	0	0		
访问页 7	7	→∧	∧	调入页 7，块 0 访问位置 1，p 指针后移	√
	1	0	0		
访问页 0	7	0	→∧	调入页 0，块 1 访问位置 1，p 指针后移	√
	1	1	0		

访问页 1	→7	0	1	调入页 1，块 2 访问位置 1，p 指针后移	√
	1	1	1		
访问页 2	2	→0	1	p 指针循环后移（移动前修改访问位），找到块 0 的访问位为 0，替换进页 2，p 指针后移	√
	1	0	0		
访问页 0	2	→0	1	访问页 0 存在，修改其访问位，p 指针不移动	
	1	1	0		
访问页 3	→2	0	3	p 指针循环后移（移动前修改访问位），找到块 2 的访问位为 0，替换进页 3，p 指针后移	√
	1	0	1		
访问页 0	→2	0	3	访问页 0 存在，修改其访问位，p 指针不移动	
	1	1	1		
访问页 4	4	→0	3	p 指针循环后移（移动前修改访问位），找到块 0 的访问位为 0，替换进页 4，p 指针后移	√
	1	0	0		

知识点回顾：

CLOCK 算法是 LRU 算法的近似算法。CLOCK 算法流程图如下图所示。CLOCK 算法给每个页面设置一个访问位，标识该页最近有没有被访问过，再将内存中的所有页面通过一个指针链接成一个循环队列。

注意：若循环链表存在当前访问页时（访问页在某物理块中），直接将其访问位改为 1，指针 p 不移动（命中后指针不移动）；否则，若当前 p 指针指向页面的访问位为 0，则淘汰该页，调入新页，将其访问位改为 1，指针 p 移到下一个物理块；若当前 p 指针指向页面的访问位为 1，则将其访问位改为 0，并移动 p 指针到下一个物理块。



CLOCK 算法流程图

8．B。

文件控制块与文件一一对应（Ⅲ错误），创建文件时（create）建立对应的 FCB，而不是打开文件时创建的（Ⅱ错误）。人们把文件控制块的有序集合称为文件目录，即一个文件控制块就是一个文件目录项（Ⅰ正确）。在文件控制块中，通常含有 3 类信息，即基本信息、存取控制信息及使用信息（Ⅳ正确）。

所以Ⅰ，Ⅳ正确，Ⅱ，Ⅲ错误。错误的个数为 2，所以选 B。

9．C。

题目中暗含有时间顺序，“依次有 4 个等待着”，即最早来的等待着是要访问 37 号柱面的，所以Ⅰ正确。

考虑 50 号两个方向最近的柱面号请求，50−37=13 和 65−50=15，即拥有最短寻道时间的是 37 号柱面，所以Ⅱ也正确。

电梯调度算法，总是从磁头当前位置开始，沿磁头的移动方向（小磁道方向）去选择离当前磁头最近的那个柱面的请求，即 37。

循环扫描算法是电梯算法的改进版，但也是按当前移动方向（大磁道方向）去选择离当前磁头最近的那个柱面的请求，即 65。不同的是为了减少延迟，规定磁头单向移动，即只能有一个移动方向。

10．B。

这类题只要记住，时间换空间就是空间变大了，空间换时间就是时间缩短了，一般就不会做错了。总结如下：

时间换空间：虚拟存储技术、覆盖与交换技术等。

空间换时间：SPOOLing 技术、缓冲技术等。

解释如下：

各种虚拟存储技术都是时间换空间的技术，包括请求分页、请求分段、请求段页式，这些都是让访问时间增加了，但是扩充了主存的逻辑容量，使得大于主存容量的程序也可以得到执行。

如果空间换时间，则各类的缓冲区、缓冲池都是，本来需要在速度很慢的设备上 I/O 的，但是自从划分了些存储区域做缓冲，那么就可以减少访问时间。SPOOLing 技术需有高速大容量且可随机存取的外存支持，通过预输入及缓输出来减少 CPU 等待慢速设备的时间，这是典型的以空间换时间策略的实例。

所以本题选 B。