

操作系统

第一章 操作系统引论

主要学习操作系统的四大管理功能

- CPU
- 内存
- I/O设备
- 外设

操作系统（OS）的定义

- **OS是直接控制和管理计算机硬件、软件资源，合理地各类作业（例如：打开qq这一动作）进行调度，以方便用户使用的程序集合。**
- OS是配置在计算机硬件上的第一层软件，是对硬件系统的首次扩充。

OS在计算机中的地位（自底向上）

- 裸机
- 操作系统
- 应用软件

1.1 操作系统的目标和作用

1. 目标

- **方便性**：利用命令，而非使用机器语言
- **有效性**：提高系统资源的利用率；提高系统的吞吐量。
- **可扩充性**
- **开放性**：处理软硬件的兼容问题。

2. 作用

- **作为用户和计算机间的接口**：OS处于用户与计算机硬件系统之间

> 用户使用计算机的三种方式：

- > - 命令方式
- > - 系统调用方式
- > - 图标—窗口方式

- **作为计算机系统资源的管理者**：

- 处理机管理
- 存储器管理
- 设备管理
- 文件管理

- **实现了对计算机资源的抽象**

*作为系统资源的管理者:

- 执行一个程序前需要将该程序**放到内存中**，才能被CPU处理
- 用 QQ 和朋友视频聊天的过程:
 - Step 1:在各个文件夹中找到 QQ 安装的位置(D 😊 Tencent / QQ / Bin)
 - Step 2:双击打开 QQ . exe
 - 需要把该程序相关数据放入内存
 - Step 3:QQ程序正常运行
 - 对应的进程被处理机(CPU)处理 ,找到 QQ.exe 逐层打开文件夹
 - Step 4:开始和朋友视频聊天
 - 文件的存放位个程序(可执行需要将摄像头设备分配给进程

3.推动操作系统发展的主要动力：（了解）

- **不断提高计算机资源利用率**（最初）
 - 形成多道批处理系统
- **方便用户**
 - 实现人机交互、图形界面
- **器件的不断更新换代**
 - 微机OS8位, 16位, 32位
- **计算机体系结构的不断发展**
 - 多处理机OS
 - 网络操作系统
- **不断提出新的应用需求**
 - 实时性：实时系统
 - 多媒体
 - 安全性

1.2 操作系统的发展过程

单道批处理系统 <- 资源利用率 （连续运行）

多道批处理系统 <- 资源利用率 + 系统吞吐量

分时系统 <- 人机交互

无OS

1. 人工操作方式

- 工作方式：
 - 用户：用户是专业人员
 - 输入输出：纸带或卡片
- 缺点：
 - 用户独占全机
 - CPU等待人工操作

2. 脱机I/O方式（卡带->磁带）

- 优点
 - 减少了CPU的空闲时间
 - 提高了I/O速度

批处理系统（有OS）

- 用户使用系统提供的作业控制语言来描述自己对作业运行的控制意图，并将这些控制信息连同作业一起提交给计算机。
- 由OS去控制、调度各作业的运行并输出结果
- 由于作业进入系统后用户不在干预，从而提高了效率。

• 优点

- 提高系统资源的使用效率
- 提高作业吞吐量。

1. 单道批处理系统

• 特点

- 一次运行一个作业
- 一个一个连着完成

• 缺点

- CPU没有被充分利用
- 内存资源浪费

2. 多道批处理系统

多道程序设计的基本概念 [p7 图1-6 重要!!]

- 在多道批处理系统中,用户所提交的作业都先存放在外存上并排成一个队列,称为“**后备队列**”;
- 然后,由作业调度程序**按一定的算法**从后备队列中选择若干个作业调入**内存**,使它们共享 CPU 和系统中的各种资源
- 优点:
 - **资源利用率高**。 并发执行，资源共享
 - **系统吞吐量增大**。 CPU和其他资源保持忙碌；作业完成/运行不了才切换，系统开销小
- 缺点:
 - **平均周转时间长**（需要排队一次处理）
 - **没有人机交互功能**

****多道批处理系统需要解决的问题（了解）：***

- 处理机争用问题：满足运行，同时提高利用率
- 内存分配和保护问题
- I/O设备分配问题
- 文件的组织和管理问题：如何读写文件，关于权限。
- 作业管理问题
- 用户和系统的接口问题

1.2.4 分时系统

- 一台计算机连接多个终端,用户通过各自的终端把作业送入计算机;
- 计算机又通过终端向各个用户报告其作业的运行情况。
- 计算机以**时间片**为单位**轮流为各个用户/作业服务**
- 终端用户服务,并能及时地对用户服务请求予以响应
- **目标：**
 - 对用户的请求及时响应
 - 尽量提高系统资源的利用率

功能

- **人机交互**（边运行边调试）

- **共享主机**（设备昂贵）

关键问题

- 及时接收：多路
- 及时处理：
 - 作业直接入内存
 - 时间片轮转运行方式

特征

- **多路性**：即同时性，宏观上同时，微观上轮流
- **独立性**：每个用户感觉独占主机
- **及时性**：较短时间响应（1-3秒）
- **交互性**：用户请求多方面服务

1.2.5 实时系统

- **及时响应**，在**规定时间内**处理

1. 实时系统的类型

- **工业（武器）控制系统**
 - 火炮的自动控制系统
 - 飞机的自动加时系统
 - 导弹的制导系统
- **信息查询系统**
 - 飞机或火车的订票系统
- **多媒体系统**
 - DVD播放器所播放的数字电影
- **嵌入式系统**
 - 智能仪器和设备

2. 实时任务的类型

- 按周期性划分
 - 周期性实时任务
 - 非周期性实时任务
 - **开始截止时间**：在某时间以前必须开始执行
 - **完成截止时间**：在某时间以前必须完成
- 按照截止时间划分
 - **硬实时任务**：严格
 - **软实时任务**：不严格

3. 实时系统的特征

- 快速的响应时间
- 有限的交互能力
- 高可靠性

**了解

	多路性	独立性	及时性	交互性	可靠性
--	-----	-----	-----	-----	-----

	多路性	独立性	及时性	交互性	可靠性
批处理系统	无	无	差	差	一般
分时系统	多终端服务	有	好	好	可靠
实时系统	多路采集、多路控制	有	最好	一般	高度可靠

1.2.6 微机操作系统的发展 (考的少)

单用户单任务操作系统

- CP/M, DOS

单用户多任务操作系统

- Windows

多用户多任务操作系统

- Unix
- Linux

人名对应
Linux - Linux Torvalds
Unix - Dennis Ken Thompson Bell 公司
Dijkstra

1.3 操作系统的基本特征

1.3.1 并发 Concurrency

并行与并发

- 并发：
 - 指两个或多个时间在**同一时间间隔**内发生
 - 宏观同时，微观交替
- 并行：指两个或多个事件在**同一时刻**发生

注意（重要考点）

- 单核CPU：同一时刻只能执行一个程序，各个程序只能并发地执行
- 多核CPU：同一时刻可以同时执行多个程序，多个程序可以并行执行。
- 并发性是操作系统一个最基本特性

重要定义

- 进程：
 - 在系统中能独立运行并作为**资源分配的基本单位**，它是由一组机器指令、数据、堆栈等组成的，是一个能独立运行的活动实体。
- 原子，原语

1.3.2 共享 Sharing

1. 互斥共享方式

- **临界资源：一段时间内，只允许一个进程访问**
- e.g. 打印机，磁带机

2. 同时访问方式

- 允许在一段时间内由多个进程“同时”对资源进行访问。
- 单处理机：宏观同时，围观交替

并发和共享的关系：相辅相成

- 失去并发性，只有单进程进行，共享性无意义。
- 失去共享性，不允许多个进程访问同一个资源，也无法并发。

- 互斥共享方式：同时使用QQ和微信视频，同一时间段内摄像头只能分配给其中一个进程。
- 同时访问方式：同时使用QQ和微信发送文件，宏观上同时发送，微观上交替访问内存

1.3.3 虚拟 Virtual

一个程序需要放入内存，（创建进程）并给它分配CPU才能执行

- **时分复用技术**：利用空余时间，去服务别人
- **空分复用技术**：利用空余空间，进行程序

*1.3.4 异步 Asynchronism

- 进程的执行不是一贯到底的，系统中并发执行的多道程序“走走停停”。
- 已**不可预知**的速度向前推进。

1.4 操作系统的主要功能

1.4.1 处理机管理功能

- 进程控制
- 进程同步
 - 进程互斥方式：A访问，B就不能访问
 - 进程同步方式：进程间 同步完成
- 进程通信
- 调度
 - 作业调度
 - 进程调度

1.4.2 存储器管理功能

- 内存分配
 - 静态分配方式
 - 动态分配方式
- 内存保护
- 地址映射
- 内容扩充

1.4.3 设备管理功能

- 缓冲管理

- 设备分配
- 设备处理

1.4.4 文件管理功能

- 文件存储空间的管理
- 目录管理
- 文件的读/写管理和保护

***1.4.5 操作系统与用户之间的接口**

- 用户接口
- 程序接口

***1.4.6 现代操作系统的新功能**

- 系统安全
 - 认证技术
 - 密码技术
 - 访问控制技术
 - 反病毒技术
- 网络功能和服务
 - 网络通信
 - 资源管理
 - 应用互操作
- 支持多媒体
 - 接纳控制功能
 - 实时调度
 - 多媒体文件的存储

1.5 OS结构设计

1.5.1 传统操作系统结构

1. 无结构操作系统

- 缺点
 - 既庞大又杂乱，缺乏清晰的程序结构
 - 程序错误多，调试难、阅读理解难、维护难。

2. 模块化操作系统结构

- 优点
 - 提高了OS设计的正确性、可理解性和可维护性
 - 容易扩充，增强了OS的可适应性
 - 加速了OS的开发过程
- 缺点
 - 模块及接口划分较困难
 - 从功能上划分模块，未区别共享资源和独占资源
 - 由于管理的差异，使OS结构变得不够清晰

3. 分层式操作系统结构

- 原理：
 - 从资源管理观点出发，划分层次。
 - 各层模块间只能是单向调用关系，使模块间的调用变为有序性。
- 优点
 - 易保证系统的正确性。**调用关系清晰（高层对低层单向依赖）**
 - 易扩充和易维护
- 缺点
 - 系统效率低

** 操作系统的运行机制

特权指令vs非特权指令

- CPU设计和生成的时候就划分了特权指令和非特权指令。
- 应用程序：只能使用非特权指令。
- 内核程序：会让CPU执行一些特权指令
 - 例如 内存清零

内核态（核心态、管态） vs 用户态（目态）

- 处于**内核态**时，CPU执行**特权指令**
- 处于**用户态**时，CPU执行**非特权指令**
- ◦ CPU中有一个程序状态字寄存器（PSW），有一个二进制位，1表示“内核态”，0表示“用户态”。
- 内核态、用户态的切换
 - **内核态 → 用户态**
 - 执行一条**特权指令**-- **修改PSW的标志位为“用户态”**
 - = 操作系统主动让出CPU使用权
 - **用户态 → 内核态**
 - 由“**中断**”引发，**硬件自动完成变态过程**。
 - = 操作系统将强行夺回CPU的使用权

1.5.4 操作系统的体系结构

操作系统的内核

- 原语
 - 一种特殊的程序，具有原子性。
 - 这段程序的运行必须一气呵成，不被中断。
- 内核是操作系统最基本、最核心的部分
- 实现操作系统的内核功能的那些程序就是内核程序。

大内核 vs 微内核

微内核效率一定比大内核高么？

不一定。因为微内核会频繁进行变态操作。

微内核的OS结构

- 特点：
 - 小而精炼
 - 灵活性和可扩充性好
 - 可靠性高

- 适用于分布式系统

客户/服务器模式

- 优点
 - 充分模块化
 - 减少系统的内存需求
 - 高可移植性
- 缺点
 - 各模块与微内核通过通信机制交互，系统**运行效率较低**
 - 不可靠

微内核的基本功能

- 进程（线程）管理
 - **机制与策略分离**
- 低级存储器管理
 - 逻辑地址和物理地址映射
- 终端和陷入处理

微内核OS的优缺点

- 优点
 - 提高了系统的可扩展性
 - 增强了系统的可靠性
 - 可移植性强
 - 提供了对分布式系统的支持
 - 融入了面对象技术
- 缺点
 - 运行效率低

中断和异常

- 中断的作用
 - **“中断”是让操作系统内核夺回CPU使用权的唯一途径**
- 中断的类型
 - 内中断：与当前执行的指令有关，中断信号来自于CPU内部
 - 外中断：与当前执行的指令无关，中断信号来自于CPU外部

＞ 补充：陷入指令（非特权指令）：主动地将CPU控制权还给操作系统内核。

＞

＞ 内中断案例：

＞ 案例1：在用户态执行特权指令，引发中断

＞ 案例2：在用户态执行非法程序，例如除法分母为0，引发中断

＞ 案例3：在用户态执行陷入指令，引发中断

＞

＞ 外中断案例：

＞ 案例1：利用时钟部件，每隔30ms给CPU发送一个时钟中断信号。

＞ 案例2：I/O中断请求

- 中断机制的基本原理(不考)
 - 查询“中断向量表”，来执行对应程序

补充 **OS接口

系统调用与库函数的区别

- 普通应用程序：系统调用 / 调库函数
- 编程语言：向上提供库函数
- 操作系统：向上提供系统调用。
- 裸机

系统调用的必要性：

- 操作系统内核对共享资源进行统一的管理，并向上提供“系统调用”
- 否则会出现，两个人同时使用打印机，而输出的内容是两个人需求的混合。

与共享资源有关的操作，必须通过系统调用完成

- 分类：
 - 设备管理
 - 文件管理
 - 进程控制
 - 进程通信
 - 内存控制

系统调用的过程

- 传递系统调用参数
- 执行陷入指令（**用户态**）：产生中断信号，请求系统调用
- 执行响应的内核请求程序处理系统调用（**核心态**）
- 返回