

2012 ~2013 学年第 1 学期期末考试试卷参考答案

《操作系统原理 1》(A 卷 共 5 页)

(考试时间: 2013 年 1 月 8 日)

题号	一	二	三	四	成绩	核分人签字
得分	10	40	30	20		

一、单项选择题(10 道小题, 每小题 1 分, 共 10 分。请将答案填在题后的表格中, 写在其他位置的答案无效!)

- 中断扫描机构是 () 扫描一次中断寄存器。
A、每隔一个时间片 B、每条指令执行周期内最后时刻
C、每当进程释放 CPU D、每产生一次中断
- 若一个信号量的初值为 3, 经过多次 P、V 操作之后当前值为 -1, 则表示等待进入临界区的进程数为 ()
A、1 B、2 C、3 D、4
- 考虑页面置换算法, 系统有 m 个物理块供调度, 初始时全空, 页面引用串长度为 p, 包含了 n 个不同的页号, 无论用什么算法, 缺页次数不会少于 ()
A、m B、p C、n D、 $\min(m, n)$
- 操作系统为了管理文件, 设计了文件控制块 (FCB), 文件控制块的建立是在调用 () 时。
A、create() B、open() C、read() D、write()
- 总体上说, “按需调页” (Demand-Paging) 是个很好的虚拟内存管理策略。但是, 有些程序设计技术并不适合于这种环境。例如 ()
A、堆栈 B、线性搜索 C、矢量运算 D、二分法搜索
- 能否使用管程, 主要取决于:
A、程序员的编程技巧 B、编程语言的编译器支持
C、操作系统是否支持线程 D、是否有相应硬件的支持
- 弹出式线程的优点在于:
A、没有历史, 创建迅速 B、安全性高
C、执行效率高 D、不需要操作系统内核支持线程
- 把进程地址空间中使用的逻辑地址变成内存中物理地址的过程称为:
A、重定位 B、物理化 C、逻辑化 D、加载
- 设备分配问题中, 算法实现时, 同样要考虑安全性问题, 防止在多个进程进行设备请求时, 因相互等待对方释放所占设备所造成的 () 现象。
A、瓶颈 B、碎片 C、系统抖动 D、死锁
- 主存与辅存间频繁的页面置换现象被称为 ()。
A、请求调页 B、碎片整理 C、系统抖动 D、输入输出

选择题答题区

1	2	3	4	5	6	7	8	9	10
B	A	C	B	D	B	A	A	D	C

二、简答题（8 道小题，每小题 5 分， 共 40 分）

1、什么是系统调用（System call 或称为 System API）？简述一下系统调用的使用方法和执行过程。

答：系统调用是由操作系统内核提供的服务例程。（2 分）

以 C 语言函数或中断的方式使用，例如 fork()或 int 0x80。（2 分）

执行过程：保护现场、陷入内核态、执行服务例程、返回用户态，恢复现场。（2 分）

2、在 Shell 中执行一个命令，从输入命令开始到命令结束，至少可能会涉及到哪些系统调用，这些系统调用的功能分别是什么？

答：fork() 分裂产生子进程（2 分）

exec() 子进程替换代码段（2 分）

以下系统调用写出任意一个即可（1 分）

wait() 父进程等待子进程结束

waitpid() 父进程等待特定 PID 的子进程结束

exit() 子进程退出，唤醒父进程

3、缺页中断产生后，被中断的进程应该转入什么运行状态？讨论一下缺页中断的执行过程，并说明中断处理完毕后返回被中断的进程时应该执行哪条语句。

答：转入阻塞态（1 分）

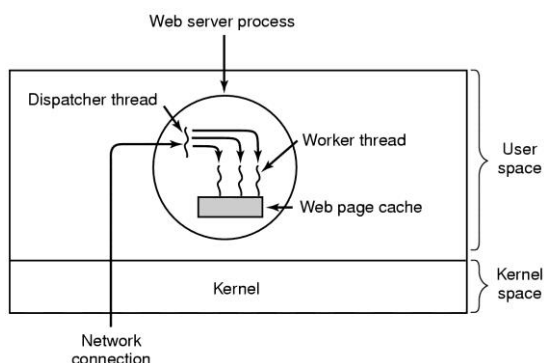
保护现场、陷入内核态、如果内存空间不足则选择淘汰、调需要的页面进内存（1 分）、更新页表和快表（1 分）、返回用户态，恢复现场、（1 分）重新执行被中断的语句。

执行被中断的语句（1 分）

4、如果通过资源图发现存在死锁，通常的方法是结束环上的一个进程来破坏死锁。请说明选择这个进程的一般原则。

答：选择最年轻的进程，因为消耗资源有限，恢复较容易。（5 分）

5、在 Web Server 中使用线程，可以提高对客户请求的响应效率。请简述图中 web page cache 的作用。



答：线程共享进程资源，所有的线程都可以访问进程的 web page cache（1 分），cache 在内存中（1 分），被访问过的页面存放在 cache 中，当任何线程再次需要这个页面时就可以从 cache 中得到，而不需要再次读取硬盘。（3 分）

- 6、请讨论一下页面置换算法中工作集（Working Set）置换算法的工作原理。
答：进程设置一个虚拟时钟，执行一个时钟周期就加 1，不执行就不增加。（1 分）
每个页面被访问时，记录最后访问的虚拟时间，R 位置 1。R 位定期清除。（1 分）
如果 R=1, 则保留，将当前时间记录下来。（1 分）
如果 R=0 对比当前虚拟时间与页面最后访问时间差 age 与阈值 τ ，如 $age > \tau$ 则淘汰（1 分）。如 $age \leq \tau$ ，则记录其访问时间，必要时淘汰其中最旧的。（1 分）
- 7、请讨论一下使用 XCHG 语句实现加锁的方法和原理。
答：首先寄存器设为 1（1 分），再用 XCHG 将寄存器与内存变量 lock 交换（2 分），然后检查此时寄存器（即原来 lock 内）的值，是 1 就等待，是 0 就通过（1 分）。
关键在 XCHG 保证交换的原子性。（1 分）
- 8、在内存管理的方法中，分段式管理比分页式管理有什么优势？段页式与其他方式相比有什么好处？
答：分段比分页更有逻辑性，将同类的或相关的内容放在一个段内，这样不会由于页面置换算法选择不当而形成“抖动”现象。（1 分）
同类内容划分在一个段内，可以实现段的保护，如代码段设置为只读，数据段设置为读写。（1 分）
公共代码段可以通过映射共享到多个进程。（1 分）
段页式既按照相关性划分段，继承了分段的优势（1 分），又有分页管理可以不连续存储，能够充分利用空间的好处。（1 分）

三、计算题（6 道小题，每小题 5 分，共 30 分）

- 1、设有两个优先级相同的进程 P1 和 P2，共享 x、y、z 三个变量，执行代码见下表。信号量 s1 和 s2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各是多少？

进程 P1	进程 P2
1. y=1;	1. x=1;
2. y=y+2;	2. x=x+2;
3. V(s1);	3. P(s1);
4. z=y+1;	4. x=x+y;
5. P(s2);	5. V(s2);
6. y=z+y;	6. z=x+z;

解：根据信号量的约束，P1-6 一定晚于 P2-5 执行，而 P2-4 一定晚于 P1-3，而其他顺序不受限制。执行顺序不同后导致结果不同的语句有：P1-6, P2-4, P2-6

- (1) 由于 P2-4 一定晚于 P1-3，但一定早于 P1-6，而 P1-3~P1-6 期间 y=3，所以最终 x=6
(2) 如果 P2-6 早于 P1-4 执行，则 P2-6 使 z=6，然后 P1-4 使 z=4，P1-6 使 y=7
(3) 如果 P2-6 晚于 P1-4，但早于 P1-6 执行，则 P1-4 使 z=4，P2-6 使 z=10，P1-6 使 y=13

(4) 如果 P2-6 晚于 P1-6 执行, 则 P1-4 使 $z=4$, P1-6 使 $y=7$, P2-6 使 $z=10$

综上所述:

如果 P2-6 早于 P1-4 执行, 则 $x=6, y=7, z=4$

如果 P2-6 晚于 P1-4, 但早于 P1-6 执行, 则 $x=6, y=13, z=10$

如果 P2-6 晚于 P1-6 执行, 则 $x=6, y=7, z=10$

(以上三种情况, 答对一种得 2 分, 两种得 4 分, 三种得 5 分)

2、若磁头的当前位置为 100 磁道, 磁头正向磁道号增加方向移动, 假设每移动一个柱面需要 3ms 时间。

现有磁盘读写请求队列: 23, 376, 205, 132, 19, 61, 190, 398, 29, 4, 18, 40。

(1) 试采用电梯扫描算法 (SCAN) 和循环电梯扫描算法 (CSCAN), 分别计算为完成上述访问总共花费的寻道时间。

(2) 与 SCAN 算法相比, CSCAN 算法有什么优点?

解: (1) SCAN 的访问顺序: 100, 132, 190, 205, 376, 398, 61, 40, 29, 23, 19, 18, 4, (1 分) 则寻道距离 $32+58+15+171+22+337+21+11+6+4+1+14=692$, 则时间为 $692*3ms=2076ms$ (1 分)

CSCAN 的访问顺序: 100, 132, 190, 205, 376, 398, 4, 18, 19, 23, 29, 40, 61, (1 分) 则寻道距离 $32+58+15+171+22+394+14+1+4+6+11+21=749$, 则时间为 $749*3ms=2247ms$ (1 分)

(2) SCAN 在寻道时照顾中间磁道而对两侧磁道不公平, CSCAN 克服了这个问题。

(1 分)

注: 如果忽略 CSCAN 从 398 磁道回到 4 磁道的时间, 即寻道距离为 355, 时间为 $355*3ms=1065ms$, CSCAN 的优点总结为从最大磁道直接回到最小磁道而节省时间者, 可记为正确。

3、设某文件索引节点中有 12 个地址项, 其中 9 个地址项是直接地址索引, 2 个地址项是一级间接地址索引, 1 个地址项是二级间接地址索引, 每个地址项大小为 4 字节。若磁盘索引块和磁盘数据块大小均为 1KB。

(1) 这个文件系统可表示的单个文件最大长度是多少?

(2) 如果磁盘索引块和数据块的大小保持不变, 索引节点中地址项总个数不变, 请问如何修改, 使得支持的单个文件大小达到 16GB。

解: (1) 索引块为 1KB 大小, 地址项为 4 字节, 则一个索引块可以包含 256 个指针。单个文件最大长度: $(9+2*256+1*256*256) * 1KB = 66057KB = 64MB+521KB$ (3 分)

(2) 可以将一个一级间接地址索引改为三级间接地址索引, 则文件总大小可以达到 $256*256*256*1KB=16G$ (2 分)

4、对于一个将存放在内存中的分页系统, 页面大小为 4KB, 一次内存的访问时间是 100ns, 一次快表 (TLB) 的访问时间是 10ns, 处理一次缺页的平均时间 10^8 ns (已含更新 TLB 和页表的时间)。请问:

(1) 如果不考虑缺页的情况, 对于已经载入内存的页面, 快表命中率为 90%, 则平均有效访问时间是多少?

(2) 如果所需页面未在内存中, 则读入所需数据需要的总时间是多少?

解：(1) 命中：查快表+访问内存存取数据 $10+100 = 110\text{ns}$ (1分)

未命中：查快表+查页表+访问内存存取数据 $10+100+100 = 210\text{ns}$ (1分)

则平均有效访问时间： $110 \times 90\% + 210 \times 10\% = 120\text{ns}$ (1分)

(2) 查快表+查页表+处理缺页中断+查快表+访问内存存取数据
 $10+100+10^8+10+100 = 10^8+220\text{ns}$ (2分)

5、某个系统有 A、B、C、D 四类资源，当前的资源的分配情况和进程的最大需求量如下表所示。当前四类资源的空闲量为 (1, 5, 2, 0) 请使用银行家算法回答下面的问题：

	已分配				最大需求量			
	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	1	1	2
P1	1	0	0	0	1	7	5	0
P2	1	3	5	4	2	3	5	6
P3	0	0	1	4	0	6	5	6

(1) 当前系统是否处于安全状态？如安全，请给出一个安全序列。

(2) 如果从进程 P1 发来一个请求 (0, 4, 2, 0)，这个请求能否立刻被满足？如果能，请给出分配后的一个安全序列。

解：(1)

	已分配				最大需求量				还需要			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	1	1	2	0	1	0	0
P1	1	0	0	0	1	7	5	0	0	7	5	0
P2	1	3	5	4	2	3	5	6	1	0	0	2
P3	0	0	1	4	0	6	5	6	0	6	4	2

空闲量 (1, 5, 2, 0) > P0 需要量 (0, 1, 0, 0)，P0 得到最大需求量后归还全部资源，

空闲量 (1, 5, 3, 2) > P2 需要量 (1, 0, 0, 2)，P2 得到最大需求量后归还全部资源，

空闲量 (2, 8, 8, 6) > P1 需要量 (0, 7, 5, 0)，P1 得到最大需求量后归还全部资源，

空闲量 (3, 8, 8, 6) > P3 需要量 (0, 6, 4, 2)，P3 得到最大需求量后归还全部资源。

故存在安全序列 P0, P2, P1, P3 或 P0, P2, P3, P1。 (2分)

(2) 如果满足 P1 的请求，则分配后空闲量 (1, 1, 0, 0)： (1分)

	已分配				最大需求量				还需要			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	1	1	2	0	1	0	0
P1	1	4	2	0	1	7	5	0	0	3	3	0
P2	1	3	5	4	2	3	5	6	1	0	0	2
P3	0	0	1	4	0	6	5	6	0	6	4	2

空闲量 (1, 1, 0, 0) > P0 需要量 (0, 1, 0, 0)，P0 得到最大需求量后归还全部资源，

空闲量 (1, 1, 1, 2) > P2 需要量 (1, 0, 0, 2)，P2 得到最大需求量后归还全部资源，

空闲量 (2, 4, 6, 6) > P1 需要量 (0, 3, 3, 0)，P1 得到最大需求量后归还全部资源，

空闲量 (3, 8, 8, 6) > P3 需要量 (0, 6, 4, 2)，P3 得到最大需求量后归还全部资源。

故存在安全序列 P0, P2, P1, P3。 (2分)

6、一台计算机有 4 个页框，装入时间、上次引用时间、它们的 R (读) 与 M (修

改) 位见下表 (单位: 时钟周期), 请问 FIFO、LRU 和 CLOCK 算法各将替换哪一页?

页	装入时间	最后引用时间	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

解: FIFO: 第 2 页。

LRU: 第 1 页。

CLOCK: 第 0 页。

(以上三问, 答对一种得 2 分, 两种得 4 分, 三种得 5 分)

四、编程题 (2 道小题, 每小题 10 分, 共 20 分)

- 1、 某火车订票系统, 可共多个用户同时共享一个订票数据库。规定允许多个用户同时查询该数据库, 有查询者时, 用户不能订票; 有用户订票而需要更新数据库时, 不可以有其他用户使用数据库。当有用户申请订票时, 后续的查询者的请求会被暂时挂起直到订票操作完成。请在下面程序的空格处填入 P、V 操作写出查询者和订票者的同步执行程序, 一个空格处可能会有一条或多条 P、V 操作。

答: 程序中没有出现对信号量直接读写的语句, 且 P、V 操作中没有出现非信号量作为参数。(1 分) (程序中的给分项必须 P、V 操作成对完整且位置正确方可得分, 分值标在相应的 P 操作上)

semaphore mutex=1 (1 分), db=1 (1 分), w=1 (1 分); //此处为信号量初始化

int count=0; //共享变量, 查询用户的个数

query() //查询过程

```
{  
    P(mutex) (1 分); P(w) (1 分);  
    count=count+1;  
    if (count==1) { //是第一个查询者  
        P(db) (1 分);  
    }  
    V(w); V(mutex);  
    查询余票;  
    P(mutex) (1 分);  
    count=count-1;  
    if (count==0) { //是最后一个查询者  
        V(db);  
    }  
    V(mutex);  
}
```

```

book() //订票过程
{
    P(w) (1分); P(db) (1分);
    订票;
    V(db); V(w);
}

```

2、在解决类似生产者—消费者问题等进程间通信的问题时，问题中可能会涉及到解决程序间互斥和同步关系，也可能需要共享一些变量，请讨论如何使用 Sys V 的 IPC 机制编写解决这类进程间通信程序。

答：需要使用 Sys V 的信号量 semaphore 和共享内存 shared memory 机制。

信号量 semaphore 函数由 semget、semop、semctl 三个函数组成：（5分）

semget() 得到一个信号量集标识符或创建一个信号量集对象

semop() 完成对信号量的 P 操作或 V 操作

semctl() 得到一个信号量集标识符或创建一个信号量集对象

共享内存 shared memory 函数由 shmget、shmat、shmdt、shmctl 四个函数组成。

（5分）

shmget() 得到一个共享内存标识符或创建一个共享内存对象

shmat() 把共享内存区对象映射到调用进程的地址空间

shmdt() 断开共享内存连接

shmctl() 共享内存管理

注：如果只讲 P、V 操作的一般原理，或介绍解决互斥和同步的一般原理，或给出一些伪代码，酌情给 5 分。