

操作系统 试题卷（A）

题号	一	二	三	四	总分
分数					

一、填空题（每空格 1 分，共 30 分）

得分

1. 操作系统是一直运行在计算机上的程序，它是（1），（2），也是（3）的平台。
2. 多道程序系统（Multi-Programming System）的目的是（4）。
3. 不可中断的指令一般称为（5）指令。
4. 操作系统提供给编程人员的接口是（6）。
5. 为了保护操作系统，操作系统一般提供双重运行模式，分别为（7）和（8）。
6. 操作系统通过（9）来管理和控制进程的相关信息。
7. DMA 直接访问内存是在专门的硬件控制下，实现（10）和（11）之间自动成批交换数据的技术。
8. 程序是（12）的实体，而进程是（13）的实体，它在系统中一般具有（14），（15），（16），（17），（18）5 种状态。
9. 若某一个进程拥有 100 个线程，这些线程都属于用户级线程，则每一个线程在系统调度时间上占用的时间片是（19）。
10. 多线程进程中，因某一个线程的阻塞导致整个进程会被阻塞的关系模型是（20）。
11. 一个进程的运行一般由（21）和（22）组成。
12. CPU 调度算法一般大分为（23）和（24）。
13. 临界区是指进程中用于访问操作（25）的那段代码段。
14. 发生死锁的四个必要条件是（26），（27），（28），（29）。
15. 文件访问控制信息存储的数据结构叫做（30）。

学号:

班级:

姓名:

遵守  
考试  
纪律  
注意  
行为  
规范

## 二、简答题（每题 3 分，共 30 分）

得分

1. 传统的 I/O 操作会涉及到 CPU、内存、I/O 设备控制器、I/O 设备本地缓冲区。请简述 I/O 操作工作流程。
2. 多处理器环境下为什么每个处理器都必须持有自己的寄存器和高速缓存。
3. 简单说明进程在内存中的表现形式，也就是一个进程在内存中结构。
4. 在多线程进程中线程之间可共享的和不可共享的都有哪些。
5. CPU 调度算法的调度准则有哪些。
6. 简单说明解决临界区问题需要满足的三个条件。
7. 请比较说明忙等待（busy-waiting）与阻塞唤醒（block-wakeup）机制。
8. 请简单说明静态链接，动态链接，动态加载，以及它们之间的区别。
9. 请简单说明页表的三种实现方式。
10. 共享文件和目录的实现方法有两种分别为符号链接（symbolic link）和非符号链接（hard link）。请简单说明它们的概念以及区别。

## 三、论述题（每题 5 分，共 20 分）

得分

1. 把一个整数 A 从磁盘读到内存，然后进行加一操作，再把结果存储到硬盘的流程如图 1 所示。

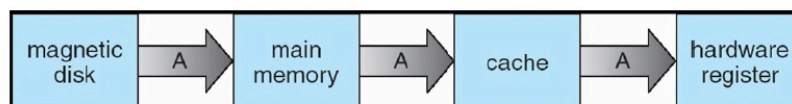


图 1 硬盘 I/O 流程

请根据上面图简述流程，并在多任务环境下，CPU 切换到另一个进程读取 A 的指的时候，请说明可能发生的问题。

2. 通过计数信号量可以实现优先约束，假设要求 P1 的语句 S1 完成之后，执行 P2 的语句 S2。设共享信号量为 synch，信号量初始化为 0，请编写伪代码。
3. 请根据图 2 的资源分配图，先说明当前资源分配和资源请求情况，问：（1）当前是否是死锁状态；（2）什么情况下会发生死锁。
4. 页置换发生两次页的传输，即换入和换出。它会导致页处理时间加倍，增加内存访问时间。为了提高性能采取的有效方法有哪些，请详细说明。

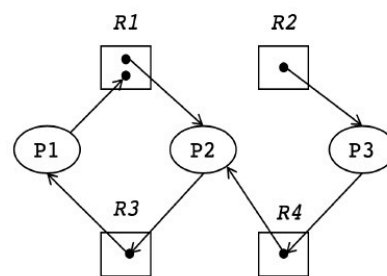


图 2 资源分配图



---

---

#### 四、应用题（每题 10 分，共 20 分）

得分
----

1. 给定的进程有 4 个进程，其到达时间和运行时间分别为  $P1(0,6)$ ， $P2(1,4)$ ， $P3(4,2)$ ， $P4(5,4)$ ，请分别利用“抢占式最短作业优先调度算法”和“非抢占式最短作业优先调度算法”进行调度，用甘特图画出其调度顺序。

2. 银行家算法是在多资源、多实例环境下解决死锁问题的方法。它有资源请求算法和安全性算法组成。资源请求算法判断是否允许资源的请求，安全性算法判断计算机系统是否处于安全状态。符号表示如下： $req$  表示请求资源数量； $avail$  表示可分配的资源数量； $alloc$  表示已占有的资源数量； $need$  表示为完成任务仍然需要的资源数； $finish$  表示当前请求是否安全， $i$  表示  $i$ -th 进程。

● 资源请求算法伪代码

```
IF ( req_i <= need_i)
    IF ( req_i <= avail_i)
        (1) _____ i
        alloc_i = alloc_i + req_i;
        (2) _____ i
        do safety check algorithm;
    ELSE
        waiting;
    End IF
ELSE
    output error message;
End IF
```

● 安全性算法伪代码

```
safety check algorithm:
work = avail_i;
For all i, finish[i] = false;
For all i do
    IF ( finish[i] == false && need_i <= work)
        (3) _____;
        (4) _____;
    End IF
End For
IF for all i, (5) _____
    Then the system is safety
End IF
```