**南开大学计算机与控制工程学院本科生2017-2018年度第一学期操作系统原理课程期末试卷（A卷）**

专业**▁▁▁▁▁**年级**▁▁▁▁▁**姓名**▁▁▁▁▁▁**学号**▁▁▁▁▁▁**成绩**▁▁▁▁▁**

|  |
| --- |
| **得 分** |
|  |

1. **简答题（本题共30分，每题6分，必做） 草稿区**
2. 描述x86体系结构CR3寄存的作用，并描述进程切换时对CR3和TLB进行的操作。

CR3寄存器用于存放当前进程的页表（3分）

进程切换时，CR3中将被写入即将进入运行状态的进程的页表，同时TLB中的内容被自动清空

1. 操作系统是如何保护内核所在的地址空间，使得用户空间的程序程序指令无法访问内核空间的？

CPU有不同的权限状态（或者写出X86的ring0-ring3的四种特权级），

当前运行权限足够高，才可以访问相应的内存空间（答出硬件有权限区别得3分）

利用段描述符将内核所在的空间设置为高权限模式，用户空间的代码段为低权限模式

当内存访问时，CPU硬件检查当前代码的权限状态和要访问的段的权限状态，当不满足权限条件时触发异常

（答出硬件自动检查权限并限制权限不足的情况，得3分）

**草稿区**

1. 什么是抖动现象？什么是Belady现象？

抖动是指当给进程分配的内存小于所需求的工作集区域时，由于内外存之间频繁交换，使得访问外存的时间大

大增加，造成CPU利用率降低的现象。（3分）

Belady现象是指在某些页面淘汰算法时，有时就会出现分配的物理页面数增多但缺页率反而提高的异常现象。

（3分）

1. 在银行家算法的设计过程中，安全状态和死锁状态的逻辑关系是什么？该算法能够使程序处于哪种状态？

安全状态，不安全状态和死锁状态，其中不安全状态存在风险，但未必一定会进入死锁状态。（4分）

银行家算法能够找到系统的安全状态（2分）

**死锁**

**不安全**

**安全**

1. 请写出三种常用的进程调度的方法。

先来先服务，短任务优先，剩余时间短任务优先，时间片轮转，多级队列，优先级

任意写出3种即可，每种占2分，如果不在答案中但方案合理也可得分

|  |
| --- |
| **得 分** |
|  |

**二、编程计算题（本题共5小题，共计40分，选做4题，多做不得分） 草稿区**

* **请在下面的表格中指定答题顺序，在对应的分值下列明题号。每格只许列出一个题号，否则做无效处理。**
* **下表中必须写明所有题目的题号，如果填写不完全，视为放弃未填写部分题目。**
* **如填写内容无效或者不填写表格，则按照前四个小题的成绩计算总分**

|  |  |  |  |
| --- | --- | --- | --- |
| **第一题（10分）** | **第二题（10分）** | **第三题（10分）** | **第四题（10分）** |
|  |  |  |  |

1. 内存访问时间问题：一个二维矩阵Array，使用如下方式初始化：

**int Array [][] = new int [100][100];**

假设系统使用基于页的内存管理模式，每个页的大小为200字节，并假设**Array** [0][0]位于地址200处。一个进程的指令

存放在第0页（即地址范围是0-199），这部分指令用于实现对**Array**的访问。假设该进程只有3个物理页帧，其中指令

部分占据第一个页帧，其他两个页帧是空的，用于加载数据。如果系统使用LRU算法，计算下列代码产生的缺页次数

A：for(int i=0;i<100;i++)

for( int j =0; j< 100; j++)

**Array** [i][j]=0;

B: for(int j=0; j<100;j++)

for(int i=0;i<100;i++)

**Array** [i][j]=0;

注：请写出计算过程。假设系统为32位环境，编程语言为C++，i和j使用寄存器存储不再占据内存空间，不计cache对

内存访问的影响。

答：C++中二维数组按行排列，整数型占4字节，因此数组Array占据40000字节，共200个页面

对于程序A，访问的地址是Array+(i\*100+j)\*4，依次是200,204,208,212……596,600，604……

顺序递增的，因此第一个页面用完了，会把第二个页面换进来

依此类推，因此程序A会依次换入这200个页面换入，共有200次缺页

对于程序B，访问的地址依次是200,600,1000，……10000,204,604……，所以每一个j的循环都会失效100次，共计10000次

本题中虽然提到LRU置换方法，但实际上并没有什么用处。

**草稿区**

1. 进程通信问题： 以一点点段伪代码中，c1和c2是两个共享的整数型变量，初始值都为1

/\*进程1执行的代码\*/

1. void procedure1(){

2． while(1){

3. /\*题目无关的功能代码\*/

4. Other section 1;

5. do{

6. c1=1-c2;

7. }while(c2!=0)

8. /\*临界区代码\*/

9. Critical section;

10. c1=1;

11. }

12. }

/\*进程2执行的代码\*/

21. void procedure2(){

22． while(1){

23. /\*题目无关的功能代码\*/

24. Other section 2;

25. do{

26. c2=1-c1;

27. }while(c1!=0)

28. /\*临界区代码\*/

29. Critical section;

30. c2=1;

31. }

32. }

以上是否是一段有效的临界区互斥访问机制？请从“空闲则入”，“忙则等待”，“有限等待”三个方面进行分析。

（注：每行代码前的数字为代码的行号）

答：不是有效的机制 （2分，但如果此问答错，整题不得分）

1. 可以保证空闲则入，c1,c2 都为1时即为空闲，此时任何一个进程均可通过将自身标记变为0而进入（2分）
2. 不能保证忙则等待，如果p1先执行，到执行在第9行时，此时c1=0，c2=1, 发生调度，P2执行，在第27行之前停下，再调度回P1,执行过第10行，c1=1，此时再调度，P2即可进入临界区，而此时c2=1, c1=1.P2进入临界区之后发生调度，因为c1,c2都是1，P1可以进入临界区。（答出此问得4分）
3. 不能保证有限等待。只要调度不能发生在第9行之后到第6行之前的过程中，那么就会导致p2永远无法调度（1分）

**草稿区**

1. 一个磁盘逆时针旋转，共有3个磁道和一个磁头，每个磁道有12个扇区，最外侧为磁道0，包含扇区0-11，中间为磁道1

包含扇区12-23，最内侧是磁道2，包含扇区24-35。如图所示，现在磁头停在扇区6的位置上。一次磁盘的访问时间可以

分成三部分，寻道时间（磁头移动到相应磁道所需的时间）+旋转时间（等待扇区转到磁头下方的时间）+传输时间（读取

扇区中的数据所需的时间）。

假设磁头在相邻磁道间移动需要40个时间单位，磁盘每旋转1度需要1个时间单位，读取一个扇区的数据需要30个

时间单位（包含了旋转和读取的时间），每个扇区对应的圆心角为15度。

从当前状态开始，使用FIFO（先来先服务）的方式，依次读取编号为10,12,24,1的4个扇区

求完成读取所花费的时间，写出计算过程和计算依据。请提出对改进读取速度的方案。



答：10： 0+105+30

12: 40+335+30

24： 40+305+30

1：80+280+30

本题的主要的失分点在于要考虑到磁头移动过程中盘片仍然在旋转。计算结果完全正确得6分，如果忽略了这部分的时间重叠但其他计算步骤正确可以得4分。

本题中，跨磁道移动是主要的时间消耗，因此考虑最短寻道时间优先的方案，设计思路中在减少磁道移动次数的，可以得3分

在本题中，24和1号为相对位置连续的扇区，但是由于旋转延迟，使得24读完后转到1时，1被错过了。因此，提到扇区斜进技术的，或者重新编制扇区号码的，得1分。

提出的改进方案只要合理，都可以酌情给分。

**草稿区**

1. 某操作系统中共有R1，R2和R3三类资源，共有四个进程需要使用这三类资源。在T0时刻P1，P2，P3和P4对资

源的占用和需求情况如下表所示，此时系统的可用资源量为（R1＝ 2，R2＝ 1，R3＝ 2），以下省略资源编号。

表1 T0时刻4个进程的资源最大需求和占用情况

|  |  |  |
| --- | --- | --- |
|  | 最大资源需求量 | 已分配资源数量 |
| R1 R2 R3 | R1 R2 R3 |
| P1 | 3 2 2 | 1 0 0 |
| P2 | 6 1 3 | 4 1 1 |
| P3 | 3 1 4 | 2 1 1 |
| P4 | 4 2 2 | 0 0 2 |

请回答以下问题：

1. 在资源分配过程中，什么是安全状态？
2. 在T0时刻，P1发出了（1，0，1）的资源分配请求，请问能否分配给P1这些资源？为什么？
3. 在T0时刻，P2发出了（1，0，1）的资源分配请求，请问能否满足P2的要求？为什么？

**1. 本题为三问，其中第一问占总分的20%，第二问和第三问各占题目分值的40%，第三问占题目分值的20%。**

**2. 第一问：何谓安全状态，是指OS可以遵循某一条分配路径，将可用I/O资源分配给各个进程，使得所有进程均能正常运行结束。**

**3. 根据各个进程的已分配资源和最大需求资源，可计算出进程P1尚需要资源（2,2,2），进程P2尚需要资源（2,0,2），进程P3尚需要资源（1,0,3），进程P4尚需要资源（4,2,0）；**

**4. 第二问：进程P1申请（1,0,1），如果分配完成，则剩余可用资源为（1,1,1），无法满足任何进程运行完成的需求，不可以分配。**

**5. 第三问：进程P2申请（1,0,1），如果分配完成，则剩余可用资源为（1,1,1），可以满足进程B的剩余资源需求，进而获得安全路径，可以。**

**草稿区**

1. 在对文件系统进行优化时，常使用“文件控制块分解法”以加快文件目录的检索访问速度。假设文件的控制块中包括文件

名、文件所在位置、文件属性等信息，这些信息保存在文件所在的目录文件中，如下图所示，ABCD四个文件的控制块信息

存在目录文件dir1中，列出dir1中的文件时，会从硬盘中读取这4个文件的控制块数据。

假设一个文件控制块的大小为64字节。为了加快检索速度，将文件控制块分成两部分，一部分包含

文件名和位置信息，占10字节，另一部分是其他属性信息，占54字节，将两部分的信息分别集中存储在一起，以减少在

检索该目录下的文件名时的磁盘读取操作，而在找到目标文件后，才去获取这一目标文件的第二部分信息，从而得出这一

文件的完整的控制块。

假设某一目录文件中有254个文件控制块，每块的大小为64字节，每次磁盘访问时的数据块大小为512字节。

请计算和对比在使用“文件控制块分解法”前后，从该目录中获取某个指定名字的文件的控制块所需平均磁盘访问次数。

（假设文件块顺序存放，每个文件被访问的概率相同，且不存在访问目标不在目录中的情况）



答案：每个文件被访问概率相同，且不存在找不到目标的情况，也就表示从概率上讲要依次查找半数的控制块才能找到目标。

在优化前

(（64\*254）/512)/2 = 16 次

在优化后

（（10\*254）/512）/2+1 = 4 次

算对每一个计5分

忘记除以2，扣2分，忘记优化后的加1，扣2分

|  |
| --- |
| **得 分** |
|  |

**三、系统分析题（本题共3小题，共计30分，选做2题，多做题目不得分）**

**草稿区**

* **请在下面的表格中指定答题顺序，在对应的分值下列明题号。每格只许列出一个题号，否则做无效处理。**
* **下表中必须写明所有题目的题号，如果填写不完全，视为放弃未填写题目。**
* **如填写内容无效或者不填写表格，则按照前两个题目的成绩计算总分**

|  |  |
| --- | --- |
| **第一题（15分）** | **第二题（15分）** |
|  |  |

1. 分析下图所展示的使用虚拟存储技术的多进程操作系统中进程并发数与CPU利用率之间的关系：
2. 解释三个阶段现象，并分析产生的原因。
3. 作为操作系统的管理员，如何侦测到系统正处于阶段3，如何解决这时的出现的CPU利用率低下的问题。



阶段1为进程数不足，进程中的IO等事件发生时，放弃CPU，使得CPU利用率不足 （3分）

阶段2中，进程数设置合理，随着进程数的增加，CPU可以交替使用，从而提高了CPU的利用率 （2分）

阶段3中发生了抖动，即进程数过多，使得每一个进程的得到的可用页面数太少，在运行过程中会频繁引发缺页中断，使得其他进程换出，而自身进程调入。这个调入过程，每次中断产生，都会引起调度，而一旦调度，被调入的进程可能会发生缺页以求换入新的数据。从而使得CPU利用率极速正是降 （5分）

监测这一现象，可以发现系统中进程数量较多，而处理器利用率不高，同时系统中伴随大量的缺页中断产生。（3分）

解决这一问题，可以杀死或挂起部分进程 ，将进程数控制在合理范围内（2分）

意思对，方案合理即可。

**草稿区**

1. 缓存管理问题：磁盘缓存是用来平衡硬盘与内存的读取速度差而引入的一种技术，其结构示意图如下。在系统的内存中开辟

一块内存区域，用于存储硬盘上某些扇区的数据，同时保存这些扇区数据的相关信息，如扇区号，磁道号，访问次数等。

每当系统中产生磁盘的数据访问时，首先按照磁道号和扇区号在缓存中进行查找，如果缓存中存有相应的数据，则称为缓存

命中，将直接使用缓存中的数据而不再访问磁盘；如果缓存未命中，则执行磁盘的读写操作。由于内存资源有限，系统通常

只能分配固定大小的区域用于做扇区数据缓存，因此需要根据系统的运行状态及时对缓存中的数据进行转换。

假设磁盘访问的可以分成以下三类：

1. 针对文件中的热点数据的访问，可能短时间内产生密集的

产生针对若干个扇区的极大数量的访问请求

1. 针对文件夹的热点数据的访问，每间隔一段时间，这些

数据就会被访问一次

1. 非热点数据的访问，访问一次或若干次后就几乎不再访问

请为磁盘的缓存设计一种置换算法，并说明你的设计思路

以及设计出的算法的优势。

旧**区域**

**中间区域**

**新区域**

**栈顶**

**栈**底

· · ·

· · ·

· · ·

把LRU算法中的特殊栈分成三部分，并在每个缓存块增加一个引用计数

栈中缓存块被访问时移到栈顶；如果该块在新区域，引用计数不变；否则，引用计数加1

在新区域中引用计数不变的目的是避免密集访问对引用计数不利影响

在中间区域和旧区域中引用计数加1是为了使用LFU算法

未缓存数据块读入后放在栈顶，引用计数为1

在旧区域中引用计数最小的缓存块被置换

中间区域的定义是为了避免新读入的缓存块在第一次出新区域时马上被置换，有一个过渡期

此题的答案是访问频率置换算法，答出类似方案也给分。如果仅采用了LRU或者LFU等，得5分

如果思路中有两种算法的结合，分别应对了情况（1）和情况2则可以得10分

原因解释合理或者优点分析合理再得5分

**草稿区**

1. 右图为操作系统中使用的两种线程模型的示意图。

请解释这两种线程模型的特点，并从灵活性、线程切换代价、调度管理方面对比这两种模型的优缺点。



左图为用户空间的线程模型ULT，右图内核空间的线程实现KLT。

ULT中整个线程包放在用户空间，内核对线程包一无所知。从内核角度考虑，就是按正常的方式管理，即单线程进程

KLT中每一个用户空间的线程都有一个内核线程与之对应，内核使用进程调度的方法实现对线程的调度和管理。

（5分）

灵活性：ULT更加灵活，不需要OS的支持，也可以自己定义调度的策略 （2分）

线程切换代价：ULT的线程切换代价更低。在KLT中，线程切换需要陷入到保护模式中，从而产生了上下文切换的问题

因此切换代价更高 （3分）

调度管理：ULT不能使用时间片轮转机制，只能等待线程主动放弃执行权，而无法实现线程抢占（2分）

ULT对于内核而言是一个进程，因此一旦一个线程触发IO等阻塞的调用就会导致所有线程都阻塞，调度性较差（3分）