判断5X2、单选5X2、填空5X2、名词解释5X4、解答5X4、算法2X15

多道程序特点：

内存中存放多道相互独立的程序、多个程序交替占用CPU、并发运行（宏观上并行，微观上串行）。

大大提高CPU利用率。

操作系统主要功能：

处理机管理、设备管理、存储管理、文件管理

用户接口分为：

普通用户（操作级接口，如图形用户接口）、程序开发人员（程序级接口，如系统调用和库函数）

操作系统的设计目标：

可靠性、可用性、安全性、性能。

C程序内存管理：

代码区：存放C函数编译后代码

数据区：字符串变量、全局变量、静态变量

栈区：函数的参数、返回值、局部变量

堆区：动态开辟的存储空间

程序的顺序执行：

具有独立功能的程序独占CPU直到得到最终结果的过程

顺序执行的特点：

顺序性、封闭性、可再现性

引入并发执行的目的：

提高系统的资源使用率

并行和并发的区别：

多个程序在同一时刻，同时在执行；同一时刻只有程序占用处理资源，但是多个程序交替使用资源，从宏观上，显现出类似并行执行的特征。

并发执行的特征：

间断性（异步性）、失去封闭性、失去可再现性

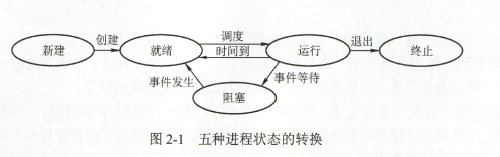
进程的定义：

一个具有独立功能的程序在一个数据集合上的一次动态执行过程。进程是程序的一次执行活动。进程是系统资源分配的基本单位和CPU调度的基本单位。

进程的特征：

动态性、独立性、并发性、异步性、结构化（进程=代码+数据+PCB）。

进程状态转换：



PCB：

进程控制块，PCB是系统感知进程存在的唯一标志。

可能造成进程挂起的原因：

换入换出、用户交互需求、时间驱动、父进程请求

原语：

由若干条指令构成的“原子操作(atomic operation)”过程，作为一个整体而不可分割——要么全都完成，要么全都不做。

线程分为：

内核级线程和用户级线程。

内核级线程的优缺点：

优点：一个线程阻塞，其它线程仍可执行；可以利用多处理机优势；操作系统内核本身也可实现为多线程。

缺点：线程切换开销大。

用户级线程的优缺点：

优点：线程切换无须内核介入，减少了切换开销；调度策略“因应用而异”；与操作系统关联少，所以可移植性好。

缺点：一个线程阻塞，将导致所有线程阻塞；纯ULT中，如果一个进程只能分配一个处理机，那么ULT无法利用“多处理机”带来的优势。

进程的同步和互斥：

同步：直接制约；互斥：间接制约

临界区：

它指程序中的一段特定的区域，在这个区域里，程序可能访问公共数据。

临界区使用准则：

空闲让进、忙则等待、有限等待、让权等待。

临界资源：

临界资源是一次仅允许一个进程使用的共享资源。各进程采取互斥访问的方式，实现资源共享。

信号量：

本质上是一个特殊的变量，是一个整型值。其特殊性在于，它的值只能被“初始化”和P、V两个原语操作所修改，用作控制进程的互斥与同步。

**PV大题：生产者——消费者问题或哲学家就餐问题或读者——写者问题**

进程间通信机制：

共享存储器、消息传递、管道、套接字。

管道：

管道是一条在进程间以字节流方式传送的通信通道，是单向的。

套接字：

套接字是在网络环境下，标志一个进程的机制。

死锁：

一组进程中，每个进程都无限等待被该组进程中另一进程所占有的、因而永远无法得到的资源，这种现象称为进程死锁。

死锁发生的四个条件（全部满足则一定发生死锁）：

互斥、请求和保持、不可抢占、循环等待。

解决死锁的思路：

死锁预防（打破四个条件中的一个）、死锁避免（银行家算法）、死锁检测与恢复。

安全状态与死锁的关系：

安全状态一定不是死锁状态，死锁状态一定是不安全状态，但不安全状态不一定是死锁状态，

如果资源分配处理不当，系统可能从安全状态变为不安全状态。

**银行家算法：考一个大题**

死锁定理：

S为死锁状态的条件是：当且仅当S状态的资源分配图是不可完全简化的。

作业调度：

周转时间：作业等待时间+作业执行时间。

平均周转时间：n个作业各自周转时间之和/n。

带权周转时间：周转时间/执行时间。

平均带权周转时间：n个作业各自的带权周转时间之和/n。

进程调度的时机：

正在执行的进程执行完毕；中断发生；进程的阻塞与唤醒；分时系统中时间片用完；系统调用执行完毕，从系统程序返回到用户程序时，进行进程调度；可抢占方式下，就绪队列中某进程优先级高于执行进程。

一道简答题：简述先来先服务、短作业优先、最高响应比优先、优先级调度、时间片轮转、多级反馈队列算法。

先来先服务：按照作业提交或进程变为就绪状态的先后次序分派CPU，当前作业或进程占用CPU，直到执行完或阻塞，才出让CPU（非抢占方式）。

短作业优先：对预计执行时间短的作业（进程）优先分派处理机。后来的短作业不抢先正在执行的作业。

最高响应比优先：响应比=（作业执行时间+作业等待时间）/作业执行时间。每次要执行新的作业前重新计算所有就绪队列中作业的响应比，先执行响应比最高的作业。后来的作业不抢占正在执行的作业。

优先级调度：分为非抢占式优先级和抢占式优先级。均是优先执行优先级高的进程，不同的是抢占式的优先级算法中后来的高优先权进程不用等当前进程执行完即可抢占CPU。

时间片轮转：将系统中所有的就绪进程按照FCFS原则，排成一个队列，每次调度时将CPU分派给队首进程，让其执行一个时间片，时间到后时钟中断触发时间片切换，该作业放到就绪队列队尾，CPU重新执行队列中现在的首进程。

多级反馈队列：根据作业或进程的性质或类型的不同，将就绪队列再分为若干个子队列，每个作业固定归入一个队列，不同队列可有不同的优先级、时间片长度、调度策略等。

内存的装入模块在装入内存时，有三种方式：

绝对装入：逻辑地址与物理地址完全相同；

静态重定位：在程序执行之前由装配程序完成地址映射，无需硬件支持。

动态重定位：在程序运行过程中要访问数据时再进行地址变换，需要硬件的支持。

程序的链接：

静态链接：在程序执行之前，由链接装配程序把它们链接成一个可运行的目标程序，并在程序运行前都装入内存。

装入时动态链接：边装入边链接。装入一个模块时，若发生外部模块调用，装入程序将去找出相应的外部模块并装入。

运行时动态链接：边执行边链接。在执行过程中，若发现一个被调用模块尚未装入内存时，由OS去找该模块并将其装入内存，链接到调用者模块上。

连续分配管理方式：

单一连续分配（有内碎片）；固定分区分配（有内碎片）；动态分区分配（有外碎片，可通过紧凑解决）。

动态分区的分配算法：首次适应、最佳适应、最坏适应、循环首次适应。

**由逻辑地址变换到物理地址会有计算题**

非连续分配管理方式：

基本分页（有内碎片）、基本分段（有外碎片）、段页式（有内部碎片）。

内碎片：已经被分配出去（能明确指出属于哪个进程）却不能被该进程利用的内存空间；

外碎片：还没有被分配出去（不属于任何进程），但由于太小了无法分配给申请内存空间的新进程的内存空闲区域。

虚拟存储器基本原理：

在程序装入时，不必将其全部读入到内存，而只需将当前需要执行的部分页或段读入到内存，就可让程序开始执行;在程序执行过程中，如果需执行的指令或访问的数据尚未在内存（称为缺页或缺段），则由处理器通知操作系统将相应的页或段调入到内存，然后继续执行程序。

抖动：

进程运行中，刚刚被淘汰的页很快被重新调入，调入不久又被再次淘汰，如此反复，以致大部分机器时间都用于页面调度，只有小部分时间用于进程的实际运行。

TLB：高速联想寄存器，本质上是地址转换硬件中的一个小型存储器，用于缓存段表和页表，从而减少对内存的访问次数，加速地址变换的过程，又叫快表。

页面的置换算法：

最佳算法（OPT）、最近最久未使用算法（LRU）、先进先出算法（FIFO）、时钟算法（CLOCK）。

文件的逻辑结构：

有结构文件（顺序文件、索引顺序文件、索引文件）、无结构文件、累积文件。

外存的分配方式：

连续分配、隐式链接分配、显式链接分配（FAT）、索引分配。

外存空闲空间的管理方式：

空闲表法、空闲链表法、位示图法、成组链接法。

I/O管理的目的：

提高I/O访问效率，匹配CPU和多种不同处理速度的外设；方便用户使用，对不同类型的设备使用统一方法，协调对设备的并发使用；方便OS内部对设备的控制，增加和删除设备，适应新的设备类型。

I/O控制方式：

程序I/O、程序中断、DMA、通道。

缓冲技术引入：

缓和CPU与I/O设备之间速度不匹配的矛盾；减少对CPU的中断频率，放宽对CPU中断响应时间的限制；提高CPU和I/O设备之间的并行性。

缓冲技术分类：

单缓冲、双缓冲、环形缓冲、缓冲池。

设备独立性：

即应用程序独立于具体使用的物理设备。

SPOOLing技术：

假脱机I/O技术。实质上就是对I/O操作进行批处理。SPOOLing 技术是一种以空间换时间的技术。

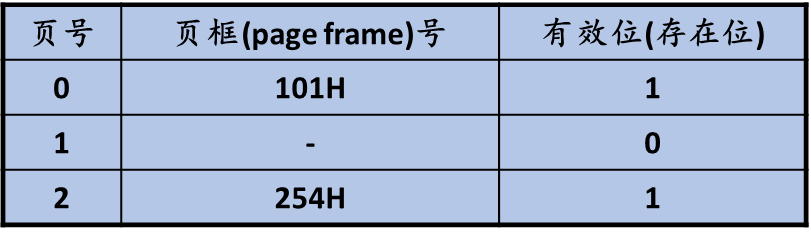
**磁盘调度算法，有一道简答。**

先来先服务：根据进程请求磁盘I/O的次序进行调度。

最短寻道时间优先（SSTF）：要求访问的磁道与当前磁头所在的磁道距离最近，可能导致饥饿。

电梯算法（SCAN）：磁头已被规定下一个访问的方向，依序访问距离最近的磁道，并在访问到最后一个磁道后折返向另一个方向上的磁道依序访问距离最近磁道。

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为108ns（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，①TLB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表示页表不在内存，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，各需要多少时间，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。

解答

页面大小为4KB，所以2362H、1565H、25A5H对应的页号分别为2、1、2， 根据页表内容，2362H、25A5H在内存， 1565H不在内存。初始快表为空，故2362H需装入快表， 25A5H直接访问快表， 1565H需装入内存，再装入快表。据此：

2362H：10(访TLB)+100(访页表)+100(访主存单元)=210ns

1565H：10 (访TLB)+100 (访页表)+108 (调页)+10 (访TLB) +100(访主存单元)=100000220ns

25A5H： 10(访TLB)+100(访主存单元)=110ns

根据页表，第0、2页在内存中，由于第2页刚刚被访问（访问2362H），根据LRU算法，将第0页替换，因此对应页框号为101H，故1565H的物理地址为101565H。