什么是操作系统：

1. 计算机**核心程序（kernel program）**
2. 用户**接口程序（interface program）**
3. 资源分配与回收的**控制程序（control program）**

操作系统是位于Hardware与User之间的**程序集合**

基本目标：方便性与高效性

工作原理：中断驱动

存储结构：内存，二级存储，分成结构

硬件保护：双模式，特权指令，内存，IO，CPU

服务：程序，IO，文件，进程，通信

**硬件（Hardware）：**

启动：执行引导程序，载入operation program kernel（ROM/EPROM中）

中断：告知操作系统执行中断处理程序/策略（interrupt/trap）

Interrupt driven：中断-挂起-中断处理-中断返回

通用中断功能：Interrupt service routine（由**中断向量**保存）、save address（程序恢复）、disable interrupt（进程同步）

Memory： 寄存器-main memory（内存）-electronic disk（U盘）-magnetic disk（磁盘）-optical disk

内存： CPU可以**直接访问**的唯一的大型存储介质

IO设备：控制器**local buffer**和**外设**交互的过程（I/O过程无需CPU干预，中断响应方式）

系统保护：Dual-Mode（用户，内核态）、CPU（进程时间控制）、内存（程序地址控制）、IO保护（全privilege）

**Privileged Instruction**特权指令：可能对**硬件**造成伤害的指令，在内核态执行

**发展(Develop)：**

单任务系统、批处理系统（批处理文件）、多道程序系统（提高CPU利用率，发挥程序**并发性**，内存，CPU，任务，IO资源控制决策）、分时系统（多用户共享，CPU调度，virtual memory）

并行：多CPU同时进行 并发：单CPU资源调配（时间段）

操作系统特征：**并发**（Concurrence），**共享**（Sharing），**虚拟**（Virtual），**异步**（Asynchronism）

CPU利用率：CPU**非空闲时间**占比

Desktop System：单处理器（Single-Processor），多处理器（Multiprocessor），紧耦合（Tightly coupled），分布式（network云端，shared storage集群系统（hot-standby非对称，对称）），实时（时间约束），手势终端（Pocket-PC，Cellular telephone，资源少）系统

优点：高吞吐量（Throughput），资源共享，可靠性高

**功能（Functions）：**

系统内部具有的能力

进程（运行中的程序），内存，磁盘（Secondary-Storage），文件（文件系统），IO管理

**服务（Services）：**

对外可以提供的服务

程序执行，IO操作，文件系统操作（File-system manipulation），进程通信，错误监控

系统执行：资源分配，审计（Accounting安全追踪，系统优化），保护

**接口（interface）：**

命令行（Command-Line），批处理（Batch 配置命令文件.bat），图形用户界面（Graphical User）

Command-line Shell命令行解释器：解读命令

**S**ystem **C**alls **I**nterface系统调用（基本功能实现）：参数放置（寄存器，内存，栈）

**A**pplication **P**rogramming **I**nterface（API）：应用程序接口，调用OS层功能（run-time support library接口使用说明）

**结构（Structure）：**

**简单**（MS-DOS，UNIX 结构区分不明显），**分层**（分为多层 hardware…功能模块化…user layer），**微内核**（Microkernels仅保留进程，内存管理和通信功能 易于扩展和移植，user和kernel转换产生额外开销），**模块化**（面向对象的方法，灵活性高）00

000

Virtual machine：模拟硬件，为操作系统提供illusion hardware interface，虚拟机间资源不共享（JVM：解释Java，可移植性很高）

**设计（Design）：**

Policy（策略 做什么），Mechanism（机制 怎么做）

C，C++，汇编

***进程：***

按照顺序结构**执行中的程序，**程序决定进程执行顺序（非一一对应）

程序代码（Text section），程序计数器（Program Counter），指令寄存器（执行内容Processor Register），堆栈（堆：程序申请的空间 栈：操作系统分配Heap-stack），数据段（Data section）

**动态性**（Dynamic），**独立性**（Independency），**并发性**（Concurrence），**结构化**（Structure）

进程状态：New（初始）-Ready（准备，获得内存资源）-Running（获得CPU使用权）-Waiting（等待IO响应）-Terminated（exit终止）

CPU核心/用户态：执行kernel/user process

**P**rocess **C**ontrol **B**lock（PCB进程控制块）：链表方式连接进程，控制进程状态转换（program counter，pointer）

Job queues所有进程队列：Ready queue**就绪**（等待执行），Device queue**设备**（等待IO操作），其他队列（均在Main memory中）

Scheduler调度器：Long-term（内存加载选择，控制多道程序度（内存中存放进程个数）），Short-term（CPU执行进程选择），Medium-term（进程交换swapping 腾出内存空间）

IO进程（以IO操作为主），CPU进程（以CPU计算为主）通过调度器控制

Context switch上下文切换：PCB，CPU，内存等信息值，用于区分进程状态（在进程切换时保存context状态）

进程创建：父进程创建子进程形成进程树，并由Process ID标识每个进程

子进程通过getID()获得自己的ID号

进程终止：结束，挂起，随父进程终止（级联终止）

**I**nter-**P**rocess **C**ommunicate进程通信：共享内存（临界区问题，需互斥访问），消息传递（producer consumer）

Bounded-Buffer：通过循环队列控制进程的内存数据读写（可用空间**BUFFER\_SIZE-1**）

Message-passing：通信连接（物理介质+逻辑链接）