定义操作系统：一直运行在计算机上的程序（通常称为内核），其他程序则为系统程序和应用程序。

操作系统的作用：

系统角度：

1. 控制程序
2. 分配资源

用户角度：

1. 提供更加简单方便高效的使用
2. 最大化资源利用率
3. 给用户提供界面
4. 实现共享资源的交换与协调

三个主要作用：控制程序，分配资源，提供用户与计算机硬件的接口（便于使用）

操作系统四大特性：

并发性，虚拟性，共享性，异步性

当打开电源或重启时，计算机开始运行，它需要一个初始化程序（引导程序）

多道程序（multi-programming）：通过组织作业（编码或数据）使CPU总有一个作业可执行。 => 提高了CPU的利用率

多任务（multi-tasking，分时系统）是多道程序的延伸。在分时系统中，CPU还是通过在作业之间的切换来执行多个作业，由于切换频率很高，用户可以在程序运行期间与之进行交互。

操作系统的经典的结构模型：层次模型

三种最基本的操作系统类型：批处理系统，分时系统，实时系统

存储层次：寄存器->高效缓存->内存->SSD->硬盘->光盘 （设计多种存储层次）

缓存思想：一些数据平时保存在慢速设备上，但是使用时需要快速读写。这些数据在使用时被临时复制到更快的存储系统中。需要信息时，先检查缓存，如果不在，再检查源地址。热数据的访问速度变快。

系统调用：是指由操作系统为用户程序提供的系统服务接口。

系统调用与API的逻辑关系：用户程序调用API，API调用系统调用来完成系统操作。

Dual Mode：操作系统提供用户模式和内核模式，特权指令只能在内核模式执行。硬件提供模式位。

原因：错误的用户程序可能会损坏操作系统。

操作系统需要提供哪些服务：

用户界面，I/O操作，程序执行，通信，文件系统操作 等

脱机I/O

目的：解决人机矛盾以及CPU与I/O设备之间速度不匹配的矛盾

方法：利用低速的外围机进行，纸带->磁带

2阶段调度：

作业调度：决定将哪几个作业调入内存，并为作业创建进程，分配资源；

进程调度：从内存的作业池中选取一个（或几个）进程分配CPU进行处理

现代操作系统大多是什么驱动的？

中断驱动

在哪些情况下，计算机系统运行在管理模式？

中断，异常，系统调用，启动过程中

用户程序可能会陷入死循环。为保证对计算机系统的掌控能力，操作系统利用了硬件的哪种保护机制，如何使用的？

CPU保护机制，使用定时器（timer），计时结束后产生中断，将控制权交给操作系统

单片结构：

所有功能集于一层

优点：性能好；

缺点：实现，维护困难

层次化结构：

操作系统分成若干层，每层只能调用更低层功能

优点：简化构造与调试

缺点：难以合理定义各层，性能差

模块化结构：

内核提供核心服务，其他服务在内核运行时动态实现

优点：比分层系统更灵活，比微内核效率高

缺点：接口难以满足实际需求，无法找到一个可靠的模块开发顺序

微内核结构：

从内核中删除所有不必要的部件

优点：便于扩展/移植。安全性，可靠性好

缺点：性能受损

机制与策略分离的设计原则

用户的需求，特点各式各样，一套机制不能满足所有人的需要

机制决定如何做，策略决定做什么

提高了灵活性

Fork（）函数虽然会向子进程复制父进程原来的内存，但两个进程操作的内存是不共享的，子进程改变数组，不会影响父进程。

终止状态的存在是一种子进程向父进程反馈信息的一种机制。子进程执行代码结束并退出后，父进程应当读取子进程的退出状态（运行结果，通过子进程的PCB）

如果子进程异常终止，终止状态下向父进程发送的反馈信息会使父进程对异常作出应对

僵尸进程：

一个进程在已经结束自己生命，但其父进程还未对其回收完毕时成为僵尸进程

僵尸进程不再被调度，所有的用户空间内存与大部分已分配的内核内存都被清理，仅在进程列表中保留一个位置（PCB）。记载该进程的退出状态等信息，供其他进程收集。

进程相关数据：

用户空间：代码段，数据段，堆，栈

内核空间：PCB（进程维护相关信息）

PCB：

进程状态

PC

CPU调度信息（优先级等）

寄存器值

进程内存管理信息

IO状态信息

内核代码

多线程优点：资源共享，响应度高，扩展性高，经济性好

线程间共享：堆，全局变量，代码段

Pthread\_attr\_init()

Pthread\_creat() 创建线程，里面传入要执行的函数runner（）

Pthread\_join() 等待线程的完成

在runner() 最后加上pthread\_exit() 表示线程结束

临界区问题 满足三个条件：

互斥，进步，有限等待

IPC（inter-process communication）进程间通信

严格轮换不满足进步，当进程0想要进入临界区两次，且此时进程1并不要进入临界区时，会因为被进程1阻塞而无法实现。

死锁：

两个或多个进程无限等待一个事件，而这个事件只能由等待的进程来产生，当出现这样的状态时，这些进程就成为死锁。

死锁要求：互斥，非抢占，循环等待，占有并等待

信号量：是一个整型变量，只能通过两个标准原子操作被访问，表示所空闲的资源的数量。

功能：互斥，计数，同步

调度准则：

最大化

CPU使用率：需要使CPU尽可能忙

吞吐量：指一个时间单位内所完成进程的数量

最小化

周转时间：从进程提交到进程完成的时间

等待时间：在就绪队列中等待所花费时间之和

响应时间：从提交请求到产生第一响应（开始响应）的时间。

FCFS：

SJF最短作业优先调度，分为抢占与非抢占，抢占SJF叫最短剩余时间优先调度：

优先级调度：具有最高优先级的进程会分配到CPU。具有相同优先级的进程按FCFS顺序调度。Number 1

RR（轮转法）：定义一个较小时间单元，称为时间片，将就绪队列作为循环队列。

周期进程，包括周期p和执行时间t

单调速率调度：采用抢占的，静态优先级的策略，调度周期性任务。当较低优先级的进程正在运行并且较高优先级的进程可以运行时，较高优先级进程将会抢占较低优先级。在进入系统时，每个周期性任务会分配一个优先级，它与其周期p成反比。

最早截止期限优先调度：根据截止期限动态分配优先级。截止期限越早，优先级越高，反之优先级越低

Segmentation fault：非法访问内存块时，发生段错误

TLB：快表or页表缓冲，是一个内存管理单元用于改进虚拟地址到物理地址转换速度的缓存，里面存放的是一些页表文件，文件记录了虚拟地址到物理地址的转换表。如果没有TLB，则每次取数据都需要两次访问内存，即查页表获得物理地址和取数据

Page fault：CPU访问内存时，访问的page不在内存中时发生page fault

Demanding paging：按需调页，进程申请内存时只为其分配虚拟地址，不分配物理地址，当进程发生页面访问时，才通过page fault为其分配真正的物理页面。

Thrashing：产生频繁的换入换出，页面置换花费的时间比实际执行的时间还要长。

情形：为进程分配的页框太少，导致刚刚换出的页面又要被访问

LRU：最近最少使用

FIFO

Optimal replacement：知道次序，把最远的替换出去

Belady’s anomaly：对于有些页面置换算法（如先进先出），随着分配页帧数量的增加，缺页错误率会增加的异常结果。

Stack算法：堆栈算法：帧数为n的内存页面集合是帧数为n+1的内存页面集合的子集，如LRU算法和最优置换算法。

局部性原理：

指cpu访问存储器时，无论是存取指令还是存取数据，所访问的存储单元都趋于集中在一个较小的连续区域中。

饿死现象：当一组进程到达时,CPU根据算法进行进程调度。有的进程因此而需要等待,而不能及时得到资源,这就叫饥饿。进程得到资源时,再完成已经不再具有意义,或者永远在等待，这就叫做饿死

RMW a1，ap

RCW a2，a3，a4

 rmw 与 rcw 相比，取数值较小的校验算法，如果相等用rcw

FCFS

SSTF

SCAN（单方向C-SCAN）

LOOK（单方向C-LOOK）

I/O控制方式：轮询，中断，直接内存访问DMA



BSS：未初始化数据区 节省空间

打开文件表：操作系统为每个进程维护一个打开文件表，文件表里的每一项代表「文件描述符」，所以说文件描述符是打开文件表项的标识

在打开文件表中维护着打开文件的状态和信息