提醒：请诚信应考，考试违规将带来严重后果！

教务处填写：

___年__月__日
考 试 用

湖南大学课程考试试卷

课程名称： 操作系统 ；**课程编码：** CS04007N ；

试卷编号： **A** ；**考试形式：** 闭卷 ；**考试时间：** **120** 分钟。

题 号	一	二	三	四	五	六	七	八	九	十	总分
应得分	6	6	8	4	12	20	16	16	12		100
实得分											
评卷人											

专业班级：

学号：

姓名：

装订线（题目不得超过此线）

（请在答题纸内作答！）

一、（6 分）多核处理器调度采用单队列调度和多队列调度，分别存在什么样的问题，如何处理？

二、（6 分）解释说明分段式内存管理与分页式内存管理的基本思想、地址转换过程，各有什么优缺点。

三、（8 分）什么是死锁？死锁产生需要哪四个必要条件？除死锁外，列举其它类型的同步问题，并简要说明。

四、（4 分）磁盘驱动器一般包含有一定容量的缓存来提高磁盘的访问性能，缓存可以分为 write back 和 write through 两种类型，解释这两种类型缓存的优缺点。

五、（12 分）描述进程与线程以及它们之间的区别和联系。分析给出下图代码（图 5-1）在控制台可能输出的信息；代码（图 5-2）可能会创建多少个进程，多少个线程。

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>

int main(int argc, char *argv[])
{
    printf("hello world (pid:%d)\n", (int) getpid());

    int rc = fork();
    if (rc < 0) {
        // fork failed; exit
        fprintf(stderr, "fork failed\n");
        exit(1);
    } else if (rc == 0) {
        // child (new process)
        printf("hello, I am child (pid:%d)\n", (int) getpid());
        char *myargs[3];
        myargs[0] = strdup("wc");   // program: "wc" (word count)
        myargs[1] = strdup("test.c"); // argument: file to count
        myargs[2] = NULL;           // marks end of array
        execvp(myargs[0], myargs);  // runs word count
        printf("this one print out");
    } else {
        // parent goes down this path (original process)
        int wc = wait(NULL);
        printf("hello, I am parent of %d (wc:%d) (pid:%d)\n",
         rc, wc, (int) getpid());
    }
    return 0;
}
```

图 5-1

```c
#include <unistd.h>
#include <sys/wait.h>
#include <pthread.h>

void *mythread(void *arg) {
    return NULL;
}

int main(int argc, char *argv[])
{
    pthread_t p;
    int rc = fork();
    if (rc == 0) { // child (new process)
        fork();
        pthread_create(&p, NULL, mythread, NULL);
    }
    fork();
    return 0;
}
```

图 5-2

六、（20 分）分页式内存管理。

（1）假设一个 128KB 大小的地址空间，页面大小为 128 字节。假设页表项（PTE）和页目录项（PDE）的大小均为 4 字节。问：给定的地址空间需要多少位表示？偏移（offset）需要多少位表示？虚拟页号（VPN）需要多少位表示？页表需要多大的存储空间？页目录需要多大的存储空间？（6 分）

（2）阅读图 6-1 所示的多级页表访问的伪代码，在带括号的编号 1，2，3，4 处填写适当的注释，在带括号的编号 5，6 处填写适当的代码？（6 分）

（3）假设一个程序的页面访问序列为 4、3、2、1、3、5、4、3、2、1、5，并采用 LRU 算法，设分配给该程序的物理页帧数分别为 3 和 4，计算该页面访问序列中发生的缺页次数和缺页率。（8 分）

```
1  VPN = (VirtualAddress & VPN_MASK) >> SHIFT
2  (Success, TlbEntry) = TLB_Lookup(VPN)
3
4  if (Success == True) // ( 1 )
5    if (CanAccess(TlbEntry.ProtectBits) == True)
6      Offset = VirtualAddress & OFFSET_MASK
7      PhysAddr = (TlbEntry.PFN << ( 5 )) | ( 6 )
8      Register = AccessMemory(PhysAddr)
9    else RaiseException(PROTECTION_FAULT)
10 else // ( 2 )
11   // ( 3 )
12   PDIndex = (VPN & PD_MASK) >> PD_SHIFT
13   PDEAddr = PDBR + (PDIndex * sizeof(PDE))
14   PDE = AccessMemory(PDEAddr)
15   if (PDE.Valid == False)
16     RaiseException(SEGMENTATION_FAULT)
17   else
18     // ( 4 )
19     PTIndex = (VPN & PT_MASK) >> PT_SHIFT
20     PTEAddr = (PDE.PFN << SHIFT) + (PTIndex * sizeof(PTE))
21     PTE = AccessMemory(PTEAddr)
22     if (PTE.Valid == False)
23       RaiseException(SEGMENTATION_FAULT)
24     else if (CanAccess(PTE.ProtectBits) == False)
25       RaiseException(PROTECTION_FAULT)
26     else
27       TLB_Insert(VPN, PTE.PFN, PTE.ProtectBits)
28       RetryInstruction()
```

图 6-1

七、（16 分）分析下面图 7-1 所示的有界缓存的生产者-消费者代码，存在哪些竞争条件？是否正确高效的解决了生产者-消费者问题？如果是，请说明理由；如果没有，请使用所学的并发原语（pthread 的锁、条件变量、信号量）添加代码到合适位置给出你认为的正确解

决方案。

```c
#define MAX 10
int buffer[MAX];

int fill = 0, use = 0, count = 0, loops = 20;

void put(int value) {
  buffer[fill] = value;
  fill = (fill + 1) % MAX;
  count++;
}

int get() {
  int tmp = buffer[use];
  use = (use + 1) % MAX;
  count--;
  return tmp;
}

void *producer(void *arg) {
  for(int i = 0; i < loops; i++){
    put(i);
  }
}

void *consumer(void *arg) {
  for(int i = 0; i < loops; i++){
    int tmp = get();
    printf("%d from the buffer.\n", tmp);
  }
}

int main(void) {
  pthread_t tid1, tid2;
  pthread_create(&tid1, NULL, producer, NULL);
  pthread_create(&tid2, NULL, consumer, NULL);
  pthread_join(tid1, NULL);
  pthread_join(tid2, NULL);
  return 0;
}
```

图 7-1

八、（16 分）假设磁盘的寻道时间（相邻两个磁道之间的寻道时间）为 40 个单位时间，旋转和传输时间都按照 360 度圆的度数来计算，比如传输一个扇区的内容所需的时间为 30 个单位时间（一个磁道包含 12 个扇区），旋转 1 度所需的时间为 1 个单位时间。如下图 8-1 所示的磁盘当前状态，磁头位于 6 号扇区的正中间，队列中有 4 个访问请求，分别访问 10，11，12，13 号扇区。

（1） 计算完成这 4 个请求所需要的时间（按单位时间计算）；

（2） 描述磁盘在处理这一组请求时存在什么问题？解决这个问题可以通过引入磁道偏斜，即相邻磁道的最小编号扇区之间增加一定的偏移量。针对上述 4 个请求，偏斜应该是多少才能尽量减少这一组请求的总时间？

（3） 考虑寻道速率和旋转速率，给出一个公式来计算偏斜（默认的寻道速率和旋转速率都为 1）。

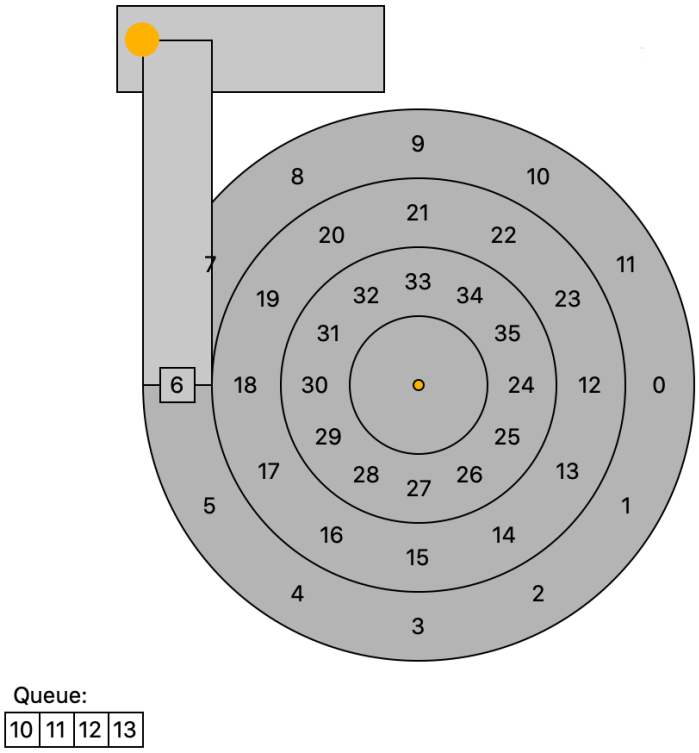

图 8-1

九、 （12 分）假设有 8 个磁盘，每个磁盘的随机读写性能为 R MB/s，顺序读写性能为 S MB/s。现在要用这 8 个磁盘组成两个 RAID（每个 RAID 用 4 个磁盘），要求其中一个 RAID（标记为 RAID-A）满足以下要求：**在某些情况下，多于 1 个磁盘失效不会影响其可靠性**；要求另一个 RAID（标记为 RAID-B）满足以下要求：**在保证可靠性以及顺序读写性能的情况下，随机写性能要求尽可能高**。对于给定的工作负载 W，W 包含 **100R** MB 的随机写请求，**200S** MB 的顺序读请求，如果磁盘调度将 20%的随机写、80%的顺序读请求分配给 RAID-A，其余的分配给 RAID-B，求工作负载 W 的完成时间（列出计算式），要求给出分析和计算过程。