

操作系统原理笔记

陈鸿峥

2019.03*

目录

1	操作系统概述	1
1.1	概述	1
1.2	发展历史	1
2	进程	2

本课程使用的教材为William Stallings《操作系统—精髓与设计原理（第八版）》。其他参考资料包括Stanford CS140、CMU15-460、*Operating System Concepts (10th ed.)*。

关于计算机系统的内容在此不再赘述，详情参见计算机组成原理的笔记。

1 操作系统概述

1.1 概述

操作系统核心即怎么虚拟多几个冯诺依曼计算机出来给程序用。操作系统是控制应用程序执行的程序，是应用程序和计算机硬件间的接口（屏蔽硬件细节）。

这里先解释几个概念

- 并发：两件事情可以同时(simultaneously)发生，没有时间限制， $t_1 > t_2$ ， $t_1 < t_2$ ， $t_1 = t_2$ 都可
- 同步：两个事件有确定的时间限制
- 异步：两件事不知道何时发生

*Build 20190314

1.2 发展历史

- 串行处理(1940s): 没有OS, 人工调度, 准备时间长
- 简单批处理系统(1950s):
 - 使用监控程序(monitor), 读入用户程序执行
 - 提供内存保护、计时器、特权指令、中断
 - 两种操作模式: 用户态、内核态(mode)
 - 单道程序(uniprogramming)批处理: 处理器必须等到IO指令结束后才能继续
- 多道程序批处理(1950s末): 多个作业同时进入主存, 切换运行, **充分利用处理器**
- 分时系统(1961): MIT CTSS(Compatible Time-Sharing System), 满足用户与计算机交互的需要, **减小响应时间**; 多个交互作业, 多个用户, 把运行时间分成很短的时间片轮流分配
- 实时系统: 专用, 工业、金融、军事

现代的操作系统通常同时具有分时、实时和多道批处理的功能, 因此被称为通用操作系统。而OS也不仅是在PC机上有, 网络OS、分布式OS、嵌入式OS层出不穷。

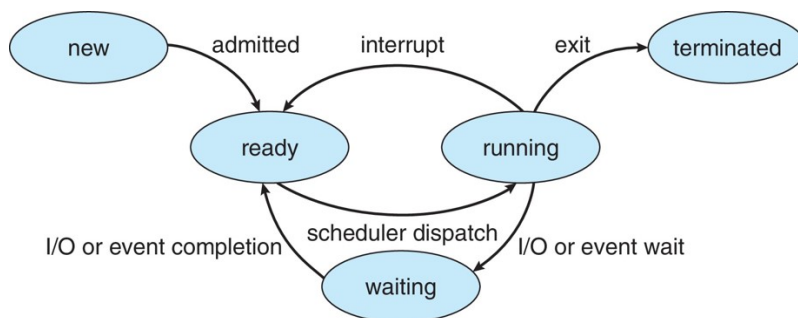
2 进程

进程(process)是运行时(**running/in execution**)程序的实例。

- 程序并发执行的特征 (多道程序): 间断性、无封闭性、不可再现性 (破坏冯顺序执行特性)
- 进程的特点: 动态性、并发性、独立性、异步性
- 进程的作用
 - 提升CPU利用率: 将多个进程重叠 (一个进程IO时另一个计算)
 - 降低延迟(latency): 并发执行, 不断切换, 防止卡住

进程控制块(Process Control Block, PCB), 在Unix是proc, 在Linux是task_struct

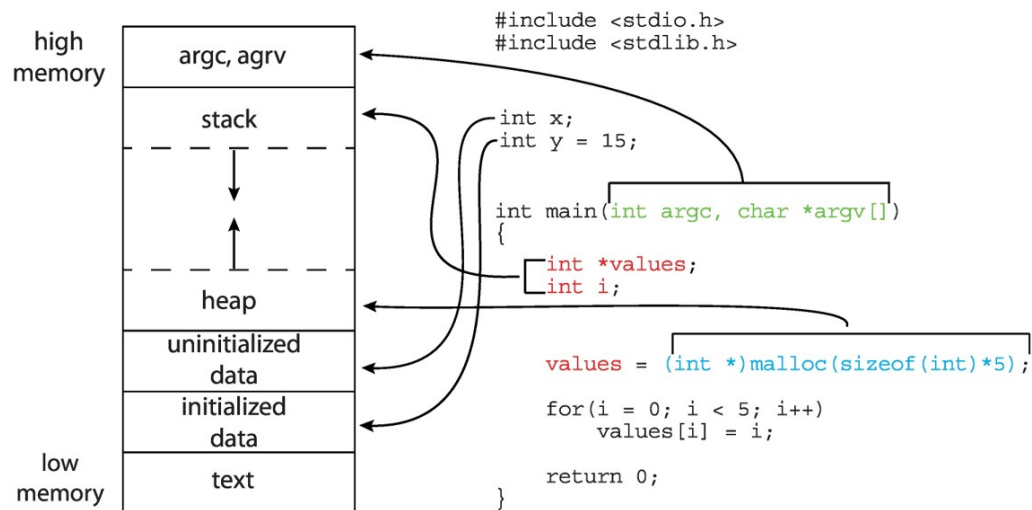
- 进程标识符/ID
- 状态: 运行、就绪、等待/阻塞、新建、退出



- 优先级

- 程序计数器(PC)
- 内存指针：报错指向程序代码、相关数据和共享内存的指针
- 上下文数据(context)：进程被中断时寄存器中的数据
- IO状态信息
- 记账信息(accounting)：占用处理器时间、时钟数总和、时间限制等

内存组织：代码段、数据段、堆段、栈段（从小地址往大地址）



可以参考原始的UNIX论文¹。

- 创建进程：fork、waitpid
- 删除进程：exit、kill
- 执行进程：execve

进程切换：扔掉堆、数据段，读入新代码段、数据段，重设栈段、堆段，初始化寄存器，开始执行

¹<http://www.scs.stanford.edu/19wi-cs140/sched/readings/unix.pdf>