

进程概述

程序是包含一系列信息的文件，这些信息描述了如何在运行时创建一个进程：

- 二进制格式标识：每个程序文件都包含用于描述可执行文件格式的元信息。内核利用此信息来解释文件中的其他信息。（ELF可执行连接格式）
- 机器语言指令：对程序算法进行编码。
- 程序入口地址：标识程序开始执行时的起始指令位置。
- 数据：程序文件包含的变量初始值和程序使用的字面量值（比如字符串）。
- 符号表及重定位表：描述程序中函数和变量的位置及名称。这些表格有多重用途，其中包括调试和运行时的符号解析（动态链接）。GDB调试
- 共享库和动态链接信息：程序文件所包含的一些字段，列出了程序运行时需要使用的共享库，以及加载共享库的动态连接器的路径名。
- 其他信息：程序文件还包含许多其他信息，用以描述如何创建进程。

- **进程是正在运行的程序的实例**。是一个具有一定独立功能的程序关于某个数据集合的一次运行活动。它是操作系统动态执行的基本单元，在传统的操作系统中，**进程既是基本的分配单元，也是基本的执行单元**。
- 可以用**一个程序来创建多个进程**，进程是由内核定义的**抽象实体**，并为该实体分配用以执行程序的各项系统资源。从内核的角度看，进程由用户内存空间和一系列内核数据结构组成，其中**用户内存空间**包含了程序代码及代码所使用的变量，而内核数据结构则用于维护进程状态信息。记录在内核数据结构中的信息包括许多与进程相关的标识号（IDs）、虚拟内存表、打开文件的描述符表、信号传递及处理的有关信息、进程资源使用及限制、当前工作目录和大量的其他信息。

- 单道程序，即在计算机内存中只允许一个的程序运行。
- 多道程序设计技术是在计算机内存中同时存放几道相互独立的程序，使它们在管理程序控制下，相互穿插运行，两个或两个以上程序在计算机系统中同处于开始到结束之间的状态，这些程序共享计算机系统资源。引入多道程序设计技术的根本目的是为了
提高 CPU 的利用率。
- 对于一个单 CPU 系统来说，程序同时处于运行状态只是一种宏观上的概念，他们虽然都已经开始运行，但就微观而言，任意时刻，CPU 上运行的程序只有一个。
- 在多道程序设计模型中，多个进程轮流使用 CPU。而当下常见 CPU 为纳秒级，1秒可以执行大约 10 亿条指令。由于人眼的反应速度是毫秒级，所以看似同时在运行。

$$1s=1\ 000ms=1000\ 000\mu s=1000\ 000\ 000ns$$

- 时间片 (timeslice) 又称为“量子 (quantum) ”或“处理器片 (processor slice) ”是操作系统分配给每个正在运行的进程微观上的一段 CPU 时间。事实上，虽然一台计算机通常可能有多个 CPU，但是同一个 CPU 永远不可能真正地同时运行多个任务。在只考虑一个 CPU 的情况下，这些进程“看起来像”同时运行的，实则**是轮番穿插地运行**，由于时间片通常很短（在 Linux 上为 5ms - 800ms），用户不会感觉到。

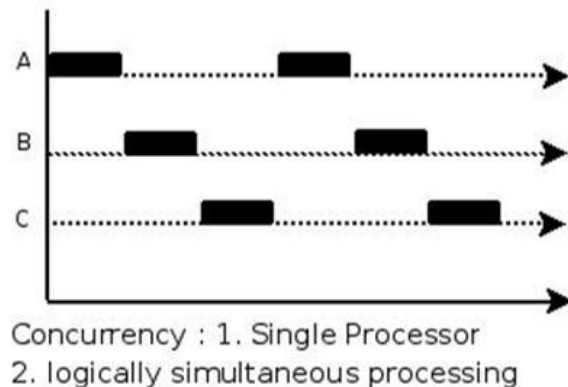
OS中的进程调度策略

- **时间片由操作系统内核的调度程序分配给每个进程**。首先，内核会给每个进程分配相等的初始时间片，然后每个进程轮番地执行相应的时间，当所有进程都处于时间片耗尽的状态时，内核会重新为每个进程计算并分配时间片，如此往复。

04 / 并行和并发

我们研究的就是并发
Linux高并发服务器开发
我们在编码阶段并行问题考虑的很少

- 并行(parallel): 指在同一时刻, 有多条指令在多个处理器上同时执行。
- 并发(concurrency): 指在同一时刻只能有一条指令执行, 但多个进程指令被快速的轮换执行, 使得在宏观上具有多个进程同时执行的效果, 但在微观上并不是同时执行的, 只是把时间分成若干段, 使多个进程快速交替的执行。

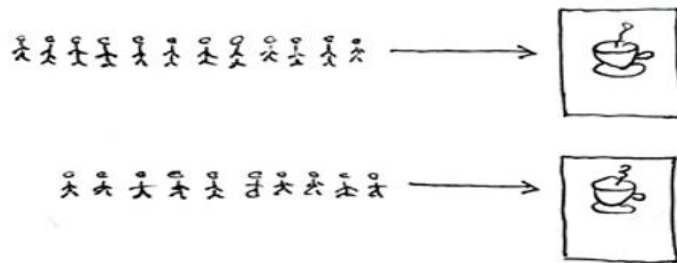


- 并发是两个队列交替使用一台咖啡机。
- 并行是两个队列同时使用两台咖啡机。

Concurrent = Two Queues One Coffee Machine



Parallel = Two Queues Two Coffee Machines



- 为了管理进程，内核必须对每个进程所做的事情进行清楚的描述。内核为每个进程分配一个 PCB (Processing Control Block) 进程控制块，维护进程相关的信息，Linux 内核的进程控制块是 `task_struct` 结构体。
- 在 `/usr/src/linux-headers-xxx/include/linux/sched.h` 文件中可以查看 `struct task_struct` 结构体定义。其内部成员有很多，我们只需要掌握以下部分即可：
 - 进程id：系统中每个进程有唯一的 id，用 `pid_t` 类型表示，其实就是一个非负整数
 - 进程的状态：有就绪、运行、挂起、停止等状态
 - 进程切换时需要保存和恢复的一些CPU寄存器
 - 描述虚拟地址空间的信息
 - 描述控制终端的信息

- 当前工作目录 (Current Working Directory)
- umask 掩码
- 文件描述符表, 包含很多指向 file 结构体的指针
- 和信号相关的信息
- 用户 id 和组 id
- 会话 (Session) 和进程组
- 进程可以使用的资源上限 (Resource Limit)



牛客大学

- 专业求职辅导 -

THANKS



关注【牛客大学】公众号
回复“牛客大学”获取更多求职资料