

2016 ~2017 学年第 1 学期期末考试试卷

《操作系统原理 1》(A 卷 共 5 页)

(考试时间: 2016 年 12 月 23 日)

题号	一	二	三	成绩	核分人签字
得分					

一、单项选择题 (每小题 2 分, 共 50 分。请将答案填在第 3 页的表格中, 写在其他位置的答案无效!)

- 1、一个快速SCSI-II 总线上的磁盘转速为7200RPM, 每磁道160个扇区, 每扇区512字节。那么, 理想状态下, 其数据传输率为()。
- A. 576000KB/s B. 7200KB/s C. 9600KB/s D. 19200KB/s
- 2、某单CPU系统中有输入和输出设备各1台, 现有3个并发执行的作业, 每个作业的输入、计算和输出时间均分别为2ms、3ms和4ms, 且都按输入、计算和输出的顺序执行, 则执行完3个作业需要的时间最少是_____。
- A. 15ms B. 17ms C. 22ms D. 27ms
- 3、系统中有3个不同的临界资源R1、R2和R3, 被4个进程P1、P2、P3和P4共享。各进程对资源的需求为: P1申请R1和R2, P2申请R2和R3, P3申请R1和R3, P4申请R2。若系统出现死锁, 则处于死锁状态的进程数至少是_____。
- A. 1 B. 2 C. 3 D. 4
- 4、进程P1和P2均包含并发执行的线程, 部分伪代码描述如下所示:

<div>进程 P1</div> <div>int x=0;</div> <div>Thread1(){</div> <div>int a; a=1; x+=1;</div> <div>}</div> <div>Thread2(){</div> <div>int a; a=2; x+=2;</div> <div>}</div>	<div>进程 P2</div> <div>int x=0;</div> <div>Thread3(){</div> <div>int a; a=x; x+=3;</div> <div>}</div> <div>Thread2(){</div> <div>int b; b=x; x+=4;</div> <div>}</div>
--	--

- 下列选项中, 需要互斥执行的操作是_____。
- A. a=1 与 a=2 B. a=x 和 b=x C. x+=1 与 x+=2 D. x+=1 与 x+=3

- 已知在UNIX文件系统中,
- 5、执行命令touch file1新建一个文件, 然后执行命令ln -s file1 link1为file1创建一个符号链接link1, 再执行ln link1 link2为link1创建一个(硬)链接, 则此时file1的link counter为_____。
- A. 0 B. 1 C. 2 D. 3
- 6、执行命令rm link1删除link1成功, 则此时尝试显示link2的内容, 执行cat link2, 结果为_____。
- A. 提示文件不存在 B. 打开文件file1 C. 打开一个空文件 D. link2被删除

在UNIX文件系统中, 给文件分配外存空间采用的是混合索引分配方式。索引节点(inode)中包含10个直接块指针、1个一级间接块指针、1个二级间接块指针和1个三级间接块指针, 间接块指向的是一个索引块, 每个索引块和数据块的大小一致, 均为1KB, 地址指针所占空间为4B。

- 7、假设该索引节点已经被加载进内存中, 则若要读取文件的前20KB的内容, 需要访问磁盘数据块和索引块共_____个。
- A. 20 B. 21 C. 30 D. 32
- 8、该文件系统能支持的文件最大容量约为_____。
- A. 64KB B. 64MB C. 4GB D. 16GB
- 9、若将数据块的大小修改为4KB, 则该文件系统能支持的文件最大容量约为_____。
- A. 64KB B. 64MB C. 64GB D. 4TB
- 10、若保持数据块大小1KB不变, 在不增加inode中的指针个数的前提下, 取消一个直接块指针, 增加一个三级间接块指针, 则能支持的文件最大容量约为_____。
- A. 64MB B. 32GB C. 64GB D. 16TB
- 11、若inode的大小为128B, NBPI (Number of Bytes Per Inode) 为2048, 则一个32GB大小的文件系统中, 用于存放数据和间接指针的数据块总大小约为_____。
- A. 4GB B. 15GB C. 30GB D. 31GB

- 在 UNIX 文件系统中, 在/dir1 目录下有一个子目录 mydir
- 12、现需要将目录mydir从目录/dir1移动到目录/dir2下, 此操作所需要的最低权限为_____。
- I. 对mydir有wx权 II. 对/dir1有wx权 III. 对/dir2有wx权
- A. 仅I B. 仅II和III C. 仅I和II D. I、II和III
- 13、现希望对mydir加以限制, 即使一个用户对此目录有wx权, 也只能删除owner是自己的文件(mydir的owner例外), 则需要为mydir目录设置_____。
- A. SUID B. SGID C. Sticky bit D. Read-only

在逻辑卷管理（LVM）中，某卷组（VG）vg00中PE大小为64MB，vg00中现有2个物理卷（PV）pv00和pv01，pv00和pv01的大小都是100GB（本题中1GB=1024MB）。

- 14、 为保证数据具有高可靠性，即如果一块硬盘损坏，数据可保不丢失，要求所有的逻辑卷的镜像因子（即数据保存几份互为镜像）设置为2。那么在vg00中能创建的最大LV大小为_____。
- A. 50GB B. 100GB C. 200GB D. 400GB
- 15、 在vg00中创建逻辑卷（LV）lv00，大小为50GB。则逻辑卷lv00的逻辑块（LE）个数为_____。
- A. 800 B. 1600 C. 3200 D. 6400
- 16、 上题中逻辑卷lv00所占物理块（PE）的个数为_____。
- A. 800 B. 1600 C. 3200 D. 6400

某操作系统的内存管理器采用请求式分页，页面大小为 1KB，逻辑地址空间为 32 位，物理地址空间大小为 4 GB，按字节编址。页表采用多级页表，一个页表项大小为 4B。TLB（快表）采用全相联映射，有 4 个页表项，内容如下表所示。

有效位	页号	页框号	...
0	FF180H	-----	...
1	3FFF1H	0F035H	...
1	FFFC6H	3054CH	...
1	03FFFH	0C153H	...

- 17、 该系统的页表项中，最多可以保存_____位标志位。
- A. 8 B. 10 C. 12 D. 16
- 18、 若采用多级页表，要求每级页表均可以装入一个页面内，则应该采用_____级页表较合适。
- A. 0 B. 1 C. 2 D. 3
- 19、 对逻辑地址3FFF1880H转换为物理地址的结果是_____。
- A. 0C153080H B. 0F035880H C. TLB 缺失 D. 缺页

有 3 个 CPU 密集型批处理作业 A、B 和 C 被提交，提交时间分别为 10:00, 10:02, 10:08，预计运行时间分别为 12、5 和 4 分钟。对于下列每种调度算法，忽略进程切换的开销，计算其平均进程周转时间。

- 20、 采用最高响应比优先调度算法，平均进程周转时间为_____。
- A. 16.33分钟 B.16分钟 C. 13.33分钟 D. 13分钟
- 21、 采用最短作业优先调度算法，平均进程周转时间为_____。
- A. 16.33分钟 B.16分钟 C. 13.33分钟 D. 13分钟

某请求页式存储管理，允许用户空间为 32 个页面（每页 2KB），主存为 16KB，如有一个用户程序有 10 页长，且某时刻该用户进程的页表如下表所示

页号	物理块号	是否在 TLB 中
0	8	是
1	7	是
2	4	否
3	9	否
4	5	否
5	3	是
6	2	是
其它	not valid	

- 22、 如果程序执行时遇到逻辑地址1AC5H，则它对应的物理地址为_____。
- A. 7AC5H B. 4AC5H C. 3AC5H D. 缺页
- 23、 页表存放在主存中，对主存的一次存取需要100ns，对TLB表的查找时间为10ns，这次访问耗费时间为_____。
- A. 10ns B. 100ns C. 110ns D. 210ns
- 24、 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80%，则访问内存中数据的平均有效访问时间是_____。
- A. 20ns B. 30ns C. 70ns D. 90ns

假设系统中有 4 个进程和 1 个可分配资源，当前分配和最大需求如下表所示，已知资源的总量为 100。系统采用银行家算法实施死锁避免策略。

进程	当前资源分配量	最大资源需求量
进程 1	20	50
进程 2	15	50
进程 3	30	50
进程 4	10	50

- 25、 在当前时刻若进程2请求该资源数量为10，是否能实施分配？若能，给出安全序列。
- A. 不能分配，因为分配后不存在安全序列。
- B. 不能分配，因为资源不足。
- C. 能分配，分配后存在安全序列 3->4->2->1
- D. 能分配，分配后存在安全序列 3->4->1->2

选择题答题区

1	2	3	4	5	6	7	8	9	10
C	B	C	BC	B	B	B	D	D	B
11	12	13	14	15	16	17	18	19	20
C	B	C	B	A	B	B	D	A	C
21	22	23	24	25					
D	B	D	B	A					

二、简答题（每小题5分，共30分）

1、操作系统中有很多设计思想是相通的，例如逻辑卷管理（LVM）和分页式虚拟内存管理就有一些相似之处，试分析一下这个问题。

答：LVM和分页式虚拟内存管理都解决了一个问题：数据可以不连续存放。这样就可以更有效的利用存储空间。

2、在磁盘臂调度算法中，循环扫描算法（或称为循环电梯算法）（C-SCAN）对扫描算法（或称为电梯算法）（SCAN）有什么改进之处？

答：SCAN算法对两端磁道的响应时间过长，相对中间磁道而言不公平，而C-SCAN公平处理各磁道的请求。

3、在虚拟存储管理中，分段式内存管理方式解决了分页式内存管理中的什么问题，又带来了什么问题呢？

答：分段式内存管理解决了分页式内存管理中划分页时仅根据大小划分，这样可能将无关的内容分到一页中，此页不便设置权限与保护，也不利于共享。也有可能把密切相关的内容分到不同页中，当页面置换算法设置不当时，内存紧张时容易形成“抖动”现象。
分段式内存管理带来的问题是段往往过大，多次分配释放后可能形成大量“外碎片”，利用内存效率不高。

4、Intel IA32体系结构中的保护模式是将逻辑地址转成线性地址再转成物理地址，这种内存管理方式是段页式内存管理方式吗，为什么？

答：Intel IA32的保护模式下内存管理方法不是段页式内存管理方法。因为段页式内存管理是分段时段内再分页，整个是一件事。而保护模式是通过分段将逻辑地址转换成线性地址，第二步通过分页将线性地址转成物理地址，这是两件事。

5、进程可以在运行、就绪和阻塞三个状态之间转换，试分析各种转换的发生时机和引发者。

答：

运行->就绪：运行态进程时间片用完，由scheduler剥夺处理器进行调度。

就绪->运行：处理器空闲时，由scheduler挑选一个就绪进程，分配其处理器开始运行。

运行->阻塞：正在运行的进程，发现缺少资源或等待特定事件发生时，主动放弃处理器。

阻塞->就绪：阻塞态进程，阻塞其运行的条件消失，如缺少的资源可用，或特定的事件发生，这时该进程被释放该资源或引发该事件的进程唤醒被阻塞的进程，被阻塞的进程转入就绪状态，等待被调度运行。

6、LRU页面置换算法是一种比较优秀的算法但是较难实现，为什么？试给出一种可行的近似算法作为LRU的取代方案。

答：

LRU 需求记录所有页面长期运行中被使用的时间和次数，需要大量的快速存储空间，而且比较算法非常复杂，不容易实现。

替代方案可以采用老化（aging）算法，每个页面有一个长度有限的计数器，记录每个 tick 内页面使用情况，记录信息的权重逐次递减。这种算法与 LRU 相比，不能记录每个 tick 中内存使用情况，而且记录的位数有限，但是实现方案较易实现。

三、编程题（每小题 10 分，共 20 分）

1、Linux 中使用 pthread 库来实现多线程。试讨论如何使用 pthread 库创建多线程以及避免可能出现的竞争条件。

答：线程库实行了POSIX线程标准通常称为Pthreads。POSIX线程具有很好的可移植性，使用pthreads编写的代码可运行于Solaris、FreeBSD、Linux 等平台，Windows平台亦有pthreads-win32可供使用。Pthreads定义了一套C语言的类型、函数与常量，它以pthread.h头文件和一个线程库实现。

● 数据类型

pthread_t: 线程ID

pthread_attr_t: 线程属性

● 操纵函数

pthread_create(): 创建一个线程

pthread_exit(): 终止当前线程

pthread_cancel(): 中断另外一个线程的运行

pthread_join(): 阻塞当前的线程，直到另外一个线程运行结束

pthread_attr_init(): 初始化线程的属性

pthread_attr_setdetachstate(): 设置脱离状态的属性（决定这个线程在终止时是否可以被结合）

pthread_attr_getdetachstate(): 获取脱离状态的属性

pthread_attr_destroy(): 删除线程的属性

pthread_kill(): 向线程发送一个信号

● 同步函数，用于 mutex 和条件变量

pthread_mutex_init() 初始化互斥锁

pthread_mutex_destroy() 删除互斥锁

pthread_mutex_lock(): 占有互斥锁（阻塞操作）

pthread_mutex_trylock(): 试图占有互斥锁（不阻塞操作）。即，当互斥锁空闲时，将占有该锁；否则，立即返回。

pthread_mutex_unlock(): 释放互斥锁

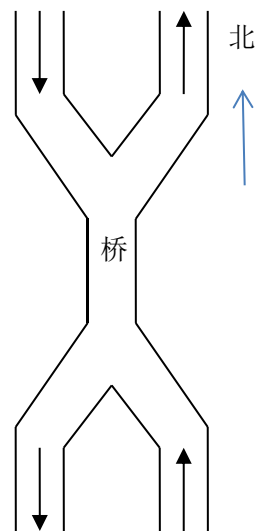
pthread_cond_init(): 初始化条件变量

pthread_cond_destroy(): 销毁条件变量

pthread_cond_signal(): 唤醒第一个调用pthread_cond_wait()而进入睡眠的线程

pthread_cond_wait(): 等待条件变量的特殊条件发生

2、有桥如下图所示，车流如箭头所示，桥上不允许两车交汇，但允许同方向多辆车依次通过（即桥上可以有多个同方向的车）。用 P、V 操作实现交通管理以防止桥上堵塞。



```
int NSCount=0, SNCount=0;
semaphore WNS=1, WSN=1, Bridge=1;
N2S() {
    while(1) {
        P(WNS);
        NSCount++;
        if(NSCount==1){
            P(Bridge);
        }
        V(WNS);
        //通过
        P(WNS);
        NSCount--;
        if(NSCount==0){
            V(Bridge);
        }
        V(WNS);
    }
}
S2N(){
    while(1) {
        P(WSN);
        SNCount++;
        if(SNCount==1){
            P(Bridge);
        }
        V(WSN);
        //通过
        P(WSN);
        SNCount--;
        if(SNCount==0){
            V(Bridge);
        }
        V(WSN);
    }
}
```