

V 操作系统试卷（2010 年）参考答案

一、名词解释题（每题 4 分，共 24 分）

1、进程控制块

答案：进程控制块是一个与动态过程相联系的数据结构，记载了进程的外部特性（名字、状态等）以及与其他进程的联系（通信关系），还记录了进程所拥有的各种资源。进程控制块是进程存在的标志。

2、原语

答案：原语通常由若干条指令所组成，用来实现某个特定的操作。通过一段不可分割的或不可中断的程序实现其功能。

3、临界区

答案：必须互斥执行的程序段称为相对于临界资源的临界区。

4、虚拟存储器

答案：虚拟存储技术是在主存和辅存之间，增加部分软件及必要的硬件支持，使主、辅之间的信息交换、程序的重定位、地址转换都能自动进行，从而主、辅存形成一个有机的整体，这种存储器的概念成为虚拟存储器。

5、缓冲区

答案：为了解决外部设备和内存或外部设备和 CPU 之间的数据传送速度不匹配的问题，在系统中引入缓冲区来暂存数据。

6、文件目录

答案：目录是文件系统层次结构的一个非终结节点，一个目录通常包含有许多目录项，每个目录项可能是一个文件或目录。

二、判断题（每题 1 分，共 6 分）

- 1、一个进程可以涉及一个或若干个程序的执行；反之，同一个程序只可以对应一个进程。（×）
- 2、信号量是只允许由 P/V 操作进行访问和修改的数据结构。（✓）
- 3、并发是指多个任务在多个处理机上正在同时运行，在微观上看，这些任务是在各自的物理处理机上分别运行。（×）
- 4、进程的同步与互斥可以发生在一个进程之中。（×）
- 5、中断方式的数据传送是在中断处理时由 CPU 控制完成的；DMA 方式则不经过 CPU，而是在 DMA 控制器的控制下完成的。（✓）
- 6、动态重定位便于程序浮动，其实现时采用的硬件机构是重定位寄存器和加法器。（✓）

三、简答题（每题 4 分，共 20 分）

- 1、实时系统和分时系统各有什么特点？有什么本质的区别？

答案：

- (1) 实时系统通常是一个专用系统，它的特点是响应时间快，快的程度依赖于实时系统的种类，如果是实时控制系统，则响应时间依赖于实时控制对象的需求，根据需要及时响应；如果是实时信息管理系统，其响应时间与分时系统的要求相似，只要使用者不抱怨响应慢即可，一般不超过 3 秒。实时系统对安全性要求较高，系统的安全可靠是实时系统的保障。
- (2) 分时系统亦称交互式系统，其特点是对用户的响应及时，当多个用户同时使用计算机时，都有独占的感觉。
- (3) 实时系统对响应时间的要求比分时系统更高，一般要求响应时间为妙级、毫秒级甚至微妙级。与分时系统相比，实时系统没有那么强的交互会话功能，通常不允许用户通过实时终端设备去编写新的程序或修改已有的程序。实时终端设备通常只是作为执行装置或询问装置，属专用系统。

2、进程与线程之间有何区别？

答案：

进程是操作系统中并发单元，也是能分得资源的最小单位。线程是在进程内部活动的并发单元，它只是进程行为的一条独立的执行路线，它能使用的资源仅限于它所在的进程范围之内，惟一能通过线程获得的资源就是使用处理机的时间片。有时也把线程称为轻量级进程。

3、简述段页式存储管理的基本原理。

答案：

段页式系统的基本原理是分段和分页原理的结合。即先将用户程序分为若干个段，再把每个段划分成若干页，并为每个段赋予一个段名。在段页式系统中，为了实现从逻辑地址到物理地址的转换，系统中需同时配置段表和页表。段表的内容还要包括页表起始地址和页表长度。

4、简述设备管理的主要功能。

答案：

- (1) 提供设备管理程序和进程管理系统的接口。当进程申请设备资源时，该接口将进程的请求转发给设备管理程序。
- (2) 进行设备分配。按照设备类型和相应的分配算法，把设备和其他相关的硬件分配给请求该设备的进程，并把未分配到所请求设备的进程放入等待队列。
- (3) 实现设备和设备、设备和 CPU 之间的并行操作。针对相应的硬件支持，采用不同的输入/输出控制方式。
- (4) 进行缓冲区管理。设备管理程序负责进行缓冲区分配、释放及有关的管理工作。

5、什么是文件的物理结构？常见的文件物理组织有几种？

答案：

- (1) 文件的物理结构是指文件记录在文件管理系统内部采用的、与物理存储介质的特性相适应的方式，是为系统使用的。
- (2) 顺序文件结构、随机文件结构、串联文件。

四、资源分配（共 5 分）

假设有三个进程 P1, P2 和 P3 并发工作。进程 P1 需用资源 S1 和 S2；进程 P2 需用资源 S3 和 S1；进程 P3 需用资源 S2 和 S3。请回答：

- (1) 若对资源分配不加限制，是否会发生死锁现象？请举例说明。（2 分）
- (2) 为保证进程的正确工作，可采用怎样的资源分配策略？为什么？（3 分）

答案：

- (1) 可能会发生死锁。例如：进程 P1, P2 和 P3 分别获得资源 S1, S3 和 S2 后，再继续申请资源时都要等待，即发生循环等待。(或进程在等待新源时均不释放已占资源)
- (2) 可有几种答案：
 - A. 采用静态分配：由于执行前已获得所需的全部资源，故不会出现占有资源又等待别的资源的现象(或不会出现循环等待资源现象)。
 - B. 采用按序分配：不会出现循环等待资源现象。
 - C. 采用银行家算法：因为在分配时，保证了系统处于安全状态。

五、进程同步（共 15 分）

设有三个并发进程：进程 Reader 负责从输入设备读入信息并传送给进程 Handler，进程 Handler 将信息加工并传送给进程 Printer，进程 Printer 将进行打印输出。其中，三个进程共享同一个缓冲区，且缓冲区大小为 K。请使用 P/V 操作，写出正确的并发程序。请注意以下说明：

- (1) 所使用的信号量：同步信号量或(和)互斥信号量，并说明信号量的名称、含义及初值。（3 分）
- (2) 分别写出进程 Reader、Handler、Printer 及主进程的代码。（12 分）

答案：

- (1) 同步信号量：empty，表示空缓冲块数目，初值为 k；full，表示可进行信息加工的缓冲块数目，初值为 0；ok，表示可进行信息输出的缓冲块数目，初值为 0。
互斥信号量：mutex，用于实现临界区互斥访问，初值为 1。
- (2) 代码如下：

```
var
empty, full, ok, mutex: semaphore;
inR, outR, inP, outP: integer;
buffer: array 0..k-1 of item;
procedure Reader
begin
while true do
begin
    输入数据 data1;
    P(empty); //减少一个空缓冲块
    P(mutex); //占用缓冲区
    buffer(inR) := data1; //信息放入缓冲块
    inR := (inR+1) mod (k); //指针指向下一个缓冲块
    V(mutex); //释放缓冲区
```

```
V(full); //增加一个可以加工的缓冲块
end
end
procedure Handler
begin
while true do
begin
P(full); //减少一个可以加工的缓冲块
P(mutex); //占用缓冲区
data2 := buffer(outR); //取出将要加工的信息
outR := (outR+1) mod (k); //指针指向下一个缓冲块
V(mutex); //释放缓冲区
对 data2 加工;
P(mutex); //占用缓冲区
buffer(inP) := data2; //将加工后的信息放入缓冲块
inP := (inP+1) mod (k); //指针指向下一个缓冲块
V(mutex); //释放缓冲区
V(ok); //增加一个可以输出的缓冲块
end
end
procedure Printer
begin
while true do
begin
P(ok); //减少一个可以输出的缓冲块
P(mutex); //占用缓冲区
data3 := buffer(outP); //取出将要输出的信息
outP := (outP+1) mod (k); //指针指向下一个缓冲块
V(mutex); //释放缓冲区
V(empty); //增加一个空缓冲块
打印 data3;
end
end
begin
seminitial(empty.v,k; full.v,0; ok.v, 0; mutex.v,1);
inR:=0; outR:=0;
inP:=0; outP:=0;
cobegin
Printer;
Handler;
Printer;
coend
end
```

六、银行家算法（10 分）

假设有 A、B、C、D 四类资源，在银行家算法中，若出现如下资源分配情况：

Process	Allocation	Need	Available
P0	0032	0012	1623
P1	1000	1750	
P2	1354	2356	
P3	0332	0652	
P4	0014	0656	

请问：

- (1) 当前状态是否是安全的？若是，给出一个安全序列。（5 分）
- (2) 如果进程 P2 提出安全请求 $Request[2]=(1,2,2,2)$ ，系统能否将资源分配给它？说明原因。（5 分）

答案：

- (1) 当前状态是安全状态。

令 $Work = Available = (1, 6, 2, 3)$ ，运行安全性检测算法：

- 1) $Finish[0]=false$ 并且 $Need[0]=(0, 0, 1, 2) < Work$ ，则 $Work = Work + Allocation[0] = (1, 6, 2, 3) + (0, 0, 3, 2) = (1, 6, 5, 5)$ ； $Finish[0] = true$ ；
 - 2) $Finish[3]=false$ 并且 $Need[3] = (0, 6, 5, 2) < Work$ ，则 $Work = Work + Allocation[3] = (1, 6, 5, 5) + (0, 3, 3, 2) = (1, 9, 8, 7)$ ； $Finish[3] = true$ ；
 - 3) $Finish[4]=false$ 并且 $Need[4] = (0, 6, 5, 6) < Work$ ，则 $Work = Work + Allocation[4] = (1, 9, 8, 7) + (0, 0, 1, 4) = (1, 9, 9, 11)$ ； $Finish[4] = true$ ；
 - 4) $Finish[1]=false$ 并且 $Need[1] = (1, 7, 5, 0) < Work$ ，则 $Work = Work + Allocation[1] = (1, 9, 9, 11) + (1, 0, 0, 0) = (2, 9, 9, 11)$ ； $Finish[1] = true$ ；
 - 5) $Finish[2]=false$ 并且 $Need[2] = (2, 3, 5, 6) < Work$ ，则 $Work = Work + Allocation[2] = (2, 9, 9, 11) + (1, 3, 5, 4) = (3, 12, 14, 15)$ ； $Finish[2] = true$ ；
- 因此，可以找到一个安全进程序列 $\langle p0, p3, p4, p1, p2 \rangle$ ，它使对于所有 $0 \leq i \leq 4$ ， $Finish[i]=true$ ，因而系统当前处于安全状态。

- (2) 运行银行家算法，由于 $Request[2] = (1, 2, 2, 2) \&\& Need[2] = (2, 3, 5, 6)$ ，因而请求合法。进一步， $Request[2] = (1, 2, 2, 2) \&\& Available = (1, 6, 2, 3)$ ，故该请求是可以满足的。假设将资源分配给 p2，则系统状态变为：

Process	Allocation	Need	Available
P0	0032	0012	0401
P1	1000	1750	
P2	2576	1134	
P3	0332	0652	
P4	0014	0656	

运行安全性检测算法， $Work=Available=(0, 4, 0, 1)$ ， $Finish[i]=false$ ，此时所有 $Need[i] \&\& Work[i]$ 均不成立，结果 $Finish[i]$ 均为 false，不存在安全进程序列，系统处于不安全状态。系统将取消资源分配并恢复原来状态，进程 p2 等待。

七、存储管理（20 分）

- 1、假定某页式存储管理系统，主存为 64KB，分成 16 块，块号为 0, 1, 2, …, 15。假设某作业有 4 页，其页号为 0, 1, 2, 3，被分别装入主存的 2, 4, 1, 6 块。请问：（共 8 分）

- (1) 该作业的总长度为多少字节？（按十进制）（2 分）

- (2) 写出该作业每一页在主存中的起始地址。(2 分)
(3) 若给出逻辑地址[0,100], [1,50], [2,0], [3,60], 请计算出相应的内存地址。
(4 分)

答案:

(1) 每块的长度=64KB/16=4KB, 因为块的大小与页面的大小相等, 所以每页为4KB。因此, 作业的总长度为 4KB*4=16KB。

(2) 因为页号为 0, 1, 2, 3, 被分别装入主存的 2, 4, 1, 6 块中, 即块表为:

页号	块号
0	2
1	4
2	1
3	6

所以该作业的:

第 0 页在主存中的起始地址为 $4K \times 2 = 8K$;

第 1 页在主存中的起始地址为 $4K \times 4 = 16K$;

第 2 页在主存中的起始地址为 $4K \times 1 = 4K$;

第 3 页在主存中的起始地址为 $4K \times 6 = 24K$ 。

(3) 逻辑地址[0,100]的内存地址为 $4K \times 2 + 100 = 8192 + 100 = 8292$

逻辑地址[1,50]的内存地址为 $4K \times 4 + 50 = 16384 + 50 = 16434$

逻辑地址[2,0]的内存地址为 $4K \times 1 + 0 = 4096 + 0 = 4096$

逻辑地址[3,60]的内存地址为 $4K \times 6 + 60 = 24576 + 60 = 24636$

2、在一个请求页式存储管理系统中, 进程 P 共有 5 页, 访问串是 4、3、2、1、4、3、5、4、3、2、1、5, 且开始执行时主存中没有页面。当分配给该进程的物理页面数为 3 和 4 时, 试用如下页面淘汰算法, 计算访问过程中发生的缺页率, 并比较所得结果。(12 分)

- (1) FIFO
(2) LRU
(3) OPT

答案:

(1) 根据所提供的访问次序, 采用 FIFO 淘汰算法的页面置换情况如下:

访 问 次序	4	3	2	1	4	3	5	4	3	2	1	5
物 理 页 1	4	3	2	1	4	3	5	5	5	2	1	
物 理 页 2		4	3	2	1	4	3	3	3	5	2	
物 理 页 3			4	3	2	1	4	4	4	3	5	
缺页	缺	缺	缺	缺	缺	缺	缺			缺	缺	
缺页率为 9/12。												
访 问	4	3	2	1	4	3	5	4	3	2	1	5

次序												
物 理	4	3	2	1	1	1	5	4	3	2	1	5
页 1												
物 理		4	3	2	2	2	1	5	4	3	2	1
页 2												
物 理			4	3	3	3	2	1	5	4	3	2
页 3												
物 理				4	4	4	3	2	1	5	4	3
页 4												
缺页	缺	缺	缺	缺			缺	缺	缺	缺	缺	缺

缺页率为 10/12。

由结果可以看出，对于 FIFO 页面淘汰算法，增加分配给进程的物理页数，缺页率反而上升。因此，FIFO 页面淘汰算法有异常现象。

（2）根据所给访问串，采用 LRU 淘汰算法的页面置换情况如下：

访 问	4	3	2	1	4	3	5	4	3	2	1	5
串												
物 理	4	3	2	1	4	3	5	4	3	2	1	5
页 1												
物 理		4	3	2	1	4	3	5	4	3	2	1
页 2												
物 理			4	3	2	1	4	3	5	4	3	2
页 3												
缺页	缺	缺	缺	缺	缺	缺	缺			缺	缺	缺

缺页率为 10/12。

访 问	4	3	2	1	4	3	5	4	3	2	1	5
串												
物 理	4	3	2	1	4	3	5	4	3	2	1	5
页 1												
物 理		4	3	2	1	4	3	5	4	3	2	1
页 2												
物 理			4	3	2	1	4	3	5	4	3	2
页 3												
物 理				4	3	2	1	1	1	5	4	3
页 4												
缺页	缺	缺	缺	缺			缺			缺	缺	缺

缺页率为 8/12。

由结果可以看出，对于 LRU 页面淘汰算法，增加分配给进程的物理页数，缺页率降低。

（3） 根据所给访问串，采用 OPT 淘汰算法的页面置换情况如下：

4、3、2、1、4、3、5、4、3、2、1、5

访 问 串	4	3	2	1	4	3	5	4	3	2	1	5
物 理 页 1	4	4	4	4	4	4	4	4	4	2	2	2
物 理 页 2		3	3	3	3	3	3	3	3	3	1	1
物 理 页 3			2	1	1	1	5	5	5	5	5	5
缺页	缺	缺	缺	缺			缺			缺	缺	

缺页率为 7/12。

访 问 串	4	3	2	1	4	3	5	4	3	2	1	5
物 理 页 1	4	4	4	4	4	4	4	4	4	4	1	1
物 理 页 2		3	3	3	3	3	3	3	3	3	3	3
物 理 页 3			2	2	2	2	2	2	2	2	2	2
物 理 页 4				1	1	1	5	5	5	5	5	5
缺页	缺	缺	缺	缺	缺		缺				缺	

缺页率为 6/12。

由结果可以看出，对于 OPT 页面淘汰算法，增加分配给进程的物理页数，缺页率下降。OPT 页面淘汰算法仅是一种理论算法，因为它根据未来页面的走向决定淘汰哪一页，而在实际执行时无法准确地知道未来行为。所以，该算法不作为实用算法，仅用于算法的比较和评价。