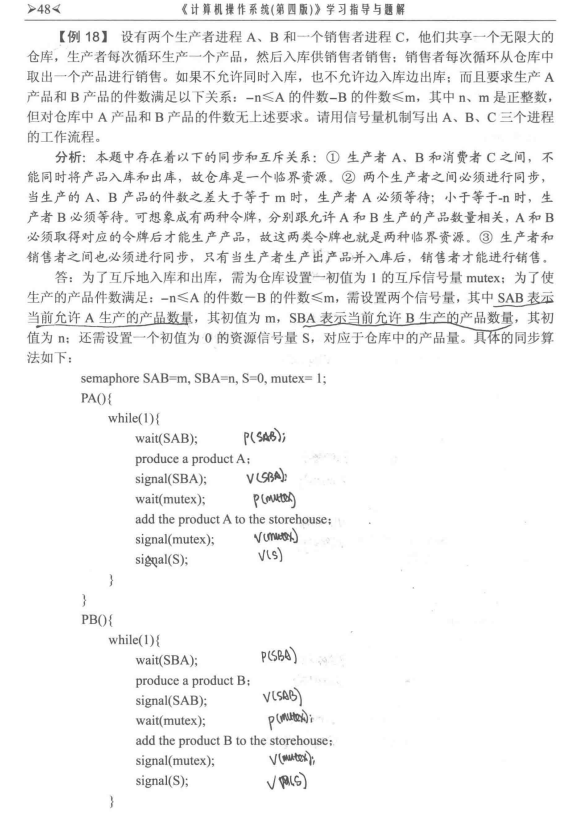
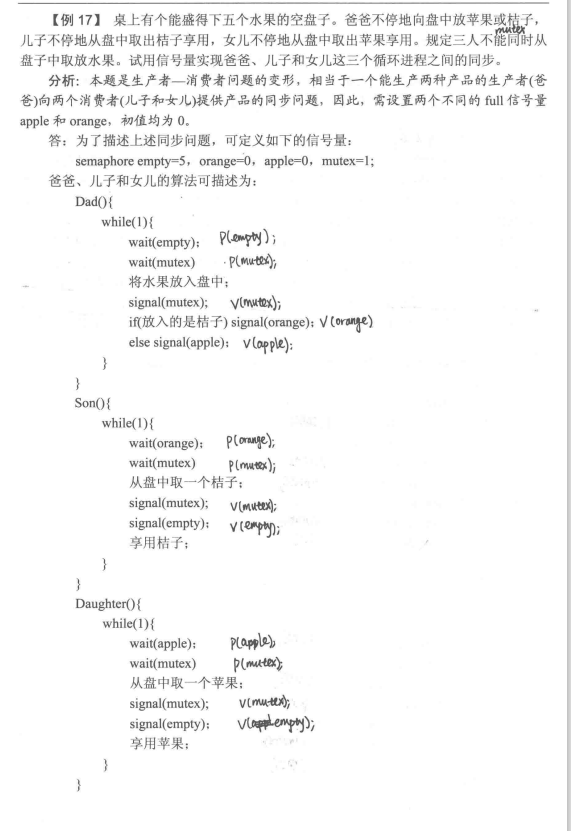
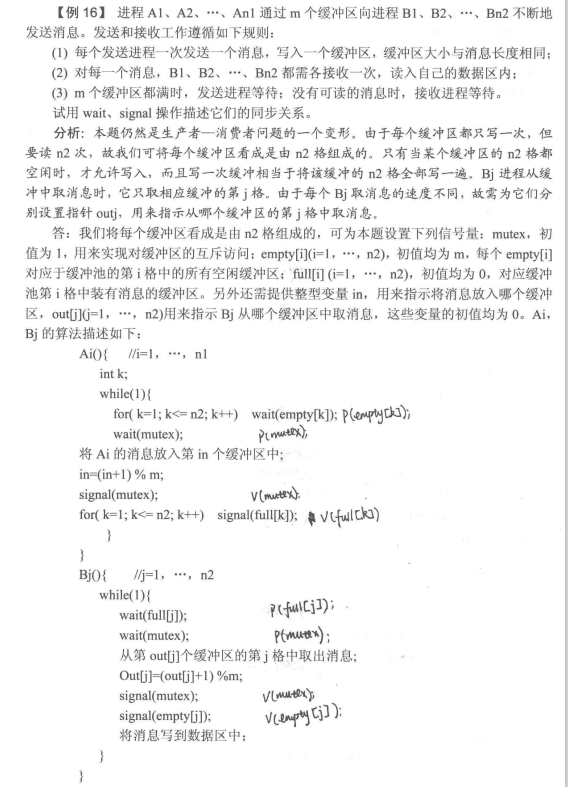
一：综合题

1. PV：



例1 父亲Father、女儿Daughter、儿子Son互斥使用一个包含20个格子的容器。Father每次取一个水果（苹果或香蕉），用putfruit()把水果送入容器的某一个空格子中；Daughter每次用getapple()从该容器中取出一个苹果并用countapple()统计苹果的个数；Son每次用getbanana()从该容器中取出一个香蕉并用countbanana()统计香蕉的个数。请用信号量机制实现三者的同步与互斥活动。

参考答案：

semaphore mutex=1;

semophore apple=0,banana=0;

semophore empty=20;

main()

{ cobegin

{ 进程Father：

While(true)

{ 取水果;

P(empty);

P(mutex);

putfruit();

V(mutex);

If 水果是苹果 V(apple);

else V(banana);

}

进程Daughter：

While(true)

{ P(apple);

P(mutex);

getapple();

V(mutex);

V(empty);

countapple();

}

进程Son：

While(true)

{ P(banana);

P(mutex);

getbanana();

V(mutex);

V(empty);

countbanana();

}

}

coend.

}

例2 某寺庙，有小和尚、老和尚若干。有一水缸，由小和尚用水桶从井中提水入缸，老和尚用水桶从缸里取水饮用。水缸可容10桶水，水取自同一井中。水井径窄，每次只能容一个水桶取水。水桶总数为3个。每次入、取缸水仅为1桶，且不可以同时进行。试用P、V操作给出小和尚、老和尚动作的算法描述。

分析：

小和尚从井中取水并向缸中倒水为一个进程，而老和尚从缸中取水为另一个进程。

有关互斥的资源有：

水井（一次仅允许一个水桶进出）；

水缸（一次倒水、取水仅一个水桶）。

分别为它们设置信号量mutex1、mutex2来实现互斥，初值均为1。

有关同步的问题是：

3个水桶----无论是从井中取水还是倒水入缸或取水出缸都是一次一个，即为其设置信号量count，初值为3，抢不到水桶的进程只好等待。

此外，设置信号量empty来控制入缸的水量，初值为10，当水缸满时不可入水；设置信号full控制出缸的水量，初值为0，当水缸空时不可出水。

参考答案：

Begin

mutex1:=1;mutex2:=1;

empty:=10;full:=0;

count:=3;

Cobegin

小和尚i（i=1,2,…）打水；

老和尚j（j=1,2,…）取水；

Coend;

End.

小和尚i（i＝1,2,…）打水:

Begin

Repeat

P(empty); /\*看水缸满否,满则阻塞打水进程\*/

P(count); /\*申请打水的桶\*/

P(mutex1);/\*互斥使用水井,即不允许两和尚同时打水\*/

从井中取水；

V(mutex1);

P(mutex2); /\*互斥使用水缸\*/

送水入缸;

V(mutex2);

V(count); /\*归还水桶\*/

V(full) /\*水缸又多一桶水\*/

Until false

End;

老和尚j(j=1,2,…)取水:

Begin

Repeat

P(full); /\*看水缸是否有水，无水则阻塞取水进程\*/

P(count); /\*申请取水的桶\*/

P(mutex2); /\*互斥使用水缸\*/

从缸中取水；

V(mutex2);

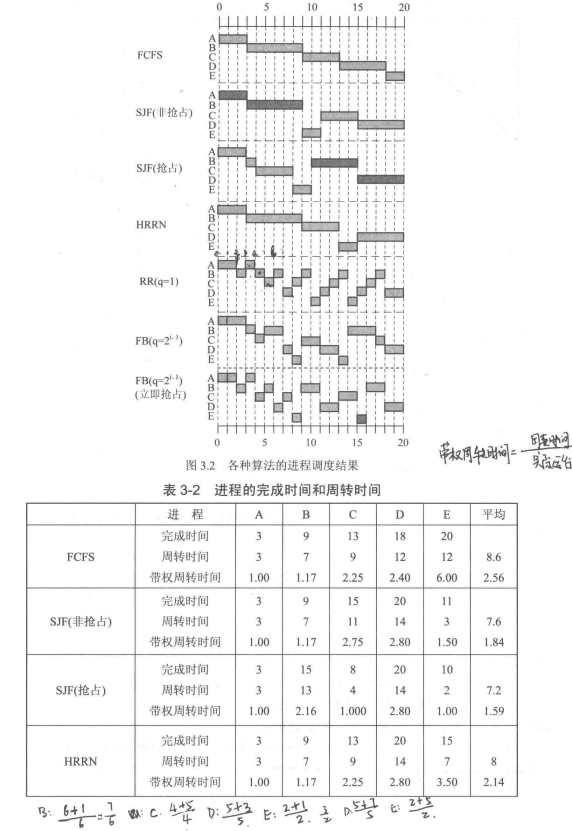
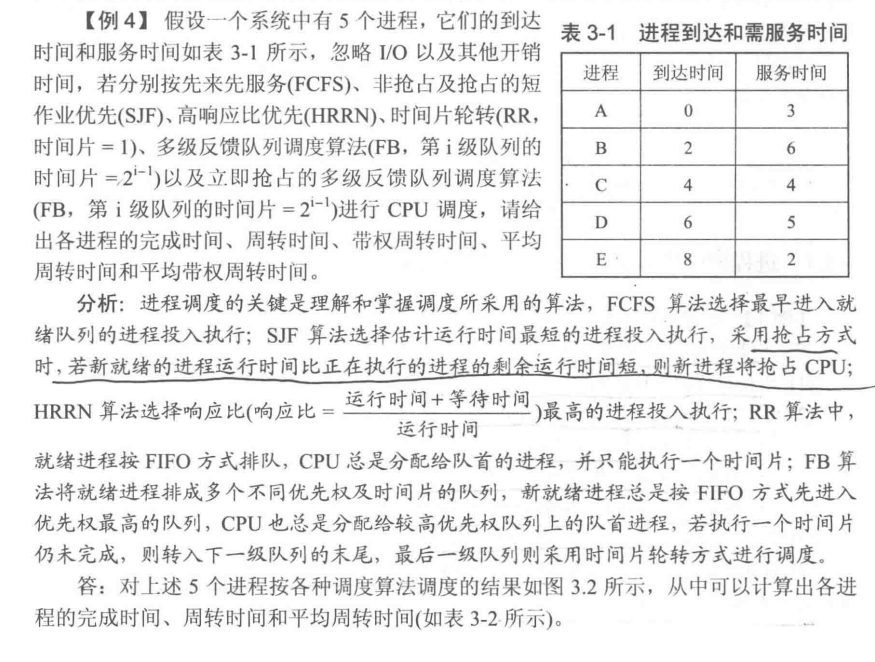
V(count) /\*归还水桶\*/

V(empty); /\*缸中少了一桶水\*/

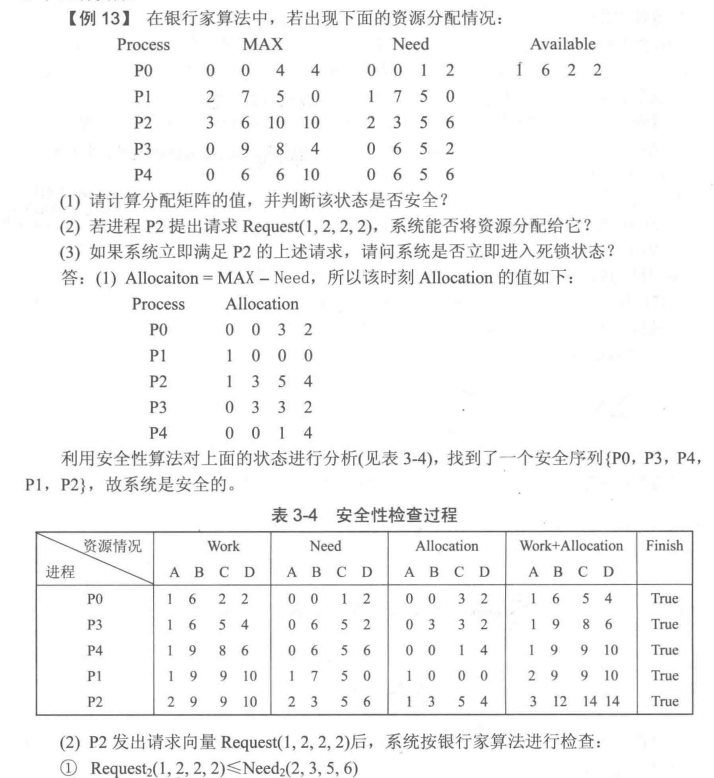
Until false

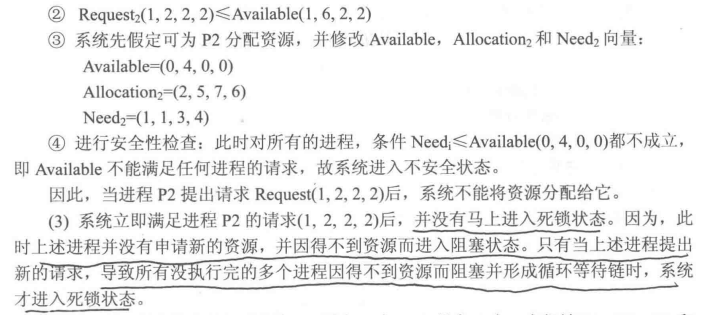
End.

1. 调度算法

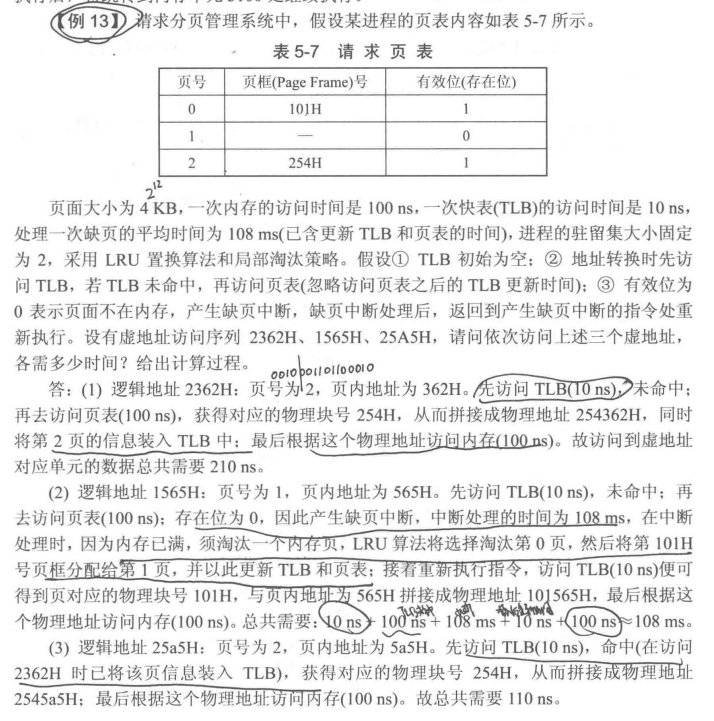


1. 避免死锁算法（银行家算法）

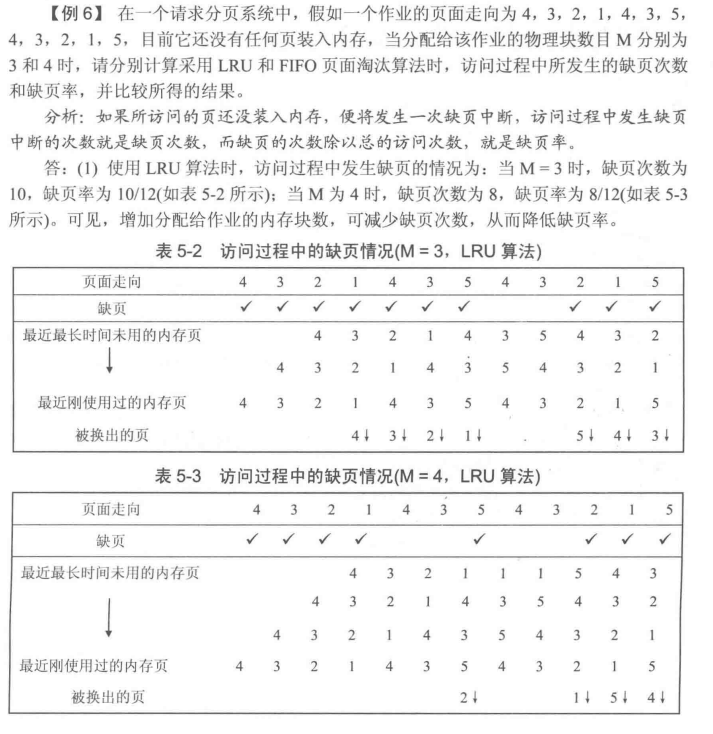




1. 逻辑地址转物理地址

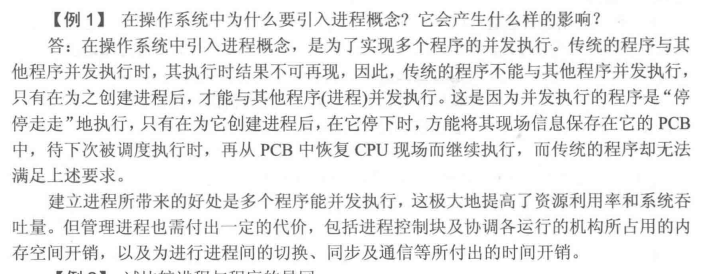


1. 页面置换算法

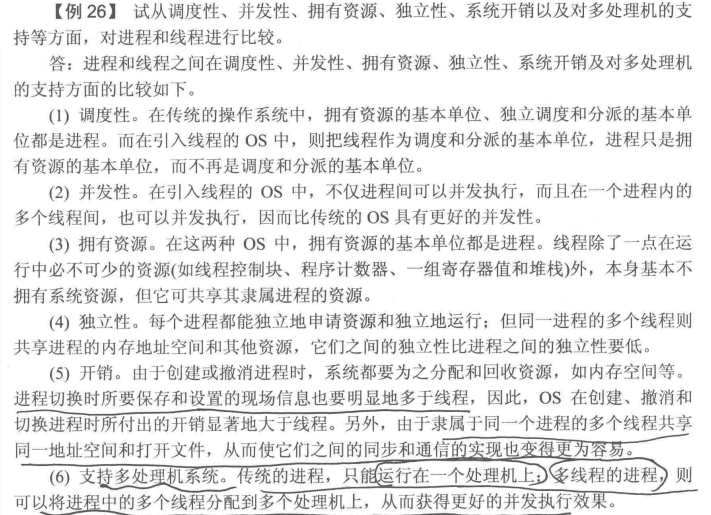


二、简答题

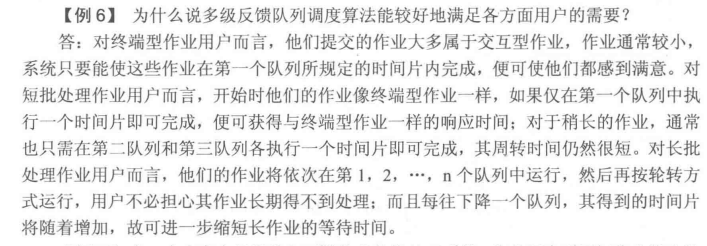
1.进程



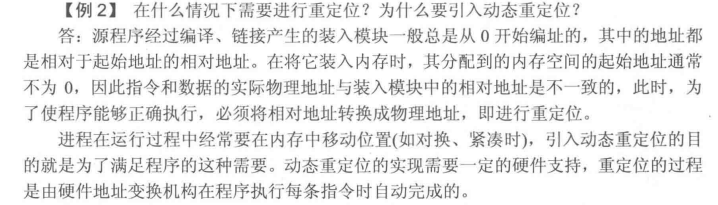
进程与线程



2.多级反馈队列



3.动态重定位



分页、分段的区别

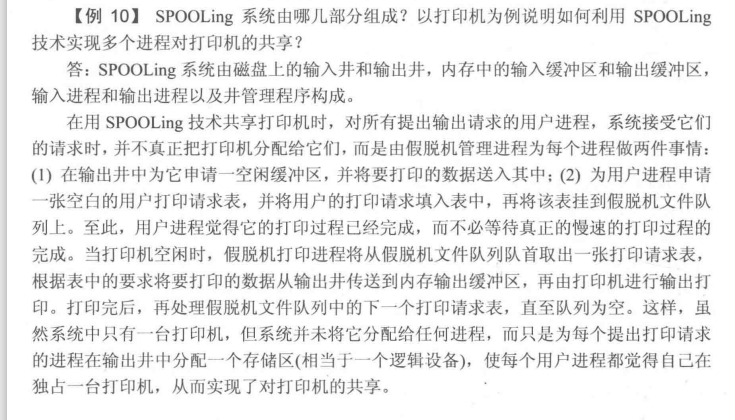
主要区别表现在以下三个方面：

(1) 页是信息的物理单位，分页是为实现离散分配方式，以消减内存的外零头，提高内存的利用率。段则是信息的逻辑单位，它含有一组其意义相对完整的信息。分段的目的是为了能更好地满足用户的需要。

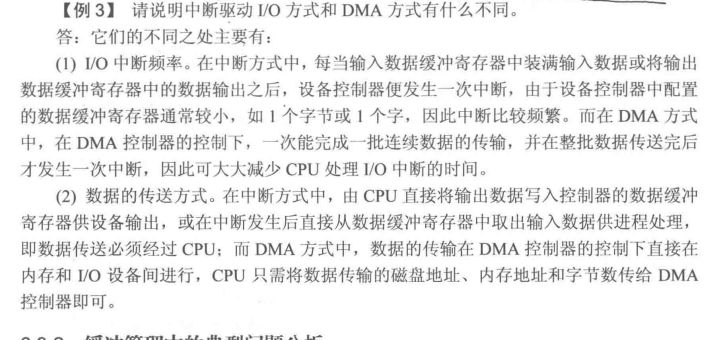
(2) 页的大小固定且由系统决定；而段的长度却不固定，决定于用户所编写的程序。

(3) 分页的地址空间是一维的，程序员只需利用一个记忆符，即可表示一个地址；而分段的作业地址空间是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

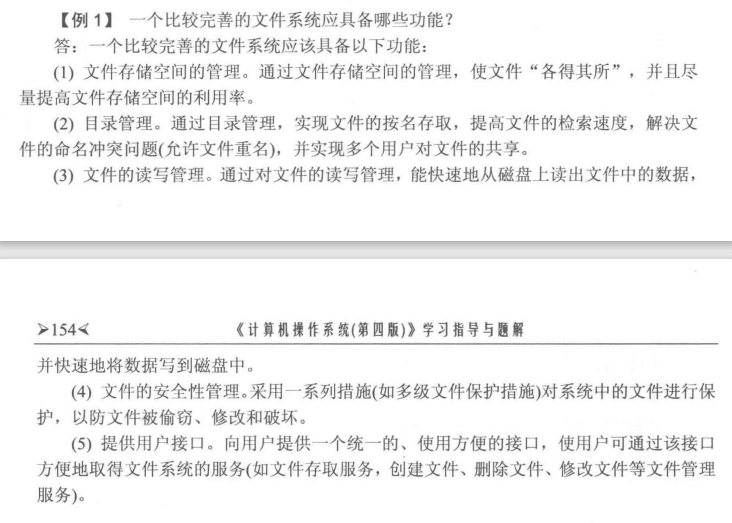
4.spooling技术



5.中断I/O与DMA区别



1. 文件系统功能



三、填空

1.信号量解决进程间的什么问题？同步问题，互斥问题

2.进程的三种状态：阻塞，就绪，执行。

以下情况分别对应哪些状态？

输入输出数据时：阻塞

输入输出结束，等待处理机被调度：就绪

正在被计算：执行

3.存储器管理：分页管理，分段管理，段页式

内存碎片尽可能少，主存利用率最高：分页管理

能动态扩充：分段管理

主存利用率高，可以对段进行保护：段页式

4.作业，进程，程序（？）

一个作业对多进程

一个程序对多进程

一个段对一张页表

一个进程对应一个段表

5.虚拟存储器

虚拟地址空间：逻辑地址，不受内存物理空间大小影响，虚拟存储器可管理的空间直接取决于处理器中地址寄存器的位数。

物理地址空间：物理地址，根据内存大小决定

6.临界资源：进程同步机制应遵循的准则：空闲让进、忙则等待、有限等待、让权等待。

7.设备的一般分类（按信息交换单位分类）：存储设备（块设备），以块为单位，如磁盘，扇区；输入/输出设备（字符设备），以字符为单位。

8.设备分配的数据结构四张表：设备控制表--DCT、控制器控制表--COCT、通道控制表--CHCT、系统设备表—SDT

9.通道：负责输入输出工作：处理机

通道所执行的程序：通道程序

10.文件分为：记录式文件、流式文件

11.Unix项由文件名和相应的索引节点号组成的。

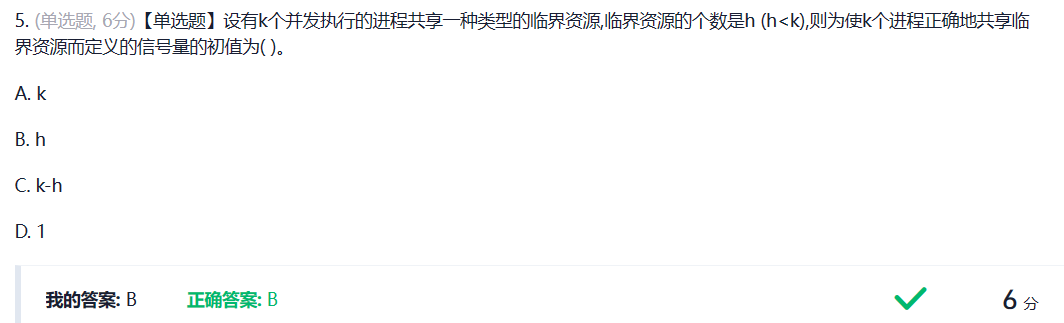
12.文件的物理结构：文件在外存上的组织形式。

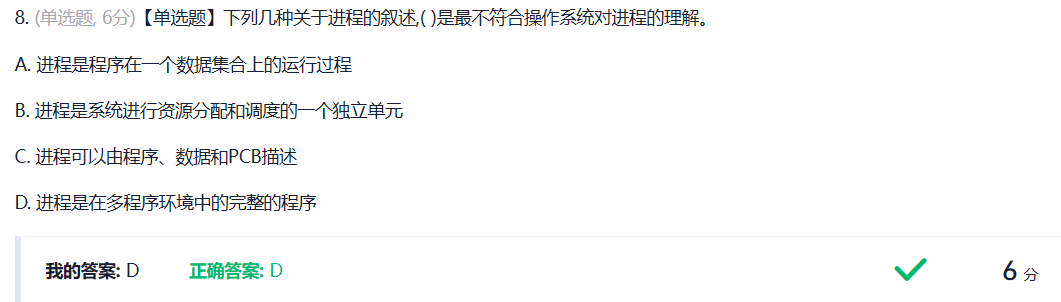
三种物理结构：顺序文件，链接文件，索引文件

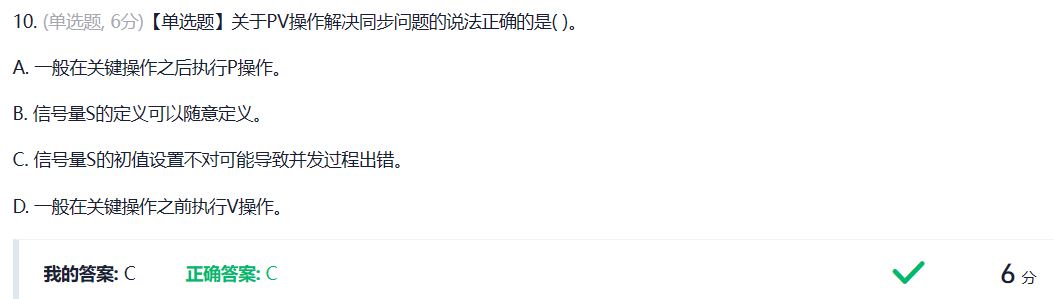
13.文件的存取方式：顺序存取，直接存取

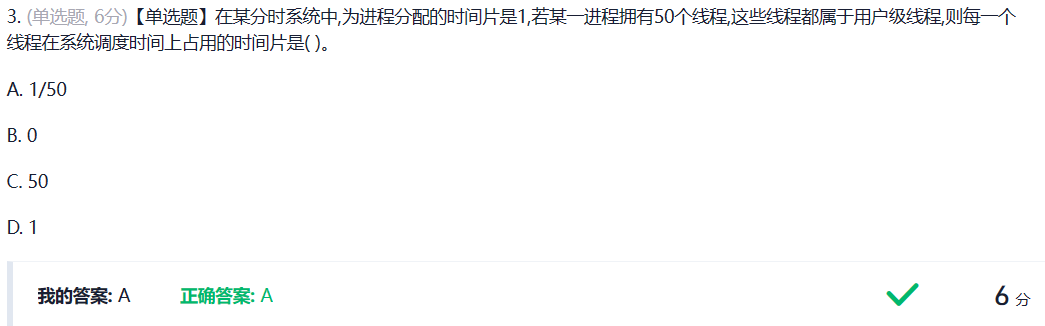
四、选择\判断

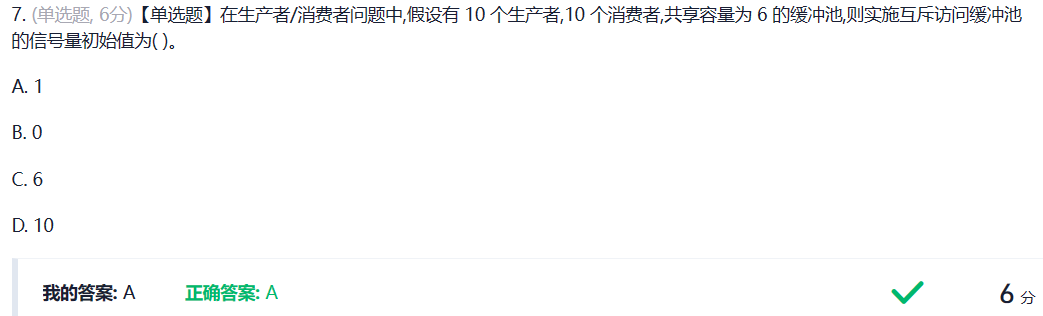
第二章：

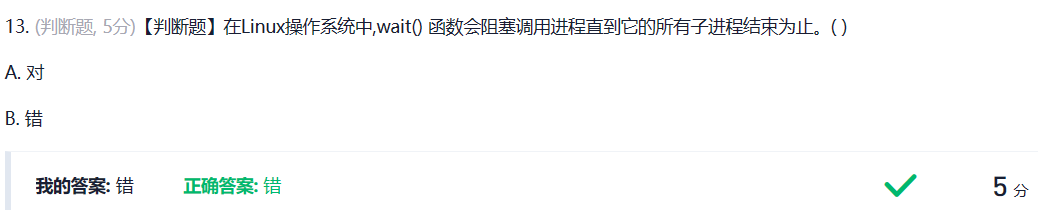


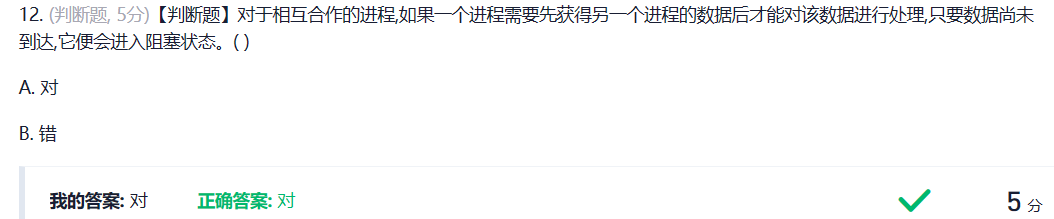


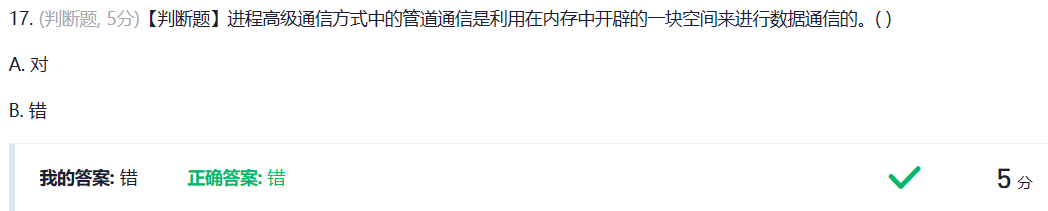
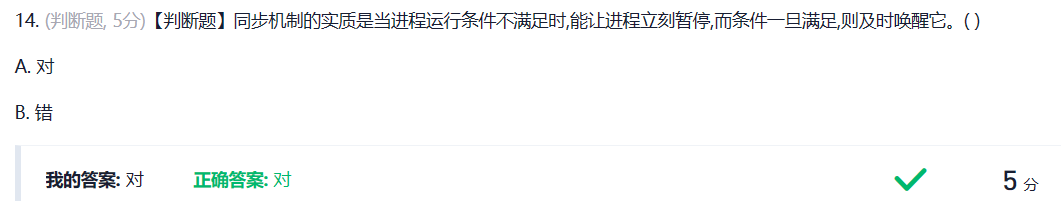
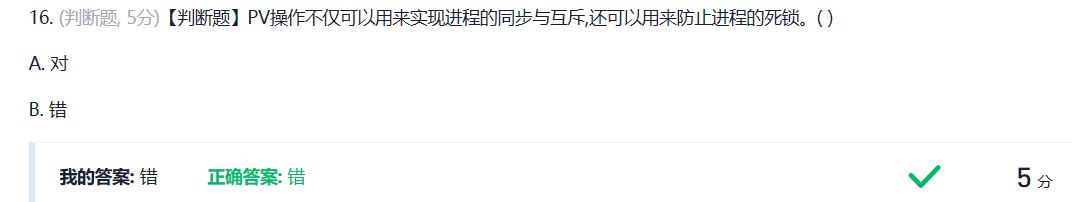




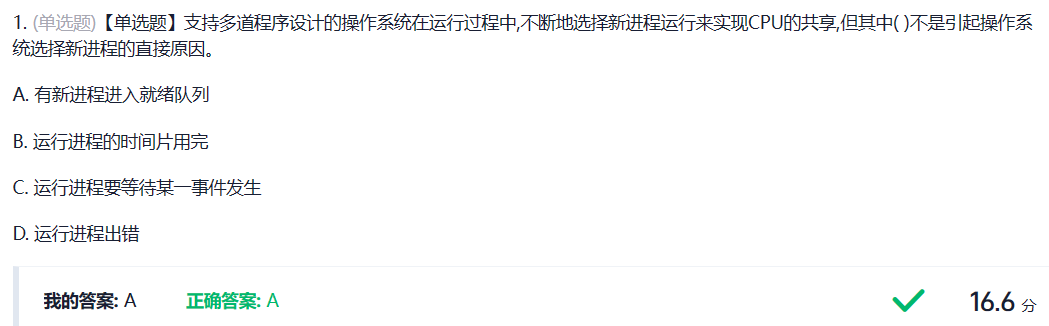


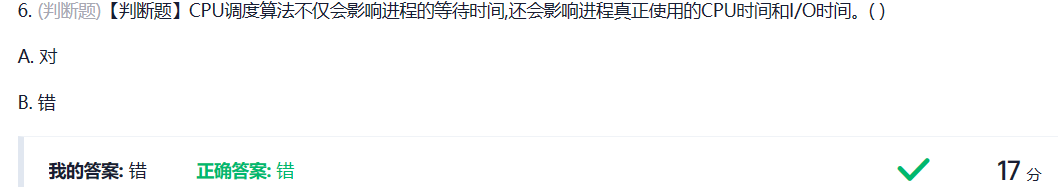




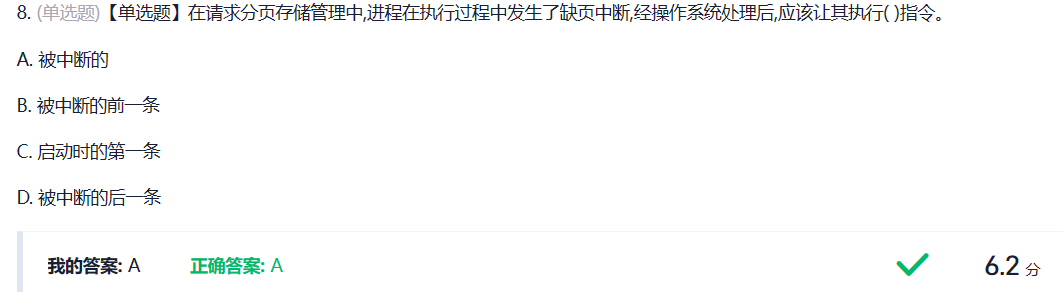


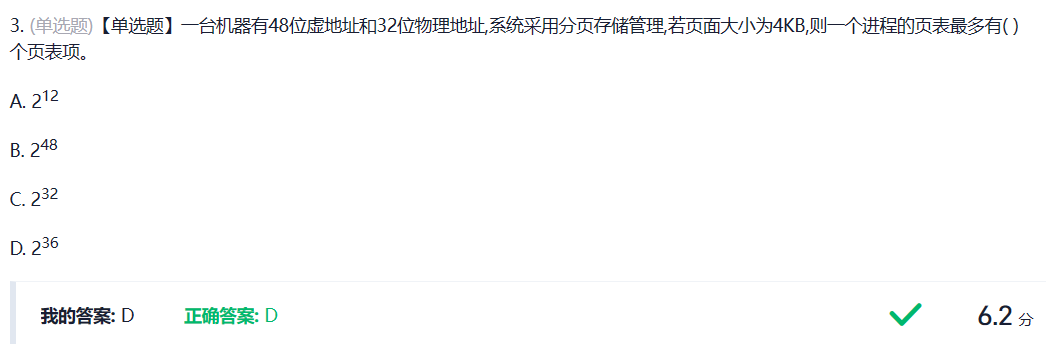
第三章：

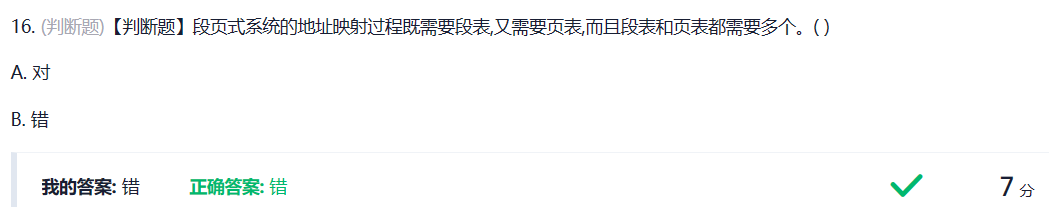


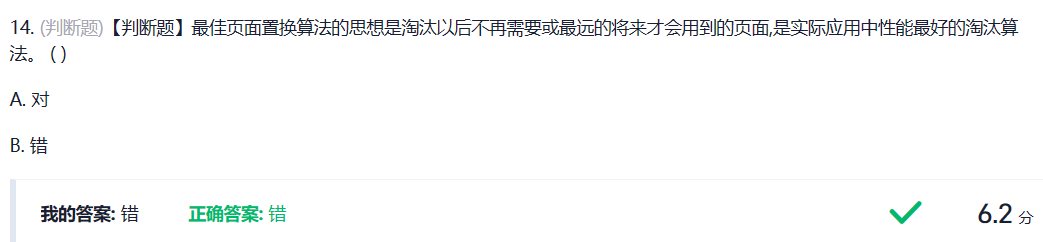


第四~五章









第六章

