

B 卷答案

一、 名词解释

(1) 中断是指当某个事件发生时，系统终止运行现行的程序，引出处理程序对该事件进行处理，处理完毕后返回断点继续执行的过程。

(2) 进程是一个具有独立功能的程序关于某个数据集合的一次可以并发执行的运行活动，是操作系统进行资源分配和调度的一个独立单位。

(3) 虚拟内存构建与内存架构之上，通过缓存与 CPU 打交道，通过磁盘与用户程序打交道。它给外界的现象就是一个速度很快、容量很大的主存。由于这个主存并不是物理上真实存在的主存，因而称之为虚拟内存。

(4) 内存管理单元是将虚拟地址转换物理地址的一种硬件设备。

(5) 在页式存储管理系统中，如果某一个或某些页面不停地从内存调入外存，又从外存调入内存，则称为内存抖动，也可以称为系统抖动。

二、 简答题

(1) 为了扩大用户能够使用的主存空间，而又不真正花钱购买实际的主存，引入了虚拟内存的概念。虚拟内存从根本上说是将容量有限的主存扩充到容量巨大的外存上，让外存成为主存的一部分。同时，虚拟内存还提供接近缓存的访问速度。虚拟内存的最大容量取决于计算机体系结构里的寻址位数，取指等于 $2^{\text{寻址位数}}$ 。当寻址位数为 32 时，虚拟内存的最大容量为 4GB。

(2) 地址空间是逻辑地址的集合，即所谓的虚拟地址空间。存储空间是物理地址的集合，即计算机系统实际的内存空间。

(3) 在文件系统中，文件目录记录文件的管理信息，又称为文件控制块，或者文件说明信息，文件系统把同一卷上的若干文件的文件目录组成一个独立的文件，这个文件全部由文件目录组成，称为目录文件。文件目录用于对单个文件的控制，它记录文件的名称、文件长度、文件存放在外存上的物理地址，以及文件属性和文件建立时间、日期等信息。目录文件是全部文件目录组成的文件，用于整个文件系统的管理。

(4) 作业调度属于高级调度，而进程调度属于低级调度。作业调度是根据系统内资源的使用情况，从后备作业队列中选择一道作业进入系统，并创建相应的进程，分配必要的系统资源，使其处于“就绪”状态。进程调度是根据 CPU 的使用情况及时地把 CPU 分配给一个“就绪”的进程，使其从“就绪”状态变为“运行”状态。

(5) 所谓线程，从操作系统的管理角度看，就是指“进程的一个可调度实体”，是处理机调度的基本单位；从编程逻辑角度看，线程是指“程序内部的一个单一的顺序控制流”。线程是进程的一个组成部分，每个进程在创建时通常只有一个线程，由这个线程再创建其他线程。通常一个进程有若干个线程，至少会有一个线程。进程和线程是构造操作系统的两个基本元素，两者之间的主要区别是：

1) 调度方面：线程作为调度分配的基本单元

2) 并发性方面：进程之间可以并发执行

3) 拥有资源方面：进程是拥有资源的基本单位，线程除少量必不可少的资源外，基本上不拥有资源，但它可以访问其隶属进程的资源

4) 系统开销：进程间开销切换时要设计进程环境的切换，开销比较大。而线程切换只需保存和设置少量的寄存器内容。因此进程间切换的系统开销远大于线程间切换的系统开销。

三、 1.采用 FCFS 调度算法时，各任务在系统中的执行情况如下表所示：

执行次序	运行时间	优先数	等待时间	周转时间
A	10	3	0	10
B	6	5	10	16
C	2	2	16	18
D	4	1	18	22
E	8	4	22	30

所以，进程的平均周转时间为： $T=(10+16+18+22+30)/5=19.2\text{min}$

2.采用时间片轮转算法时，各任务的执行情况是：(A, B, C, D, E), (A, B, D, E), (A, B, E), (A, E), (A)。
 设 A~E 5 个进程的周转时间依次为 $T_1\sim T_5$ ，显然有 $T_1=30\text{min}$, $T_2=22\text{min}$, $T_3=6\text{min}$, $T_4=16\text{min}$, $T_5=28\text{min}$ ，所以，进程的平均周转时间为 $T=(30+22+6+16+28)/5=20.4\text{min}$

四、 (1) 利用银行家算法对此时刻的资源分配情况进行分析，可得下表

	Work				Need				Allocation				Work+Allocation				Finish
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	
P0	1	6	2	2	0	0	1	2	0	0	3	2	1	6	5	4	True
P3	1	6	5	4	0	6	5	2	0	3	3	2	1	9	8	6	True
P4	1	9	8	6	1	7	5	0	0	0	1	4	1	9	9	10	True
P1	1	9	9	10	0	6	5	6	1	0	0	0	2	9	9	10	True
P2	2	9	9	10	2	3	5	6	1	3	5	4	3	12	14	14	True

从上述分析可以看出，此时存在一个安全序列{P0、P3、P4、P1、P2}，故该状态是安全的。

(2) P2 提出请求 Request2(1,2,2,2)。按银行家算法进行检查：

Request2 (1,2,2,2) <= Need (2,3,5,6)

Request2 (1,2,2,2) <= Available(1,6,2,2)

试分配并修改相应数据结构后，发现这是可用资源 Available (0,4,0, 0) 已不能满足任何进程的需求，此时系统不能将资源分配给 P2。

五、 1.主存分为 256 块，说明一共能存放 256 个页面。每块大小为 4KB，说明页内地址所占位数为 12 位($4K = 2^{12}$)。而十进制地址 9016 可以表示为：10 0011 0011 1000，也就是页面号为 2，页内地址为 001100111000（十进制为 824）。从页表可知页面 2 存放在物理内存的块 32。因此，最后的物理地址为 $32*4KB + 824 = 128KB + 824$ 。

2.若给定逻辑地址为 12300，按 (1) 中同样的方法处理可得其页号为 3（12300 表示为 11 0000 0000 1100）。从页表可知该页未装入主存，因而产生缺页中断。随后中断处理程序将该页转入主存，然后进行地址转换。

六、 1.采用 FIFO 页面置换算法时，该作业运行时缺页情况如下表所示：

时刻	1	2	3	4	5	6	7	8	9	10	11	12
访问页面	4	3	2	1	4	3	5	4	3	2	1	5
内存页面	4	3 4	2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	5 1 2 3	4 5 1 2	3 4 5 1	2 3 4 5	1 2 3 4	5 1 2 3
缺页	4	3	2	1			5	4	3	2	1	5

从表中可以看出，缺页中断次数为 10，缺页率为 $f=10/12 * 100\% = 83\%$

2.采用 LRU 页面置换算法时，该作业运行时缺页情况如下表所示：

时刻	1	2	3	4	5	6	7	8	9	10	11	12
访问页面	4	3	2	1	4	3	5	4	3	2	1	5
内存页面	4	3 4	2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 5 3 4	1 5 3 4	1 5 3 4	2 5 3 4	2 1 3 4	2 1 3 5
缺页	4	3	2	1			5			2	1	5

从表中可以看出，缺页中断次数为 8，缺页率为 $f=8/12 * 100\% = 67\%$

七、 1.每个 block 可以容纳 $2KB/4B = 512$ 个指针

直接索引可以容纳 $8 * 2KB$

一级间接索引可以容纳 $512 * 2KB$

二级间接索引可以容纳 $512 * 512 * 2KB$

所以，最大文件为 $(8 + 512 + 512 * 512) * 2KB = 525328KB = 513MB + 16KB$

2. $128MB = 128 * 2^{10} KB$, $128 * 2^{10} / 2 = 65536$ 个 block。

使用直接索引，剩余 $65536 - 8 = 65528$ 个 block，本次装载 8 个 block，已占用 8 个 block；

使用一级索引，剩余 $65528 - 512 = 65016$ 个 block，本次装载 512 个 block，已占用 $8 + 1$ （一级间接索引块）+ 512 个 block

使用二级索引，剩余 $65016 - 65016 = 0$ 个 block，本次装载 65016 个 block，已占用 $8 + 1$ （一级间接索引块）+ $512 + 1$ （二级间接索引块）+ 126 （二级中的一级间接索引块）+ $126 * 512 + 1$ （二级中的一级间接索引块， $65016 = 126 * 512 + 504$ ）+ $504 = 131330KB = 128MB + 129 * 2KB$

八、参考答案

(1)用一个具体的进程并发时序说明此方案是读者优先的方案。

用 $rb(i)$ 表示第 i 个读者提出读数据，用 $re(i)$ 表示第 i 个读者结束读数据，

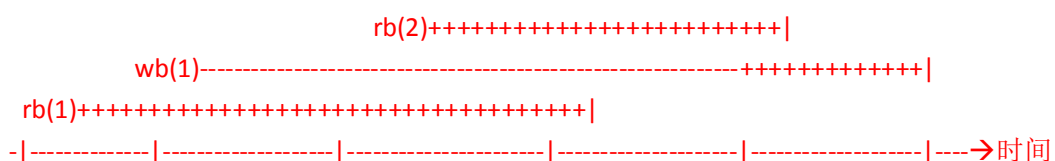
用 $wb(i)$ 表示第 i 个写者提出写数据，用 $we(i)$ 表示第 i 个读者结束读数据，

考察下面的进程并发时序：

$rb(1)...wb(1)...rb(2).....re(2).....$

即第一个读者在提出读数据且未结束读操作期间，有一个写者提出写数据和另一个读者提出读数据。那么，第 2 个读者将比第一个写者先完成操作，这说明后来的读者有较优先的权利，即所谓读者优先。

或用下图说明：



+++ 在操作数据
 ---- 在等待操作数据

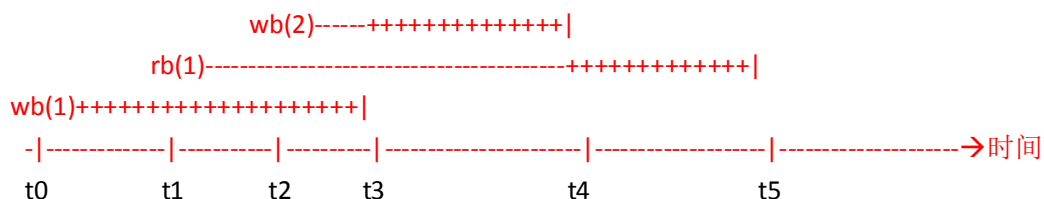
(2)将此方案改为读者优先的方案。

设置信号量

x=1,y=1,z=1,rsem=1,wsem=1;

<pre> /* program reader_and_writer */ int readcount,writecount; semaphore x=1,y=1,z=1,rsem=1,wsem=1; void reader() { while(true) { P(z); P(rsem); P(x); readcount++; if (readcount==1) P(wsem); V(x); V(rsem); V(z); READUNIT(); P(x); readcount--; if (readcount==0) V(wsem); V(x); } } </pre>	<pre> void writer() { while(true) { P(y); writecount++; if (writecount==1) P(rsem); V(y); P(wsem); WRITEUNIT(); V(wsem); P(y); writecount--; if (writecount==0) V(rsem); V(y); } } void main() { </pre>
--	---

用下图说明：



+++ 在操作数据
 ---- 在等待操作数据

第一个写者在 t0 时提出并获得写权利，直到 t3 结束写操作，这期间 writecount=1,rsem=0。

第一个读者在 t1 时提出读操作，这时 readcount=1，在 p(rsem)时被阻塞在 rsem 上(当时 rsem=0)；

第二个写者在 t2 时提出，但在 p(wsem)时被阻塞在 wsem 上(当时 wsem=0)，直到 t3，因为第一个写者结束写操作后,执行 v(wsem)时被唤醒，并获得写数据的权利。

第一个读者在 t4 时，因为第二个写者结束了写操作，这时 writecount=0，因而执行 v(rsem)

时唤醒了第一个读者，之后完成读操作；