## 进程

### 起源

在批处理系统中，每个程序执行完成后，另一个程序才会开始执行。这样的模型，程序不会交错执行，系统设计较为简单。当一个程序执行完，操作系统会自动调取另一个程序执行。在批处理系统中，我们也可以将运行中的程序称为进程。只不过系统资源会一直属于某一个进程，直到该进程执行完毕。这样系统就没必要保存进程的上下文环境，因为在进程的整个生命周期中上下文环境不会变。不像多道程序操作系统，进程会交替使用系统资源，这样就必须保存进程的资源使用状态，也必须精心为进程分配资源。

为了充分利用CPU，第二代操作系统增加了多道程序设计，使得多个程序可以交替执行。那么，当一个程序出现阻塞时，还没执行完毕，操作系统会将其移除CPU，让另一个程序执行。那么，在同一时刻就存在运行中的程序和处于暂停状态的程序。当把程序暂停时，不只是仅仅停止这个程序那么简单，还需要保存该程序运行时的数据和资源（例如PC值、寄存器值和内存分配空间等），这样才可以从它暂停的地方重新开始。那么也就是说，运行中的程序不只是其代码还有其所使用数据和资源等。因为，为了管理运行中的程序引出了进程的概念。进程就是指程序代码和程序所使用的资源。

在批处理系统和多道程序系统中都有进程的概念，只不过进程的概念在多道程序系统中才被提出。批处理系统中的进程管理十分简单，只有开始和停止；而多道程序系统中，进程的管理则更加复杂，操作包含创建、调度、终止和通信，其中调度是进程管理核心。

### 概念

进程的直白定义：运行中的程序。详细定义：由程序代码和运行所需要的资源组成的实体。当程序被载入内存并分配了执行资源，进程便生成了。

### 实现

进程是由进程控制块（Process Control Block）实现。进程控制块包含进程标识符、状态、打开的文件、父进程标识符、兄弟进程、子进程标识符、程序计数器和寄存器值等数据。

将进程切出时，将进程的上下文环境数据保存到进程控制块；将进程切入时，再从进程控制块取出上下文环境数据还原切出时的状态。上下文环境数据包含程序计数器和寄存器值等数据。

### 操作

#### 创建

使用fork()函数创建子进程。系统中的进程都是由父进程创建子进程的形式创建的，1号进程除外。

#### 终止

使用exit()函数终止进程。

#### 调度

当一个进程处于阻塞时，操作系统就调读另一个进程执行。

#### 通信

多个进程之间可以通信。