

绪论

操作系统是控制和管理计算机系统内各种硬件和软件资源、有效地组织多道程序运行的系统软件，是用户和计算机之间的接口。

定义，功能

操作系统是介于计算机和应用软件之间的一个软件系统，操作系统的上层和下层都有其他对象存在。

操作系统的目标：方便性、有效性（提高系统资源利用率、提高系统吞吐量）、可扩充性、开放性

操作系统的作用**

作为用户和计算机之间的接口，OS处于用户与计算机硬件系统之间，通过一下方式交互：

1. 命令方式
2. 脱机命令接口（Windows）= 批处理命令
3. 图形、窗口方式
4. 系统调用方式（直接给程序使用）

实现了对计算机资源的有效管理：处理机管理功能；文件管理；IO设备管理；内存管理。

特征

- 并发：区别并行和并发；引入进程
- 异步
- 共享：互斥共享方式；同时访问方式。并发和共享是多用户（多任务）OS的两个最基本的特征，互为存在的条件
- 虚拟：时分复用技术（虚拟处理机技术、虚拟设备技术、虚拟信道技术）；空分复用技术（虚拟存储器）

并发性：

并发性是两个或多个事件在同一时间间隔内发生（宏观上同时发生，微观上交替发生）；

并行性则是在同一时间间隔内发生（多处理器的支持）。

共享：

系统中的资源可供内存中的多个并发执行的进程（线程）共同使用。

1. **互斥共享方式：**为使输出结果不混淆，规定一段时间内只允许一个进程（线程）访问该资源。当当下进程访问完并释放资源后，才允许另一进程对该资源访问。在一段时间内只允许一个进程访问的资源称为临界资源或独占资源。
2. **同时访问方式：**以并发执行为条件允许一段时间内由多个进程“同时”对资源进行访问（宏观上同时，微观上交替对资源访问）。如果系统不能对资源共享实施有效管理，无法并

虚拟化：

1. **虚拟性：**把一个物理实体变为若干个逻辑上的对应物（虚拟）。
2. **虚拟存储技术：**物理存储器变为虚拟存储器，从逻辑上扩充存储器容量。

3. **虚拟设备技术**：一台物理IO设备虚拟为多台逻辑上的IO设备，并允许每个用户占用一台逻辑上的IO设备。
4. 把一条物理信道寻味多条逻辑信道（虚信道）

异步性：

多道程序环境下，允许多个进程并发执行。

发展与分类

发展：未配置操作系统的计算机系统 -> 单道批处理系统 -> 多道批处理系统 -> 分时系统 -> 实时系统

人工操作方式：

用户独占全机，CPU要等待人工操作。

后来引入脱机输入/输出方式（用外围机+磁盘实现），并有外围处理机负责控制作业的输入输出。

单道批处理系统：

单个处理器、后备作业队列、IO设备、监管程序 共同构成系统。

但内存中仅有一道程序运行，CPU有大量时间在空闲等待IO完成。

多道批处理系统：

多道性、无序性、调度性。

资源利用率高，系统吞吐量大；但平均周转时间长，无交互能力。

分时系统：

人机交互、共享主机、便于用户上机。多个用户同时输入命令，系统也应能全部地及时接收并处理。

用户请求可以被及时响应，解决人机交互问题，允许多个用户同时使用一台计算机，用户对计算机操作相互独立；

但不能优先处理紧急任务，对各个用户/作业都是完全公平的，循环地为每个用户/作业服务一个时间片，不区分任务的紧急性。

实时系统：

及时性、可靠性。

系统能及时（或即时）响应外部事件的请求，在规定时间内完成对该事件的处理，并控制所有史诗任务协调一致地运行。

可分类为：

1. 硬实时任务：系统必须满足任务对截止时间的要求，否则会出现难以预测的结果。
2. 软实时任务：对截止时间不太严格，偶尔错过也不会对系统产生太大影响影响。

操作系统的运行环境

运行机制、中断、系统调用。

运行机制

- CPU状态
 - 用户态（常态、目态）
 - 执行非特权指令
 - 内核态（核心态、特态、管态）：执行全部指令
 - 当一个任务执行系统调用而进入内核代码中执行时，进程处于内核态，执行管补指令

- 特权指令和非特权指令：
 - 特权指令只允许操作系统内核部分使用，不允许用户直接使用（包括IO指令、置终端屏蔽指令、清内存、见存储保护、设置时钟指令）。处于内核态运行的是内核程序，可以执行特权指令
 - 非特权指令所有程序均可使用。处于用户态运行的是应用程序，只能执行非特权指令
- 内核态与用户态的转换
 1. 系统调用：用户态进程主动要求切换为内核态的方式之一
 2. 异常：发生异常时会触发有当前运行进程切换到处理此异常的内核相关程序中
 3. 外围设备中断：外围设备完成用户请求后向CPU发出相应中断信号，此时CPU会暂停执行下一条即将执行的指令转去执行与中断信号对应的程序
- CPU执行的程序
 - 操作系统内核程序：操作系统的核心，最接近硬件的部分；OS的功能未必都在内核
 - 应用程序
 - 用户态

中断和异常

中断是计算机执行程序过程中出现异常情况或特殊请求，计算机停止运行现程序转而运行这些*异常情况*或特殊请求的处理，处理结束后再返回现程序。分别有：

内中断（异常、例外）：【三种类型】陷阱、陷入(trap)，故障(fault)，终止(abort)。【九种异常】缺页，除数为零，越界错误，存储保护错，非法指令，特权指令，运行超时，等待超时，算术运算错。**外中断（中断）**：时钟中断：IO中断请求。

故障是引起故障的指令在执行过程中CPU检测到一类与指令执行相关的意外事件。有些可恢复，有些不能。溢出和非法操作码则无法恢复；除数为0的情况则根据定点除法指令或浮点出发指令有着不同的处理方式。

陷阱是预先安排的“异常”时间，如同预先设定的“陷阱”。执行到陷阱指令时，CPU会调出特定程序进行相应处理。

陷阱提供了用户程序和内核之间一个像过程一样的接口（系统调用）

终止是在指令执行过程中发生了严重错误使程序无法继续执行，被迫终止运行。

系统调用

提供了操作系统提供的有效服务界面，一种可供应用程序调用的特殊函数，通过系统调用请求获得操作系统的内核服务。

调用者不需要知道调用或执行过程中做了什么，只需要遵循API并了解执行系统调用后系统做了什么。

系统调用类型：

与共享资源相关操作（内存分配、IO操作、进程管理、文件管理、设备管理、信息维护、通信等）均通过系统调用的方式向操作系统内核提出服务请求，由操作系统内核代为完成。其过程为：

应用程序 -调用-> API -存入系统调用号并触发中断[`int 0x80`]进入-> 中断处理函数 -根据调用号调用-> 系统调用 -返回-> 中断处理信号 -返回-> API -返回-> 应用程序

传递系统调用参数 --> 执行陷入指令（用户态） --> 执行相应的内请求和程序处理系统调用（核心态） --> 返回应用程序

操作系统的设计

操作系统的功能：内存管理，处理机管理，进程管理，文件管理，IO管理

- 传统操作系统结构
 - 无结构操作系统
 - 模块化OS
 - 分层式结构OS
- 微内核OS结构（现代操作系统结构）
 - 客户/服务器模式
 - 面对对象的程序设计

微内核OS结构

(非内核) OS的图形界面

(内核) 进程管理；内存管理；文件管理；IO管理

(内核) 时钟管理 / 中断管理 / 原语

硬件（裸机）

原语是一种不可被“中断”的特殊程序

微内核技术是能实现OS核心功能的小型内核，运行在核心态，开机后常驻内存。

微内核OS分为两部分：一部分用于提供各种服务的一组服务器（进程），另一部分是内核。

- 内核功能：
 1. 进程（线程）管理
 2. 低级存储器管理（用户程序逻辑空间到内存空间的物理地址的变换）
 3. 中断和陷入管理（中断和陷入）
- 具体功能：
 1. IO指令、置终端屏蔽指令、清内存、建存储保护、设置时钟指令
 2. 中断、异常、陷入
 3. 进程（线程）管理
 4. 系统调用
 5. 用户内存地址转换（逻辑 --> 物理映射）

操作系统的启动

开机时运行一个初始化程序（引导程序bootstrapProgram，位于ROM或EEPROM，称为固件），引导程序定位操作系统内核并装入内存，接着操作系统开始执行第一个进程并等待事件发生。

1. bios引导：开机 --> BIOS启动MBR --> MBR启动激活分区PBR --> 启动bootmgr --> 读取BCD --> 启动对应的系统
2. UEFI引导：开机 --> BIOS找到第一个硬盘 --> BIOS搜索BOOTx64.efi（计算机默认引导）或 bootmgrfw.efi（Windows默认引导） --> 读取BCD --> 启动BCD对应菜单的系统

虚拟机

单个计算机硬件抽象为多个不同的逻辑实体。

补充内容

- 单处理机系统中，进程与进程是不可并行的。
- 多道程序系统由于需要调度多个程序，所以系统开销相对变大。
- 不可能在用户态发生的事件是进程切换。缺页、系统调用可能在用户态发生（但必须在内核态执行）
- 访存时缺页属于异常