

## 参考答案

A 卷:

一、单项选择题（10 道小题，每小题 2 分，共 20 分。请将答案填在下面的表格中，写在其他位置的答案无效！）

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

二、简答题（10 道小题，每小题 5 分，共 50 分）

1、PV 原语常见有下面二种定义，都可以实现在信号量 s 上的操作，请分析一下这二种定义有什么不同。

第一种定义:

P(s) {s--; if(s<0) sleep; }

V(s) {s++; if(s<=0) wakeup; }

第二种定义:

P(s) { while (!s>0); s--; }

V(s) {s++; }

答：第一种定义对信号量采用睡眠与唤醒的方案，当无法进入临界区时转入阻塞态。第二种宣言为忙等待方案，无法进入临界区时循环检查，这样比较浪费 CPU 时间。

2、在引入线程概念的操作系统中，操作系统将资源分配给线程还是进程？为什么？在线程实现的二种方案中，线程实现在用户级与实现在内核级相比，有什么缺点？

答：资源分配给进程，因为进程是资源分配的最小单位，线程是在进程内，共享使用进程的资源。线程实现在用户级，当一个线程阻塞时，内核会阻塞这个线程所在进程，导致这个进程中其他可以运行的线程也被阻塞。

3、现有一个 1.5T 的数据库文件要存放，但是只有二个 1T 容量的硬盘，请问你有什么办法来解决这个问题吗？介绍一下这种方法的基本原理。

答：可以采用 RAID 技术，将二个硬盘组合成一个逻辑盘。

可以使用 RAID-0 条带化技术，二个硬盘组成的逻辑盘容量为二个硬盘容量之和。

也可以采用 LVM 技术。

二个硬盘是二个 PV，组成一个 VG，VG 的大小是二个 PV 大小之和。在 VG 中划分一个大于 1.5T 的 LV，LV 中建立 FS，即可存入此文件。

4、什么是 SPOOLing 技术？它是如何在解决死锁问题中得到应用的？

答：SPOOLing 技术又称为“假脱机技术”。它可以破坏”独占使用“条件。例如打印机为临界资源，可以在硬盘上开辟一个目录，所有的打印任务都提交的这个目录下的队列中，由一个打印队列监控程序（daemon）负责将任务送到打印机上打印。

5、为什么要使用倒排页表？倒排页表面临的最大的问题是什么？如何解决？

答：在 64 位系统中，由于虚拟地址太大，普通页表会非常大，无法存储。另一方面，实际内存相对较小，所以建立一张从物理地址索引得到相对地址的倒排页表。

最大的问题的难于从相对地址查找到绝对地址。可以采用 hash 表提高查找效率，并使用 TLB 加速查找。

6、在文件的扩展权限中，SUID 权限的设置主要为了是解决什么问题的？试举例说明这类问题。为什么说 SUID 必须要谨慎使用，否则会造成安全隐患？

答：SUID 指的是一个程序在执行时，其进程的有效 UID 将不是执行者的 UID，而是程序属主的 UID。SUID 主要解决权限分配的问题。例如，任何人都可以使用 passwd 修改自己的口令，但是口令文件又不能设置为任何人可访问，所以，属主为 root 的 passwd 命令，必须拥有 SUID，运行时以 root 身份，才可以修改口令文件。SUID 非常危险，设置不当，任何人就可以以 root 身份运行一个程序，带来系统安全隐患。

7、当检测到死锁发生时，如果必须杀死一个进程以解除死锁，请问以什么标准来选择被杀死的进程比较合理？

答：一般选择运行时间较短的进程，因为这样重新运行的代价较小，另外，程序需要可以多次运行不影响执行结果。还要考虑杀死优先级较低的进程等。

8、内存分区管理中的交换技术与请求式分段技术相比，有什么相同点和不同点？

答：相同点都是为了在内存不足的情况下装入更多的进程，都是会产生外碎片。不同点为交换技术交换的对象是整个进程而请求式分段交换的进程中的一个段。

9、在页面淘汰算法中，为什么说老化(Aging)算法只是 LRU 的一个近似实现？

答：老化算法只有有限位的存储，记录最近若干次页面使用情况，更早的使用情况会丢失。另外，老化算法没有记录在一个时间周期内页面的使用频度和时间等，所以与 LRU 相比只是一个近似实现。

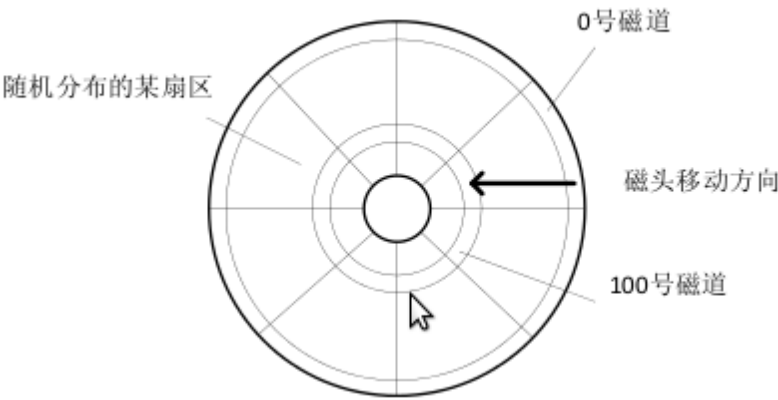
10、外部设备常常被分为块设备和字符设备，请问这二类设备各有什么主要特征？DMA 方式适应于哪种设备，它与中断方式相比有什么好处？

答：字符设备不支持随机访问，一般传输速率较慢，数据量小，如键盘、打印机等。块设备支持随机访问，一般传输时数据组织成块，数据量大，速度较快。DMA 适用于块设备。它将内存与外设之间建立直接通道，在开始和结束时由 CPU 中断处理，其他时间不需要 CPU 干预，减轻了 CPU 的负担。

三、计算题（2 道小题，每小题 5 分，共 10 分）

1、假设计算机系统采用 SCAN(电梯扫描)磁盘臂调度策略。设某单面磁盘旋转速度为每分钟 6000 转，每个磁道有 100 个扇区，相邻磁道间的平均移动时间为 1ms。若在某时刻，磁头位于

100 号磁道处,并沿着磁道号增大的方向移动(如下图所示),磁道号请求队列为 50,90,30,120,对请求队列中的每个磁道需读取 1 个随机分布的扇区,则读完这 4 个扇区点共需要多少时间?要求给出计算过程。



采用 CSCAN 调度算法,访问磁道的顺序和移动的磁道数如下表所示:

被访问的下一个磁道号	移动距离(磁道数)
120	20
90	30
50	40
30	20

移动的磁道数为 20+30+40+20=110,故总的移动磁道时间为 110ms。

由于转速为 6000r/ m,则平均旋转延迟为 5ms,总的旋转延迟时间=20ms。

由于转速为 6000r/ m,则读取一个磁道上一个扇区的平均读取时间为 0.1ms,总的读取扇区的时间平均读取时间为 0.1ms,总的读取扇区的时间为 0.4ms。

综上,读取上述磁道上所有扇区所花的总时间为 130.4ms。

2、请求分页管理系统中，假设某进程的页表内容如下表所示:

页号	页框(Page Frame)号	有效位(存在位)
----	-----------------	----------

0	101H	1
1	----	0
2	254H	1

页面大小为 4KB，一次内存的访问时间是 100ns，一次快表(TLB)的访问时间是 10ns，处理一次缺页的平均时间  $10^8$  ns(已含更新 TLB 和页表的时间)，进程的驻留集大小固定为 2，采用最近最少使用置换算法(LRU)和局部淘汰策略。假设 (1) TLB 初始为空; (2) 地址转换时先访问 TLB，若 TLB 未命中，再访问页表(忽略访问页表之后的 TLB 更新时间); (3) 有效位为 0 表示页面不在内存，产生缺页中断，缺页中断处理后，返回到产生缺页中断的指令处重新执行。设有虚地址访问序列 2362H、1565H、25A5H，请问：

- 1) 依次访问上述三个虚地址，各需多少时间?给出计算过程。
- 2) 基于上述访问序列，虚地址 1565H 的物理地址是多少?请说明理由。

(1) 根据页式管理的工作原理,应先考虑页面大小,以便将页号和页内位移分解出来。页面大小为 4KB, 即  $2^{12}$ , 则得到页内位移占虚地址的低 12 位, 页号占剩余高位。可得三个虚地址的页号 P 如下(十六进制的一位数字转换成 4 位二进制, 因此, 十六进制的低三位正好为页内位移, 最高位为页号):

2362H:P=2, 访问快表 10ns, 因初始为空, 访问页表 100ns 得到页框号, 合成物理地址后访问主存 100ns, 共计  $10ns+100ns+100ns=210ns$ 。

1565H:P=1, 访问快表 10ns, 落空, 访问页表 100ns 落空, 进行缺页中断处理  $10^8$  ns, 访问快表 10ns, 合成物理地址后访问主存 100ns, 共计  $10ns+100ns+10^8 ns+10ns+100ns=100\ 000\ 220ns$ 。

25A5H:P=2, 访问快表, 因第一次访问已将该页号放入快表, 因此花费 10ns 便可合成物理地址, 访问主存 100ns, 共计  $10ns+100ns=110ns$ 。

(2) 当访问虚地址 1565H 时, 产生缺页中断, 合法驻留集为 2, 必须从页表中淘汰一个页面, 根据题目的置换算法, 应淘汰 0 号页面, 因此 1565H 的对应页框号为 101H。由此可得 1565H 的物理地址为 101565H。

#### 四、编程题（20 分）

1、（10 分）2010 世博会在上海成功举行，很多场馆都给人留下了深刻的印象。中国馆有很多观众参观。为保持场内卫生，需要不定期的清馆打扫卫生。为保证秩序，相关部门做出了以下的管理规定：

- 1) 同时进入场馆的人数上限为 N；如果场内观众人数达到上限，新观众在场外排队等候。
- 2) 为保证打扫卫生工作的正常开始，保洁人员首先会暂停新观众进场，新观众在场外排队等候；
- 3) 如果场内无观众，则打扫卫生立即开始，如还有剩余观众，则待场内观众全部离开后，即开始打扫卫生；
- 4) 完成后重新开放。

为实现上述控制，请用 PV 原语和信号量，分别描述观众和保洁人员的行为。

```

semaphore mutex=1,r=N,w=1,s=1;
int rc=0;
reader()
{
    P(s);
    P(mutex);
    rc++;
    if (rc==1) P(w);
    P(r);
    V(mutex);
    V(s);
    读文件;
    P(mutex);
    rc--;
    if (rc==0) V(w);
    V(r);
    V(mutex);
}

writer()
{
    P(s);
    P(w);
    写文件;
    V(w);
    V(s);
}

```

2. (5 分) 假定下面的 C 语言程序在 UNIX 系统上运行，并且所有系统调用都能成功完成。其中“pthread\_create(&t, NULL, bar, NULL);”的功能是创建一个新线程来执行函数 **bar**，并返回线程对象标识 t。“pthread\_join(t,NULL);”的功能是等待线程 t 结束。试问此程序在运行过程中会打印出多少个“hello”？需要说明分析过程。

```

#include <sys/types.h>
#include <unistd.h>
#include <pthread.h>
#include <stdio.h>
static int counter = 1;
static void* bar(void *ignore){
    counter--;
    if(counter==0){
        fork();
        printf("hello\n");
    }
    return NULL;
}

```

```
static void foo(){
    pthread_t t;
    if(fork()==0){
        fork();
        pthread_create(&t,NULL,bar,NULL);
        printf("hello\n");
        pthread_join(t,NULL);
    }
}
int main(){
    foo();
    printf("hello\n");
    return 0;
}
```

答：共打印 11 次 hello。