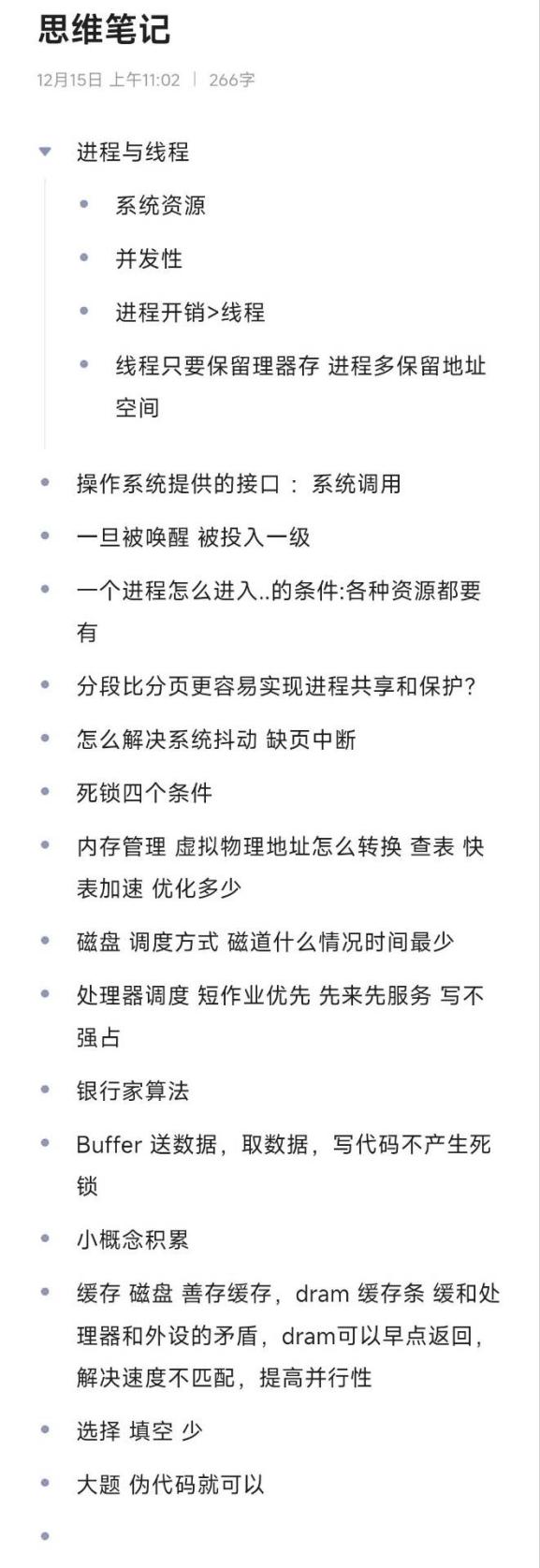
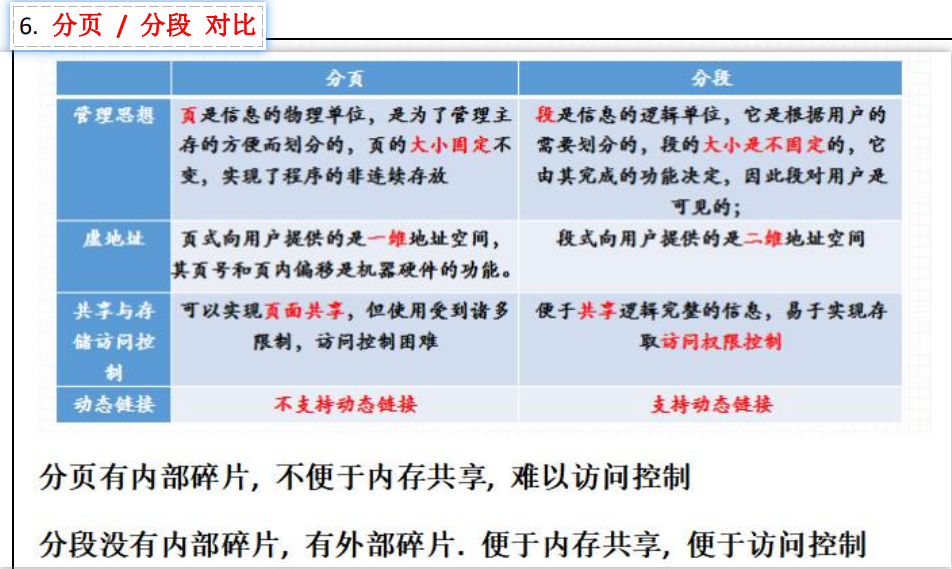
2024秋操作系统回忆版考题：

计算题（5道）：分页和快表（有效访问时间，即没有命中快表访问内存两次，命中快表访问一次快表和一次内存）、磁盘调度（总共多少磁道、访问顺序）、进程调度（STJ、FCFS）（平均周转时间）、银行家算法（求Need矩阵、是否是安全状态、request银行家算法）、pv操作（wait和signal）  
简答题：进程和线程区别，为什么分段比分页更方便共享内存巴拉巴拉，死锁的四个条件，openeuler增强特性（见本文档图片），什么是系统抖动以及产生原因以及解决策略  
选择填空多看ppt吧，选择有道考的Loongarch32（见本文档图片）

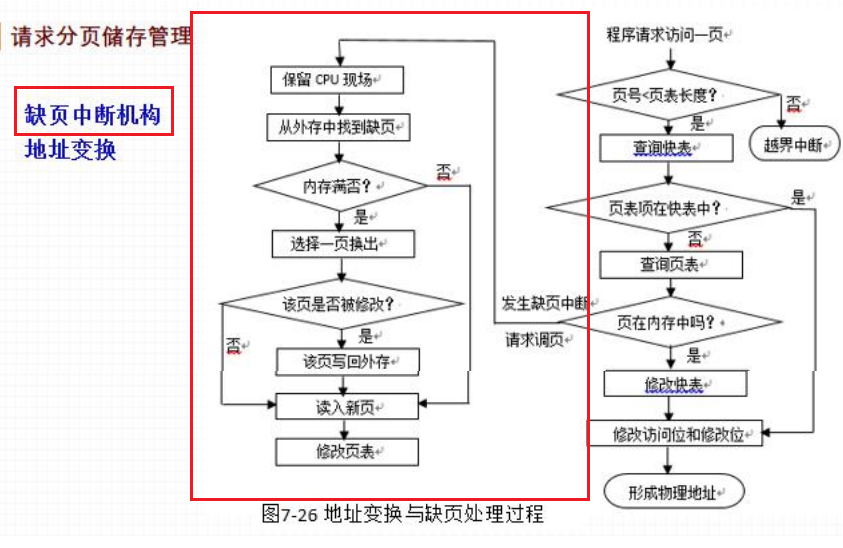
suosu

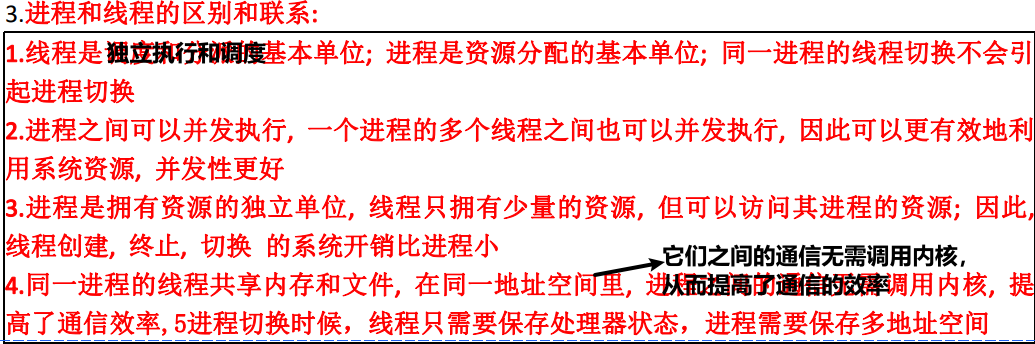


相同点：都是两次访问内存，支持TLB，存在于内存中

页是信息的物理单位，页的大小固定，对用户不可见，一维地址空间，内部碎片不超过一页，可以实现页面共享但控制困难；不支持动态链接

段是信息的逻辑单位，段的大小不固定，对用户可见，二维地址空间，外部碎片，便于共享逻辑完整的信息，易于实现存取访问权限控制；支持动态链接





6.线程共享本进程的资源，不利于资源的管理和保护，而进程之间的资源是独立的，能很好的进行资源管理和保护

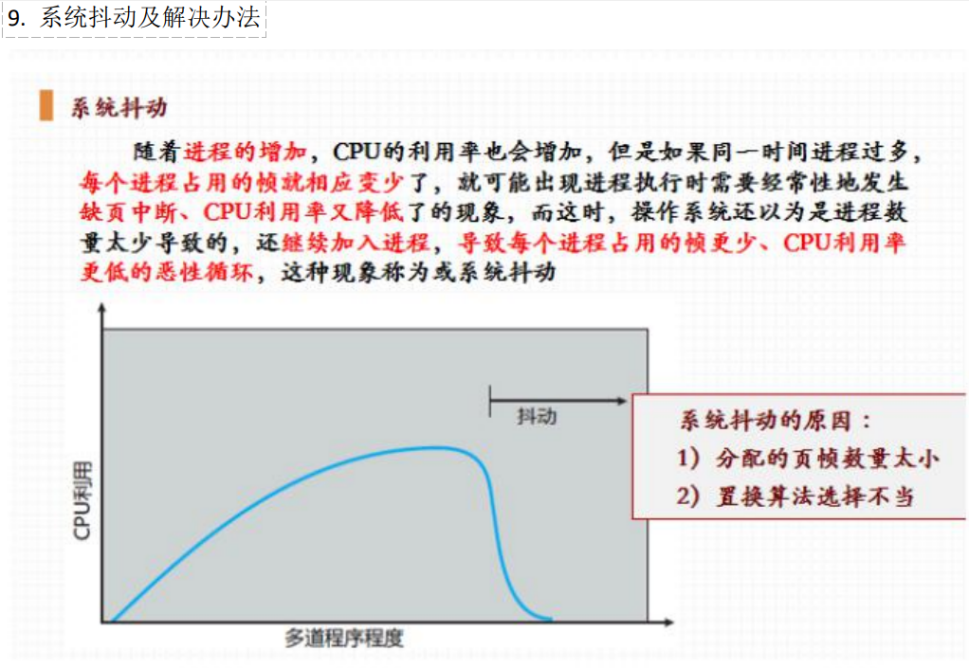
**创建原语需要完成以下步骤：**

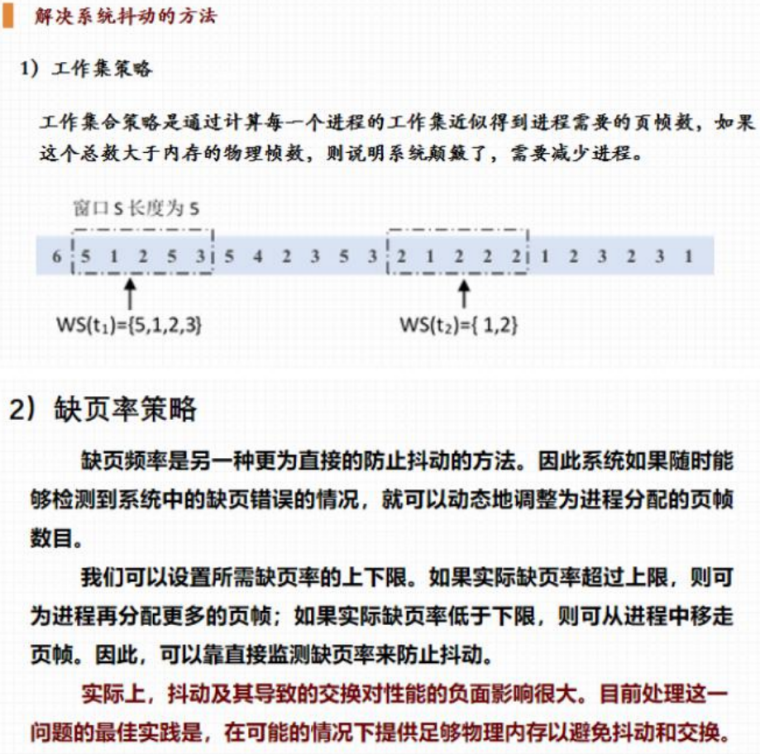
**（1）给新进程分配一个唯一的进程标识符，并申请一个空白的PCB(PCB是有限的)。若PCB申请失败则创建失败。**

**（2）给进程分配空间分配包括程序、数据、用户栈等。**

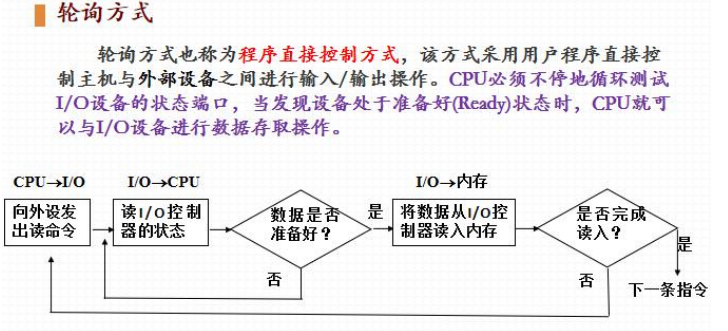
**（3）将新进程插入就绪队列和进程隶属关系族群中。**

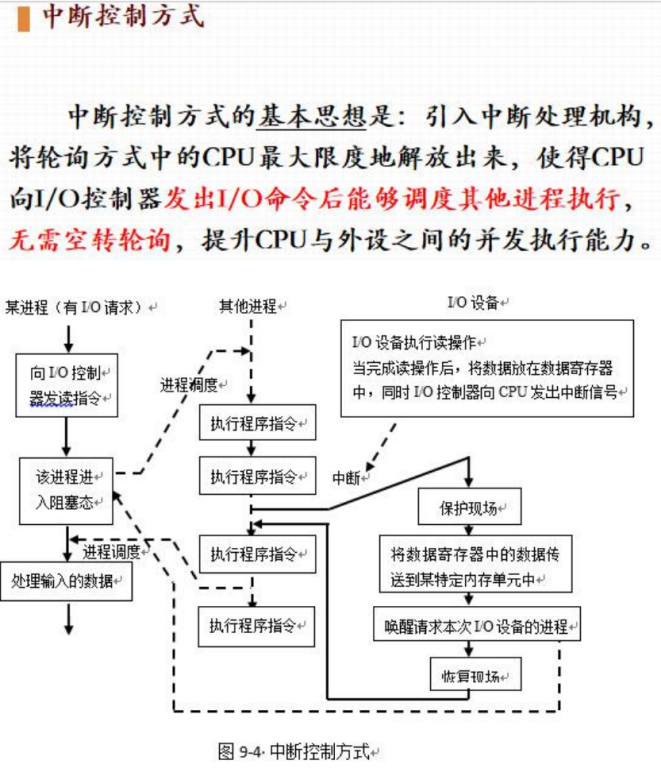
**（4）创建或扩充其它数据结构，如为进程创建记账文件等。**

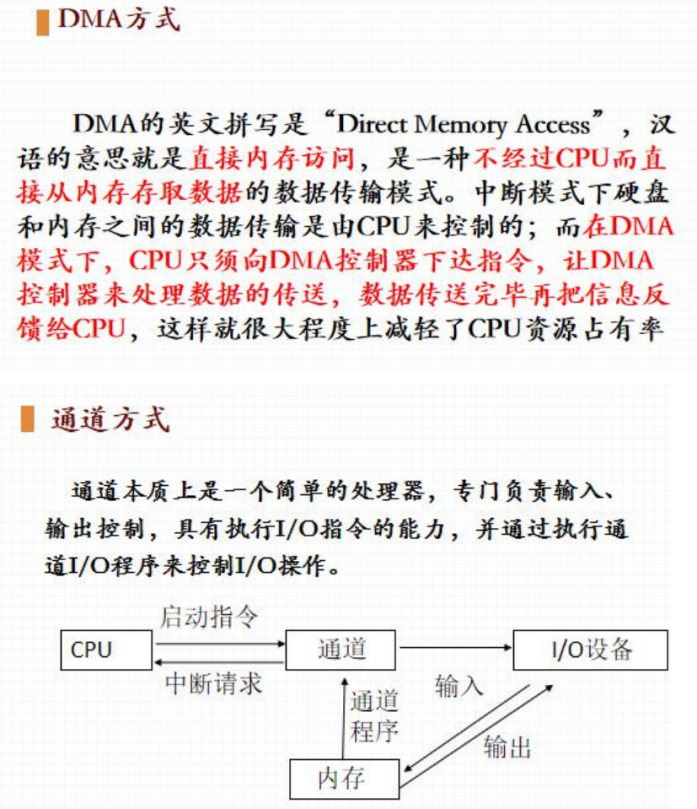




**设备数据(I/O)传输控制方法(轮询、中断、DMA、通道）**









**缓冲区的引入**

（1）缓解设备之间速度差异的矛盾。

（2）缓冲区可以缓解设备之间传输数据大小不一致的矛盾

（3）支持应用程序I/O的复制语义

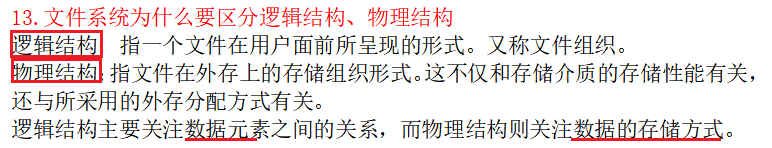
**文件分类**

按逻辑结构划分：有结构（栈结构、顺序结构、散列结构）文件、无结构文件

按物理结构划分：顺序文件、链接文件、索引文件

文件的读写方式：顺序读写、直接读写/随机读写、索引读写

按存取控制划分：可执行文件、只读文件、可读写文件



**文件目录的作用和文件目录结构**

作用：文件名到文件物理位置的映射-按名存取、快速检索文件目录、文件共享和允许文件重名。

组织结构：单级目录(查找费时、重名问题、不利于文件共享)，两级目录结构和多级目录结构。

**自底向上的文件结构组成**：（1）数据项（2）记录。（3）文件

**文件目录管理的基本功能**包括：(1) 实现“按名存取”。 (2) 快速检索文件目录。

(3)实现文件共享。(4) 允许文件重名。

支持互斥，系统要满足：**互斥、有限等待、空闲让进**

**解决实现互斥：中断禁用、专用机器指令、信号量、管程**

**高级通信机制：共享内存/管道通信/消息传递**

**微内核功能**包括低级存储管理/进程间通信/IO和中断管理

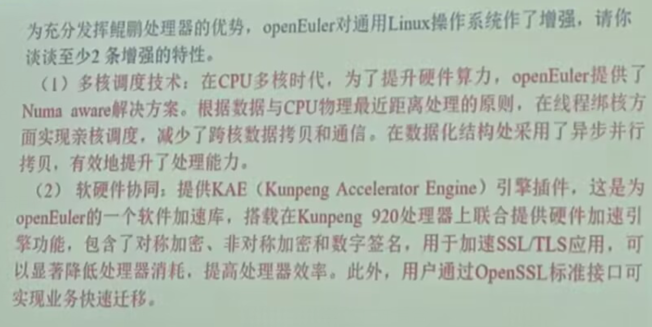
系统调用向系统传递参数——通过寄存器/内存/堆栈

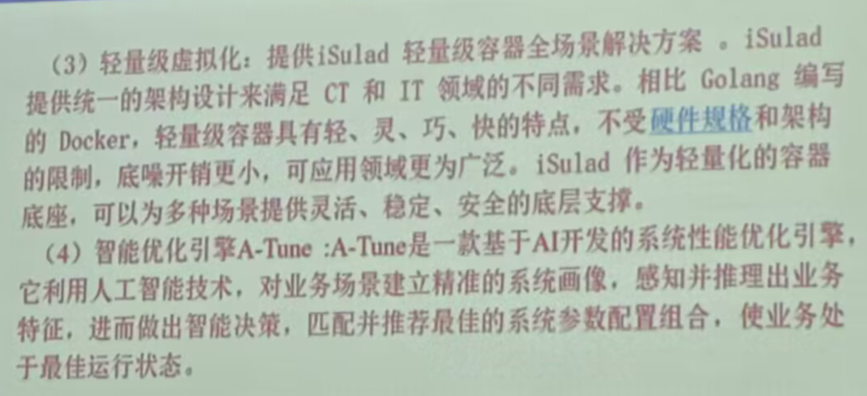
**I/O软件设计的原则**:效率和通用性

**设备分配过程**1．分配设备2．分配控制器3．分配通道

Openeuler

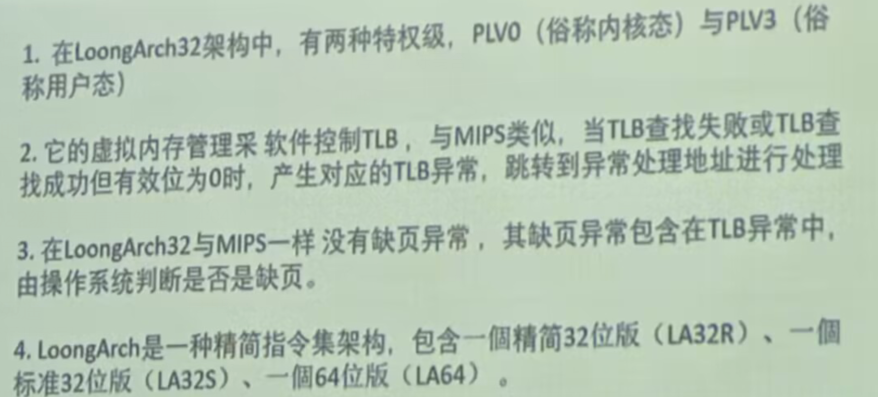
多核调度技术、软硬件协同、轻量级虚拟化、智能优化引擎A-Tune



**在CPU调度中，openEuler为什么没有采用最短进程优先（Shortest Job First, SJF）算法？**(1) SJF算法会导致运行时间长的进程响应时间越来越长，从而产生进程饥饿现象；(2) 因为很难预估每个进程的运行时间，SJF算法在实现上存在难点。

Loongarch32：

特权级：PLV0（内核态）、PLV3（用户态）

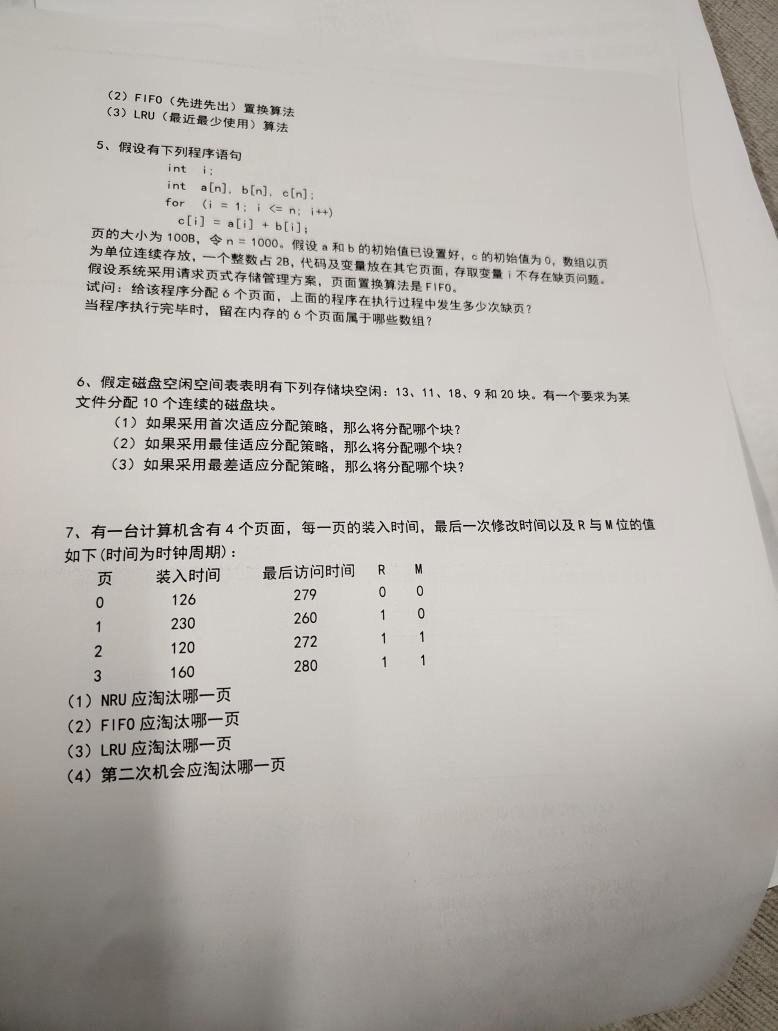
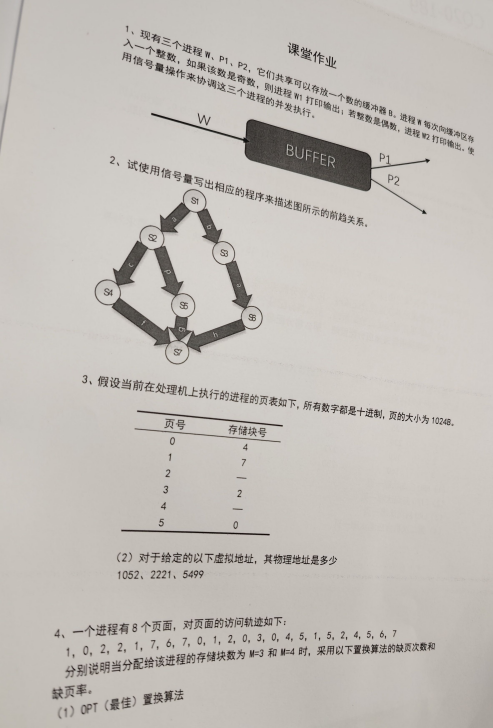


**什么是进程互斥和同步？解决进程互斥的方法有哪些？**

·进程互斥是进程之间的间接制约关系，进程互斥就是保证每次只有一个进程使用临界资源。

·进程同步是进程之间共同完成一项任务时直接发生的相互作用关系，代表了进程之间的执行次序。

方法：中断禁用、专用机器指令，信号量机制，管程等.



P2

操作系统主要功能：处理器管理，存储器管理，文件管理，设备管理，用户接口

基本操作系统类型：单道批处理系统、多道批处理系统、分时系统、实时系统

其它类型操作系统：微机操作系统、网络操作系统（unix，windows）、分布式操作系统、嵌入式操作系统

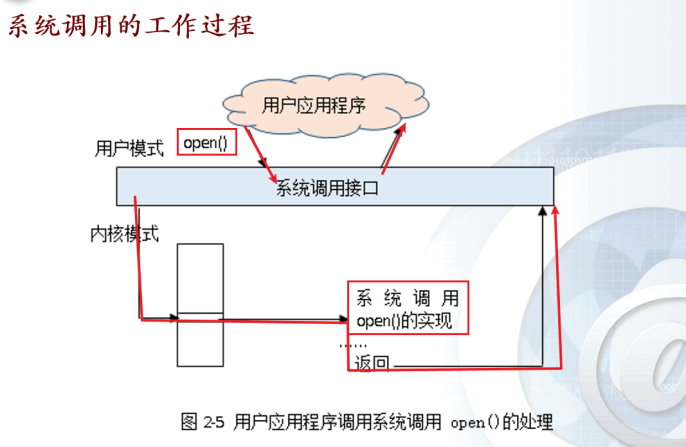
Os调用：

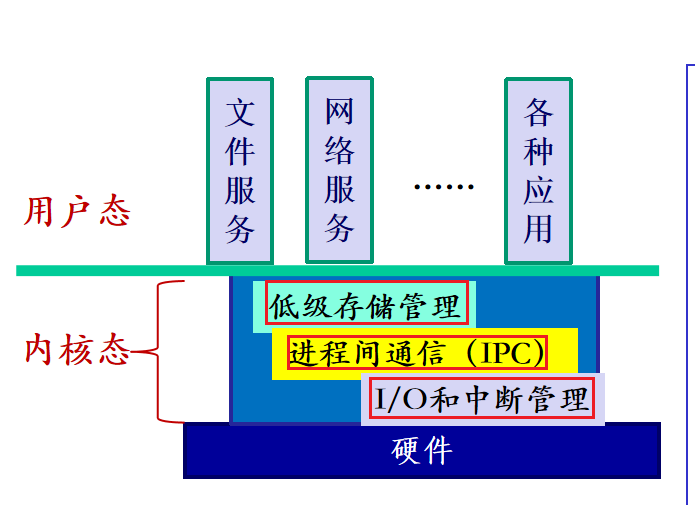
命令接口（命令行界面和图形化界面）和系统调用

内核提供一系列具备预定功能的多内核函数，通过一组称为系统调用（system call)的接口呈现给用户。系统调用把应用程序的请求传给内核，调用相应的内核函数完成所需的处理，将处理结果返回给应用程序。

API（Application Programming Interface，API） 是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码。

**API 函数**通常为应用程序员调用实际的**系统调用**

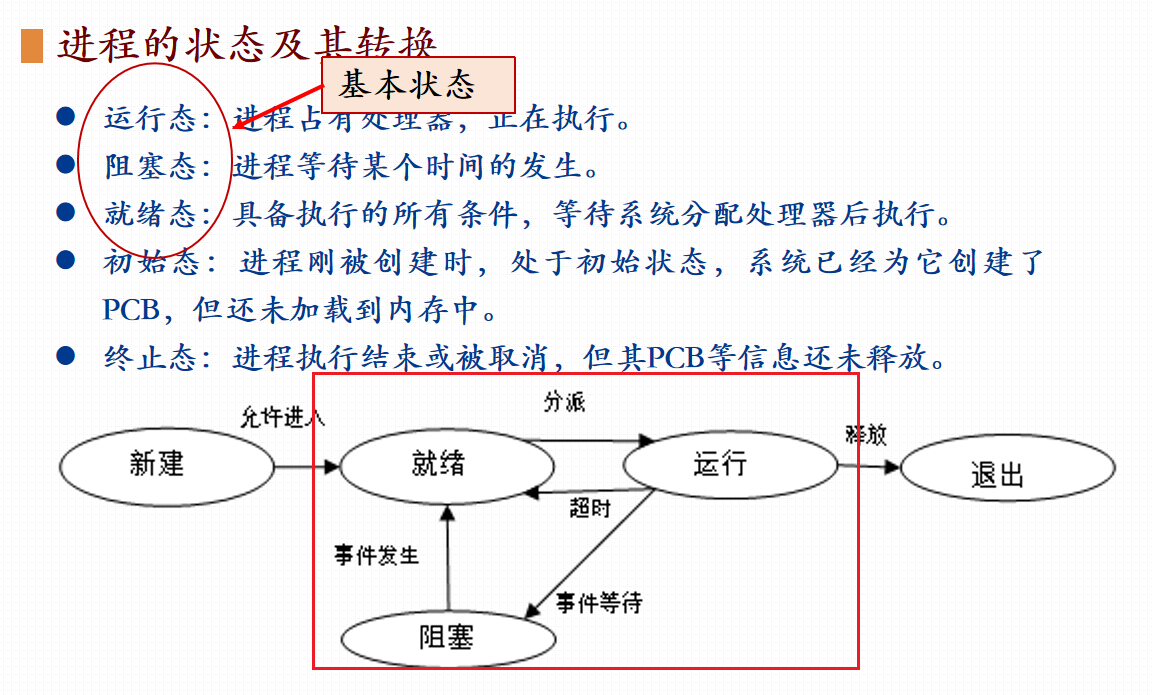




进程有以下四个特征：

动态性、并发性、独立性、异步性

进程是系统分配资源和调度的基本单位 进程控制块PCB



**进程切换**引起的事件有：

中断：时钟中断（基于时间片的调度）和I/O中断。

陷阱：与当前运行进程所产生的错误或异常有关。

系统调用：切换至操作系统进程，请求系统调用的进程转为阻塞态。

**线程是一个可以独立执行和调度的基本单位**

