

进程的基本概念

计算机中，CPU是最宝贵的资源，为了提高CPU的利用率，引入了多道程序设计的概念。当内存中多个程序存在时，如果不对人们熟悉的“程序”的概念加以扩充，就无法刻画多个程序共同运行时系统呈现出的特征。

一、进程的引入

多道程序系统中，程序具有：并行、制约以及动态的特征。程序概念难以便是和反映系统中的情况：

1. 程序是一个静态的概念

程序是完成某个功能的指令集和。系统实际上是出于不断变化的状态中，程序不能反映这种动态性。

2. 程序概念不能反映系统中的并行特性

例如：两个C语言源程序由一个编译程序完成编译，若用程序概念理解，内存中只有一个编译程序运行（两个源程序看作编译程序的输入数据），但是这样无法说明明白内存中运行着两个任务。程序的概念不能表示这种并行情况，反映不了他们活动的规律和状态变化。就像不能用菜谱（程序）代替炒菜（程序执行的过程）一样（这句话我稍微修改了一下，感觉应该是这样告诉才对）。

二、进程的定义

进程：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，是系统进行资源分配和调度运行的基本单位。

三、进程与程序的差别

1. 进程是一个动态的概念

进程是程序的一次执行过程，是动态概念。

程序是一组有序的指令集和，是静态概念。

2. 不同的进程可以执行同一个程序

区分进程的条件：所执行的程序和数据集合。

两个进程即使执行在相同的程序上，只要他们运行在不同的数据集合上，他们也是两个进程。例如：多个用户同时调用同一个编译程序编译他们编写的C语言源程序，由于编译程序运行在不同的数据集合（不同的C语言源程序）上，于是产生了一个个不同的进程。

3. 每个进程都有自己的生命周期

当操作系统要完成某个任务时，它会创建一个进程。当进程完成任务之后，系统就会撤销这个进程，收回它所占用的资源。从创建到撤销的时间段就是进程的生命期。

4. 进程之间存在并发性

在一个系统中，同时会存在多个进程。他们轮流占用CPU和各种资源。

5. 进程间会相互制约

进程是系统中资源分配和运行调度的单位，在对资源的共享和竞争中，必然相互制约，影响各自向前推进的速度。

6. 进程可以创建子进程，程序不能创建子程序

7. 从结构上讲，每个进程都由程序、数据和一个进程控制块（Process Control Block, PCB）组成

四、进程的重要特征

1. **动态特征**：进程对应于程序的运行，动态产生、消亡，在其生命周期中进程也是动态的。

2. **并发特征**：任何进程都可以同其他进程一起向前推进。

3. **独立特征**：进程是相对完整的调度单位，可以获得CPU，参与并发执行。

4. **交往特征**：一个进程在执行过程中可与其他进程产生直接或间接关系。

5. **异步特征**：每个进程都以相对独立、不可预知的速度向前推进。

6. **结构特征**：每个进程都有一个PCB作为他的数据结构。

进程最基本的特征是并发和共享特征。

五、进程的状态与转换

1. 进程的三种基本状态

- a. 运行状态：获得CPU的进程处于此状态，对应的程序在CPU上运行着。
- b. 阻塞状态：为了等待某个外部事件的发生（如等待I/O操作的完成，等待另一个进程发来消息），暂时无法运行。也成为等待状态。
- c. 就绪状态：具备了一切运行需要的条件，由于其他进程占用CPU而暂时无法运行。

2. 进程状态转换

- a. 运行状态 ==> 阻塞状态：例如正在运行的进程提出I/O请求，由运行状态转化为阻塞状态。
- b. 阻塞状态 ==> 就绪状态：例如I/O操作完成之后，由阻塞状态转化为就绪状态。
- c. 就绪状态 ==> 运行状态：例如就绪状态的进程被进程调度程序选中，分配到CPU中运行，由就绪状态转化为运行状态。
- d. 运行状态 ==> 就绪状态：处于运行状态的进程的时间片用完，不得不让出uCPU，由运行状态转化为就绪状态。

3. 进程的类型

- a. 系统进程：操作系统用来管理资源的进程，当系统进程处于运行态时，CPU处于管态，系统之间的关系由操作系统负责。
- b. 用户进程：操作系统可以独立执行的的用户程序段，当用户进程处于运行态时，CPU处于目态，用户进程之间的关系由用户负责。

六、进程控制块

1. 进程的三个组成部分

a. 程序

b. 数据

c. **进程控制块（PCB）**：为了管理和控制进程，系统在创建每个进程时，都为其开辟一个专用的存储区，用以记录它在系统中的动态特性。系统根据存储区的信息对进程实施控制管理。进程任务完成后，系统收回该存储区，进程随之消亡，这一存储区就是进程控制块。

PCB随着进程的创建而建立，撤销而消亡。系统根据PCB感知一个进程的存在，**PCB是进程存在的唯一物理标识**（这一点可以类比作业控制块JCB）。

2. 进程控制块的内容

PCB在不同的语言中，可能用不同的数据结构表示。为了系统管理和控制进程方便，系统常常将所有进程的PCB存放在内存中系统表格区（这是什么区？不懂，待我仔细查查），并按照进程内部标号由小到大顺序存放。

整个系统中各进程的的PCB集合可用数组表示。这时进程内部标号可以与数组元素下标联系。

各系统预留的PCB空间往往是固定的，如UNIX系统中规定进程数量不超过500个（这一点我有点怀疑）。

操作系统不同，PCB的格式、大小及内容也不尽相同。一般的，应该包含如下四个信息。

- a. **标识信息**：进程名。
- b. **说明信息**：进程状态、程序存放位置。
- c. **现场信息**：通用寄存器内存、控制寄存器内存、断点地址。
- d. **管理信息**：进程优先数、队列指针。

七、进程控制块的组织

系统中，有着许多不同状态的进程，处于阻塞状态的进程阻塞原因各不相同，为了便于调度和管理，常将进程控制块PCB用适当的方法组织起来。

1. 线性结构

把所有不同状态的进程的PCB组织在一个表格中。

最简单，适用于进程数目不多的操作系统，如UNIX系统，缺点是调用时，往往需要查询整个PCB表，时间复杂度略高。

2. 索引结构

分别把具有不同状态的进程PCB组织在同一个表中，于是有就绪进程表、运行进程表（多机系统中，还有现在的多核系统应该也有吧）以及各种等待事件的阻塞进程表。

系统中的一些固定单元分别指出各表的起始地址。

3. 链式结构

采用队列形式时，每个进程的PCB中要增加一个链指针表项，指向队列的下一个PCB起始地址。

为了对这些队列进行管理，操作系统要做三件事：

- 把处于同一状态的进程的PCB通过各自队列的指针链接在一起，形成队列。
- 为每一个队列设立一个对头指针，总是指向队首的PCB。
- 排在队尾的PCB的队列指针项内容应该是“-1”或者一个特殊符号，表示这是队尾PCB。

在单CPU系统中，任何时刻都只有一个处于运行态的进程

所有处于阻塞队列中的PCB应该根据产生阻塞的原因今进行排队，每一个都称为阻塞队列，比如等待磁盘I/O的阻塞队列，等待打印机输出的阻塞队列。

八、进程控制

1. 原语

要对进程进行控制，系统中必须设置一个机构，它具有创建进程、撤销进程、进程通信和资源管理等功能，这样的结构称为操作系统的**内核（kernel）**

内核本身不是一个进程，而是硬件的首次延伸，它是加在硬件上的第一层软件。内核是通过执行各种原语操作来完成各种控制和管理功能的。

原语（primitive）是机器指令的延伸，用若干条机器指令构成，用以完成特定功能的一段程序。为保证操作的正确性，原语在执行期间是**不可分割的**（这点可以类比数据库中的事务）。

用于进程控制的原语有：创建进程原语、撤销进程原语、阻塞进程原语、唤醒进程原语、调度进程原语、改变优先级原语等。

2. 创建进程原语

一个进程如果需要时，可以创建一个新的进程。被建立的进程称为**子进程**，建立者进程称为**父进程**。

所有的进程都只能通过父进程建立，不能自生自灭。

创建进程原语进程调用，用以建立子进程

该原语的主要工作：为被建立的进程简历一个进程控制块，填入相应的初始值。主要操作过程是先向系统的PCB空间申请分配一个空闲的PCB，然后根据父进程所提供的参数，将子进程的PCB表目初始化，最后返回一个子进程内部名。

3. 撤销进程原语

由父进程撤销子进程的PCB，注意，这里会撤销一个以该子进程为根的进程子树，并回收占用的全部资源。

4. 阻塞原语

在阻塞原语的作用下，进程由运行状态转化为阻塞状态。

5. 唤醒原语

在唤醒原语的作用下，进程由阻塞状态转化为就绪状态。

6. 改变进程优先级原语

进程的优先级是表示进程的重要性以及运行的优先性，供进程调度程序调度进程运行时使用。

为了防止一些进程因优先级较低，而长期得不到运行，许多系统采用动态优先级，**进程的优先级按照一些原则变化。**

通常，进程优先级和以下因素有关系：

- 作业开始时的静态优先数：**作业的优先数取决于作业的重要程度、用户为作业运行时所付出的价格和费用大小、作业的类型等因素。
- 进程的类型：**一般系统进程的优先数大于用户进程的优先数；I/O型进程的优先数大于CPU型进程的优先数。这些都是为了充分发挥系统I/O设备的效能。

c. **进程所使用的资源量**：使用CPU的时间越多，优先级越低。对其他资源使用的情况也类似的考虑。

d. **进程在系统中的等待时间**：等待时间越长，进程优先级越高。

各系统处于不同的考虑，有不同的优先数计算公式。这些公式主要来自于时间经验。

九、前台进程、后台进程、守护进程

前台与后台本身是指舞台表演的前台与后台，前台是观众们可以看见的，观众无法看见后台的内容。

这里的前台进程与后台进程同样是这个意思。

1. 前台进程：是直接为用户交互的，用户是可以直接看到的。

2. 后台进程：用户无法直接看到，前台进程与后台进程是可以相互转换的。这一概念在Android系统用的比较多，用户常常需要设置程序在后台运行，不被系统杀掉。自启动等，方便及时接受消息通知。

3. 守护进程：是一种特殊的后台进程，它独立于控制终端并且周期性地执行某种任务或等待处理某些发生的事件，它只能在后台运行。

常用的控制台命令：

`ctrl+z` 将任务放到后台去，并暂停(挂起)。

`ctrl+c` 终止进程（中断）。