第二章 进程与线程

进程

进程与线程

进程

进程控制

进程的创建（创建原语）
- 引起创建的事件
  - 用户登录 — 作业调度 — 提供服务 — 应用请求
- 原语执行过程
  - 分配进程标识号，申请PCB（PCB有限）— 为进程分配资源、为程序和数据以及用户栈分配必要的内存空间
  - 初始化PCB，包括初始化标志信息、初始化处理机状态信息、初始化处理机控制信息、设置进程优先级
  - 若进程就绪队列可以接纳新进程，进程就进入就绪态

进程的终止（撤销原语）
- 引起终止的事件
  - 正常结束 — 异常结束 — 外界干预
- 原语执行过程
  - 根据被终止进程的标识符，检索PCB，读取进程状态 — 若进程处于运行态，终止运行、剥夺处理机
  - 终止进程之下的子进程，该进程所有资源还给父进程或操作系统 — PCB从队列中删除

进程的组赛和唤醒
- 阻塞（自我阻塞）
  - 引起
    - 需要等待系统分配资源 — 需要等待相互合作的进程完成工作
  - 原语执行过程
    - 找到要被阻塞进程标识号对应PCB
    - 若该进程处于运行态，保护现场，状态转为阻塞态停止运行
    - 将PCB插入对应等待队列
- 唤醒（其他进程唤醒）
  - 引起
    - 等待事件发生
  - 原语执行过程
    - 找到等待队列中进程相应的PCB — 将其从等待队列中移出，状态置为就绪态
    - 将PCB插入就绪队列，等待调度程序调度

进程切换（运行态<-->就绪态，内核态下完成）
- 引起
  - 时间片到 — 更高优先级到达 — 进程主动阻塞 — 当前进程终止
- 原语执行过程
  - 挂起一个进程，保存处理机上下文，包括程序计数器和其他寄存器
  - 更新PCB信息 — 把进程的PCB移入相应队列（就绪/阻塞队列）
  - 选择另一个进程执行，更新其PCB，更新内存管理的数据结构
  - 跳转到新进程PCB中的程序计数器所指位置 — 恢复处理机上下文

进程通信

共享存储
- 概念（进程<-->共享空间<-->进程）— 通信进程间存在直接访问共享空间，实现信息交换
- 操作系统只负责为通信进程提供可共享使用的存储空间和同步互斥工具，数据交换由用户安排读写指令完成
  - 低级方式：基于数据结构共享 — 高级方式：基于存储区共享

消息传递
- 概念 — 以格式化的消息为单位进行数据交换，通过系统提供的发送消息和接收消息原语实现
- 直接通信方式 — 发送进程将消息挂在接受进程的消息缓冲队列上
- 间接通信方式 — 发送进程把消息发送给某一个中间实体，接收进程从中间实体中获得消息（电子邮件系统）

管道通信（一种特殊共享文件、缓冲区，受文件系统管理）
- 发送进程以字符流形式将大量数据写入管道，接收进程取 — 当管道写满时，写进程阻塞，等待读进程将数据取走
- 管道半双工通信，不可同时读写（全双工通信则双管道）— 管道变空后，读进程阻塞
- 只能一个读进程，可以多个写进程

进程与作业
- 作业是用户提交、以用户任务为单位，进程是系统自动生成，操作系统控制为单位 — 一个作业可能对应多个进程

线程

概念
- 为减少程序在并发执行中的切换开销，引入线程概念（切线程不切进程则资源不用切换）

- 进程 = 资源 + 执行执行序列（将资源和指令执行分开）
- 线程
  - 进程内（在一个资源下）启动多个轻巧的、可以来回切换的指令序列
  - 保留了并发的优点，避免了进程切换的代价
- 此处李治军老师进程切换的内容，分离为线程的指令的切换与映射表上资源的切换两部分讲，讲解内存管理的时候再讲资源问题，两者分离讲解，进程切换问题迎刃而解

调度
- 引入线程前 — 进程是资源和独立调度的基本单位
- 引入线程后 — 线程是独立调度的基本单位，进程是资源的基本单元 — 进程仅为系统资源分配单元，线程处理机分配单元（服务单元）
- 进程的地址空间相互独立，统一进程的各线程之间共享进程的资源，某进程的线程对其他进程不可见

线程控制块TCB
- 线程标识符、线程运行状态、优先级 — 线程切换保存/恢复 — 程序计数器PC、其他寄存器、堆栈指针

实现方式
- 用户级线程ULT — 有关线程管理的所有工作都由应用程序完成，内核意识不到线程的存在
- 内核级线程KLT — 线程的管理工作全部由内核完成
- 用户级线程与内核级线程：一个栈到一套栈

- 经多个用户级线程映射到一个内核级线程。线程管理在用户空间完成，用户级线程对操作系统不可见

多线程模型

多对一模型

线程库

进程

用户态
内核态
操作程序

线程管理在用户控件，无需切换

一个线程阻塞，全部线程阻塞

一对一模型

视频聊天　文字聊天　文件传输

用户级线程　用户级线程　用户级线程

内核级线程　内核级线程　内核级线程

进程

应用程序
用户态
内核态
操作程序

每个用户级线程映射到一个内核级线程上

创建线程开销大、切换频繁影响应用程序性能

多对多模型

视频聊天　文字聊天　文件传输

用户级线程　用户级线程　用户级线程

线程库

内核级线程　内核级线程

进程

应用程序
用户态
内核态
操作程序

多个线程映射到多个内核线程上

结合上述两种，既可提高并发性，有降低开销

用户级线程：代码逻辑载体

内核级线程：运行机会载体