第四讲 线程

2023年2月26日 21:26

Why?

1. 程序结构本身的并行特征（执行流间内在的逻辑关系）

   Server 应用（web server. Database, ...）

   APP 应用（视频. 图像. 大数据 ...）

2. 快速响应（前台. 后台）

   UI（word. 浏览器. 播放器 ...）

3. 性能.

   多处理器充分利用（多核并行）

   CPU/设备 并行（DMA. 播放器）

一. 线程.

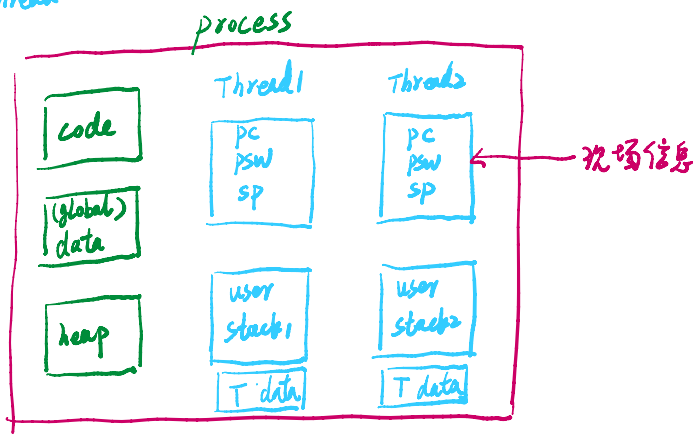
相对独立的执行流.

可并发调度.

process：资源管理的基本单位.

v.s.

thread：处理器调度的基本单位.



PCB

① 创建代价

② 任务切换代价

③ 任务间协同或通信代价

④ 提升CPU占比的保障性.

⑤ 并行任务间的契合度.

二. 线程的实现

1. 用户级别线程

2. 核心级别线程

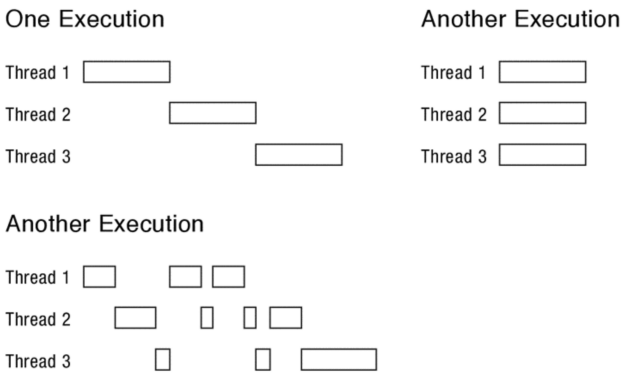
3. 混合线程

三. 线程上下文切换

1. 主动 — yield

2. 被动 — interrupt, exception, trap.

四. 线程执行速度的不确定性.



① 调度可能在任何时刻发生

② 调度算法与系统的瞬时状态相关. 任何就绪的线程都可能被运中.

(不可表现. 调试困难)

五. 并行编程模型.

① 多线程. (任何可独立调度的执行流)

② 事件驱动 (UI. server)

③ 数据并行 (SIMD < SSE. AVX / GPU)

④ 混合模型