

操作系统的处理器管理功能概要

处理器资源-最重要的资源

处理器调度-如何分配

- 作业调度
 - 作业后备队列选取若干作业装入内存
 - 作业选择与资源分配
 - 调入内存建立进程
- 进程调度
 - 就绪进程队列
 - 进程选择与处理及分配
 - 保存现场¹, 恢复现场²
- 调度算法
 - 先来先服务, nice值低的进程优先权高, 时间轮片选择的几率更大, 运行的机会更多

进程控制

- 主要任务
 - 创建进程
 - 撤销进程
 - 切换进程状态
- 进程控制机制
 - 原语³
 - 进程&线程⁴

进程同步

- 主要任务
 - 进程/现成的并发执行协调
 - 互斥/同步方式
 - 临界资源⁵ & 临界区⁶
 - 要保证各个进程互斥的访问临界资源/临界区
- 进程同步机制
 - 开/关锁原语【支持简单的/短时间的互斥问题】
 - 信号量机制
 - 管程【高级编程概念】

进程通信

- 主要任务
 - 进程(线程)间的信息交换
- 进程通信方式
 - 共享存储器
 - 管道通信

- 消息传递系统
 - 消息缓冲队列
 - 邮箱
-

内存资源管理

内存分配

- 主要任务
 - 使程序各得其所
 - 提高存储器利用率
 - 适应程序和数据的动态增长需要
- 内存分配机制
 - 内存分配用数据结构
 - 内存分配
 - 连续/离散分配方式
 - 静态/动态分配方式
 - 内存回收

内存保护

- 主要任务
 - 首先保证内核安全，其次确保程序见互不干扰
 - 存取访问控制
- 内存保护机制
 - 越界检查
 - 硬件实现
 - 保护方式
 - 上下限界限寄存器
 - 页号<页表长度
 - 段号<段表长度&&段内地址<段长
 - 特权级比较

地址映射/转换

- 主要任务
 - 逻辑地址转换为物理地址
- 基本概念
 - 地址空间&内存空间
 - 逻辑/相对/有效地址
 - 物理/绝对地址
- 地址映射机制
 - 硬件实现
 - 重定位寄存器、页表/段表、快表

内存扩充

- 主要任务
 - 从逻辑上扩充内存容量

- 内存扩充的可行性
 - 程序运行局部性原理
 - 离散分配方式
 - 内存扩充机制
 - 虚拟存储技术
 - 硬件(页/段表、缺页/段中断、地址变换)
 - 软件(请求调入/置换功能)
-

设备管理

设备分配

- 主要任务
 - 设备及相应控制器的通道的分配
- 设备分配机制
 - 设备分配用的数据结构
 - 设备分配与设备分配所对应的方式
 - 设备分配与回收算法
 - **设备独立性**⁷
 - **虚拟设备**⁸

缓冲管理

- 主要任务
 - 管理各样的缓冲区
 - 缓和CPU和I/O速度不匹配
 - 提高资源利用率和系统吞吐量
- 缓冲管理机制
 - 单/双/多缓冲类型
 - **字符缓冲**⁹ 与 **盘块缓冲**¹⁰
 - 共用缓冲池机制

设备处理

- 主要任务
 - CPU与设备控制器间的通信
 - 设备处理过程
 - I/O请求提出
 - I/O请求合法性检查
 - 了解设备状态
 - 传递参数并设置工作方式
 - 通道程序自动构成¹¹
 - 发送I/O指令和启动I/O设备
 - 及时响应中断请求
-

文件管理

文件存储空间管理

- 主要任务
 - 使每个文件各得其所
 - 提高外存空间利用率
 - 提高外存空间访问速度
- 文件存储空间管理机制
 - 存储空间所用的数据结构
 - 存储空间的分配与回收功能
 - 离散/连续的分配方式
 - 以盘块/簇¹² 为基本分配单位

目录管理

- 主要任务
 - 文件的按名存取
 - 提高文件查找速度
 - 支持文件的重命名，共享，保护
- 目录管理机制
 - 文件控制块与索引节点(inode)
 - 目录结构
 - 目录检索手段

文件的读写和存取控制

- 文件的读写管理
 - 从外存读取数据或将数据存入外存
 - 目录检索->外存地址->读写指针->读写操作
 - 文件描述符表和文件表(提高文件的读写)
- 文件的存取控制
 - 防止未经核准的用户存取文件
 - 防止冒名顶替存取文件
 - 防止以不正确的方式使用文件
 - 系统级/用户级/文件级存取控制

操作系统的特征

- 并发性
 - 并行与并发、程序与进程/线程¹³
- 共享性
 - 互斥共享方式，同时访问方式，可重入码¹⁴
- 虚拟性
 - 模拟虚拟机，虚拟内存，虚拟盘，虚拟设备
- 异步性
 - 进程的执行顺序与执行时间的不确定性

操作系统的设计原则及非结构化设计问题

操作系统的 设计原则

- 可维护性

- 纠错性/适应性/完善性/预防性维护
- 可靠性
 - 正确性/健壮性
- 可理解性
- 可用性
- 性能
 - 系统资源利用率及用户请求响应

模块化操作系统的结构评价

- 优点
 - 提高了设计的正确性、可理解性和可维护性
 - 增强了操作系统的可适应性
 - 加速了操作系统的开发过程
 - 尚存改进空间方面
 - 模块划分和接口规定难保正确和合理
 - 未能区别共享资源和独占资源
 - 管理差异导致模块间依赖关系复杂
-
-

1. 将当前寄存器中的内容入栈，并跳转至下一现场 [♂](#)
2. 把原寄存器中的内容从栈顶恢复到寄存器中 [♂](#)
3. 计算机进程的控制通常由原语完成。所谓原语，一般是指由若干条指令组成的程序段，用来实现某个特定功能，在执行过程中不可被中断。 [♂](#)
4. 线程包含于进程，共享进程在内存中的资源 [♂](#)
5. 一段时间之内只能被一个进程所访问的资源叫做临界资源 [♂](#)
6. 临界资源的代码区叫做临界区，访问临界资源的代码栈 [♂](#)
7. 程序和设备不直接关联，程序应访问逻辑设备，便于程序移植，提高程序可扩充性，提高程序的可维护性 [♂](#)
8. 独占性质设备，设置共享，通过虚拟技术，把物理上的独占设备转换成共享设备 [♂](#)
9. 对于键盘使用字符缓冲 [♂](#)
10. 对于磁盘/软盘/硬盘/使用盘块缓冲 [♂](#)
11. 仅限于通道控制方式，通道的形成，是由驱动程序自动构成的 [♂](#)
12. 簇是由多个扇区组成的逻辑结构，每个扇区512个字节，簇的大小就是扇区的整数倍 [♂](#)
13. 并行：任何一个时间点上，多个事件同时发生；并发：一段时间内，多个事件同时发生；线程：本身不用有资源，可以共享进程的资源 [♂](#)
14. 不可修改的可执行代码，大家都对代码调用和执行 [♂](#)