**四川大学期末考试试题（闭卷）**

**（2016——2017学年第 2 学期） A卷**

课程号：304019030 课序号： 课程名称：操作系统原理 任课教师： 成绩：

适用专业年级： 学生人数： 印题份数： 学号： 姓名：

|  |
| --- |
| **考 生 承 诺**  我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺：  1、已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；  2、不带手机进入考场；  3、考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。  **考生签名：** |
| 1. 单项选择题(18分，每题1分)    1. 在抢占式调度中，抢占可能发生在下面哪种地方？   \*A) 当时间片超时 B) 当进程发出一个I/O请求  C) 当进程退出 D) 以上全是   * 1. 关于DMA (Direct Memory Access)，下列说法哪个是正确的？   A) 进程可以直接读写一个外部设备  B) 内核可以直接读写进程的内存而不需要缓冲区  C) 进程可以直接读写内核内存而不需要缓冲区  \*D) 外部设备可以直接读写系统内存   * 1. 当一个程序被装入内存准备开始执行时，下面哪个段的大小是操作系统不知道的？   A) text B) data C) bss \*D) heap   * 1. 和非抢占调度相比，抢占调度支持下面哪种状态转移。   A) Ready → Running \*B) Running → Ready  C) Ready → Blocked D) Blocked → Running   * 1. 下面哪个信息是保存在线程控制块(TCB)中的？   A) 打开文件列表 \*B) 堆栈指针  C) 页表 D) 进程ID.   * 1. 某系统使用32位逻辑地址，页大小为4kbytes，以及36位物理地址。那么该系统中的页表大小为：   \*A) 2^20个页表项(2^(32-12)).  B) 2^24个页表项(2^(36-12)).  C) 2^4个页表项 (2^(36-32)).  D) 2^12个页表项   * 1. 在上下文切换期间，操作系统做了以下哪项工作？   A) 修改了页表中的某些项，以反映新进程的内存映射  \*B) 切换页表寄存器指向另外的页表  C) 为新进程修改页表中的访问权限  D) 因为页表是系统级别的资源，所以并不会修改页表   * 1. 假设某系统中的TLB的命中率大约为75%，并且使用了2级页表，那么平均内存时间为：   A) 大约是原来的1.25倍 \*B) 大约是原来的1.5倍  C) 大约是原来的1.75倍 D) 大约是原来的2倍   * 1. 更大的页表增加了： \*A) 内部碎片 B) 外部碎片  C) 页表大小 D) 工作集(Working set)大小   2. USB鼠标是：  A) 块设备 \*B) 字符设备  C) 网络设备 D) 总线设备   3. 下面哪种磁盘调度算法会发生饥饿现象?  A) SCAN \*B) SSTF C) LOOK D) FCFS   4. 下面哪一种RAID级别不能提供数据冗余保护： \*A) RAID 0 B) RAID 1 C) RAID 0+1 D) RAID 5   5. 在文件系统中，簇(cluster)的含义是：   \*A) 最小分配单位，包括多个磁盘扇区  B) 组成一个文件系统的磁盘的集合  C) 一个文件的一组变长的连续磁盘块  D) 为容错而准备的一组磁盘   * 1. 在某基于inode的文件系统中，使用了4Kbytes的块和4bytes的块号。inode中有12个直接块，1个间接块，1个二级间接块。那么该文件系统最大的文件大小为：   A) 大约 64 MB. B) 大约 1 GB.  \*C) 大约 4 GB. D) 大约 16 GB.   * 1. 下面程序会打印多少个“hello”出来？   int main(int argc, char \*\*argv) {  for (int i = 0; i < 3; i++) {  fork();  printf("hello\n");  }  return 0;  }  A) 3 B) 6 C) 8 \*D) 14   * 1. 下列哪项不会导致掉入陷阱(trap)？   A) 用户进程中被0除 \*B) 操作系统内核执行一条特权指令  C) 一个可编程定时器设定的时间到达 D) 用户进程执行一条中断指令   * 1. 下列哪项不是系统调用：   A) 复制一个打开的文件描述符 B) 获取当前工作目录  C) 执行信号量上的P操作 \*D) 创建一个链表   * 1. 日志文件系统中的日志(journaling)的主要作用是： A) 提高磁盘利用率 B) 提高文件系统访问速度 \*C) 用于系统崩溃后的文件系统恢复 D) 访问权限控制  1. 简答题(20分，每题4分)    1. 什么是写时复制(Copy on write)？为了实现它，需要操作系统的哪些支持？  答：写时复制是指当对一页共享内存执行写入操作时先将该页复制，再写入新的页，以达到既共享内存，又可写入的目的。实现写时复制，需要操作系统支持请求分页机制，并在缺页中断处理程序做相应的处理。    2. 什么是系统颠簸(抖动，thrashing)？请解释产生的原因，以及解决该问题的方法。 答：系统颠簸是指系统频繁地忙于页面换出换入。产生的原因是系统物理内存不足，导致缺页中断发生频率过高。解决方法是增加系统内存，或者减少并发的进程数量。    3. 请简单阐述输入/输出控制的几种方式。 答：轮询，中断，DMA，通道    4. 在现代操作系统中，采取哪些措施来隔离不同的进程，让进程不能破坏操作系统本身，也不能破坏其他的进程？ 答：内核态、用户态；各种陷阱；系统调用；分页内存管理，独立的逻辑地址空间    5. 请比较信号量和条件变量之间的区别。 答：信号量有值而条件变量无值；执行wait操作时，信号量会先减少值，小于0才会阻塞进程，而条件变量则直接阻塞进程；执行signal操作时，信号量会增加值，当小于等于0时唤醒一个进程，而条件变量直接唤醒一个阻塞的进程。 2. 在下图所示的汽车过窄桥的例子中，会发生死锁现象。(12分) 3. 请描述该例子中死锁发生的4个必要条件 4. 给出2种死锁预防的解决方案，并指明是摒弃的哪个条件。     答：a) 互斥：一段桥面一个时刻只能允许一辆车占用；非抢占：一辆车占用了一段桥面不能被剥夺；占有并等待：右边车占用了右边的桥面，申请左边的桥面；循环等待：左边的车等待右边车释放右边的桥面，反之亦然。  b) 互斥：将桥面修宽，双向2车道；非抢占：桥上安装一个吊车，将死锁的车子吊起来；占有并等待：一次只能一辆车上桥   1. 很多操作系统和硬件环境提供CAS(CompareAndSet，TestAndSet)原语。CAS原语可以用于实现并发环境下的原子性操作。CAS原语的伪代码如下，其含义为，比较a指向的内容和b是否相同，如果相同，则设置\*a为c，并返回1，否则返回0。(6分)  |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9 | int cas(int \*a, int b, int c) {  if (\*a == b) {  int t = \*a;  \*a = c;  return 1;  } else {  return 0;  }  } |   假设对整数的读取和写入本身是原子操作，请使用CAS实现对整数的原子性增加并返回增加后的值。函数原型如下，请完成该函数的编写。  int increaseAndGet(int \*a) {  int t1, t2;  do {  t1 = \*a;  t2 = t1 + 1;  } while (!cas(a, t1, t2));  return t2;  }   1. 在Windows和Linux操作系统下，有下面的一段程序，其含义是，用一个指针p指向变量x，然后不断增加p，并访问指向的值。请回答下列问题：(12分) 2. 执行这段程序，会打印出一些内存地址和其内存单元内容后，报错并结束。请问发生的是什么错误？哪行代码引起的这个错误？为什么会发生这个错误？ 3. 程序发生错误前打印的最后一句话中，为什么低3位(十六进制，即二进制的低12位)一定为0？  |  |  | | --- | --- | | 1  2  3  4  5  6  7  8 | char x;  int main() {  char \*p = &x;  for (; ; ) {  printf("%X\n", p); // 打印出要访问的内存地址  printf("%x\n", \*p++); // 访问指向的内存  }  return 0;  } |   答：a) 发生的错误为访问非法内存错误。是由第6行代码引起的。因为每个进程有独立的逻辑地址空间，但必须要向操作系统申请后才能实际访问，不断增加的指针会导致访问到并没有申请的内存单元。b) 在Windows和Linux中，都是采用分页内存管理的，内存分配的单位是以页为单位，而一页是4k，所以发生访问非法内存的地方一定发生在访问一个无效页的第1个字节处，所以低12位一定0.   1. 假设下面的几个进程是系统中仅有的进程，并且都没有I/O请求。给定的达到时间和CPU区间长度(CPU burst time)，回答下面的问题。(12分)  |  |  |  | | --- | --- | --- | | 进程 | 到达时间 | CPU区间长度 | | P1 | 0 | 10 | | P2 | 2 | 6 | | P3 | 3 | 1 | | P4 | 5 | 3 |   针对下面三种调度算法，分别画出甘特图，计算平均周转时间。   1. 先来先服务 (FCFS). 2. 抢占的最短剩余时间优先(SJF) 3. 时间片为4的轮转调度(假设新进程会被添加到就绪队列的尾部)   答：   1. 先来先服务(FCFS).  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | | P1 | | | | | | | | | | P2 | | | | | | P3 | P4 | | |   P1: 10-0=10, P2: 16-2=14, P3: 17-3=14, P4: 20-5=15  平均周转时间为： (10+14+14+15)/4=13.25   1. 抢占的最短剩余时间优先(SJF)  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | | P1 | | P2 | P3 | P2 | | P4 | | | | P2 | | | | | P1 | | | | |   P1: 20-0=20, P2: 12-2=10, P3: 4-3=1, P4: 8-5=3  平均周转时间为：(20+10+1+3)/4=8.5   1. 时间片为4的轮转调度(假设新进程会被添加到就绪队列的尾部)  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | | P1 | | | | P2 | | | | P3 | P1 | | | | P4 | | | P2 | | P1 | |   P1: 20-0=20, P2: 18-2=16, P3: 9-3=6, P4: 16-5=11.  平均周转时间为：(20+16+6+11)/4=13.25   1. 某系统中，有多个读者进程来读取一个共享文件，同时有多个写者进程来写入同一个共享文件。为了同步这些进程，定义了如下的一些共享变量：(10分)   int readCount;  FILE \*sharedFile;  semaphore mutex, writeBlock;  读者和写者按以下伪代码描述的动作执行：   |  |  |  |  | | --- | --- | --- | --- | |  | 读者进程 |  | 写者进程 | | 1  2  3  4  5  6  7  8  9  10  11  12 | <其他代码>  P(mutex);  readCount++;  if (readCount == 1)  P(writeBlock);  V(mutex);  Read(sharedFile);  P(mutex);  readCount--;  if (readCount == 0)  V(writeBlock);  V(mutex); | 1  2  3  4 | <其他代码>  P(writeBlock);  Update(sharedFile);  V(writeBlock); |  1. 共享变量readCount, 信号量mutex, writeBlock分别应该初始化为多少？为什么？   readCount: 0 mutex: 1 writeBlock: 1   1. 假设当一个写者正在写入，有3个新的读者到来，并且它们是此时仅有的读者进程，那么这些读者进程会发生情况？如果阻塞，请指出阻塞在哪一行？   第1个读者进程会阻塞在第5行，其他2个读者进程会阻塞在第2行。   1. 假设当有一些读者进程正在读取，这时来了一个新的写者进程，会发生什么情况？   写者进程会阻塞在第2行   1. 这段代码是否会发生饥饿现象？请解释你的判断。   会发生写者饥饿现象。当有读者进程在读取数据，有写者进程阻塞。如果不断有读者进程来到就会先于写者进入临界区读取数据，写者进程可能会无限期延迟。   1. 在某请求分页系统中，有4个可用帧，有下列页面请求序列：(10分)   1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6  分别使用LRU页面置换算法和最佳页面置换算法，填写下表，用星号(\*)标记出缺页，并计算缺页次数。   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   LRU    缺页次数：10次  最优    缺页次数：8次 |