七天学会

可以感知和控制自身进程的运行环境和状态，也可以创建子进程并与其协同工作，这使得 可以把多个程序组合在一起共同完成某项工作，并在其中充当胶水和调度器的作用。本章除了介绍与之相关的 内置模块外，还会重点介绍典型的使用场景。

开门红

我们已经知道了 自带的 模块比较基础，把一个目录里的所有文件和子目录都拷贝到另一个目录里需要写不少代码。另外我们也知道，终端下的 命令比较好用，一条 命令就能搞定目录拷贝。那我们首先看看如何使用 调用终端命令来简化目录拷贝，示例代码如下：

从以上代码中可以看到，子进程是异步运行的，通过回调函数返回执行结果。

走马观花

我们先大致看看 提供了哪些和进程管理有关的 。这里并不逐一介绍每个 的使用方法，官方文档已经做得很好了。

官方文档：

任何一个进程都有启动进程时使用的命令行参数，有标准输入标准输出，有运行权限，有运行环境和运行状态。在 中，可以通过 对象感知和控制 自身进程的方方面面。另外需要注意的是， 不是内置模块，而是一个全局对象，因此在任何地方都可以直接使用。

官方文档：

使用 模块可以创建和控制子进程。该模块提供的 中最核心的是 ，其余 都是针对特定使用场景对它的进一步封装，算是一种语法糖。

官方文档：

模块是对 模块的进一步封装，专用于解决单进程 服务器无法充分利用多核 的问题。使用该模块可以简化多进程服务器程序的开发，让每个核上运行一个工作进程，并统一通过主进程监听端口和分发请求。

应用场景

和进程管理相关的 单独介绍起来比较枯燥，因此这里从一些典型的应用场景出发，分别介绍一些重要 的使用方法。

如何获取命令行参数

在 中可以通过 获取命令行参数。但是比较意外的是， 执行程序路径和主模块文件路径固定占据了 和 两个位置，而第一个命令行参数从 开始。为了让 使用起来更加自然，可以按照以下方式处理。

如何退出程序

通常一个程序做完所有事情后就正常退出了，这时程序的退出状态码为 。或者一个程序运行时发生了异常后就挂了，这时程序的退出状态码不等于 。如果我们在代码中捕获了某个异常，但是觉得程序不应该继续运行下去，需要立即退出，并且需要把退出状态码设置为指定数字，比如 ，就可以按照以下方式：

如何控制输入输出

程序的标准输入流（ ）、一个标准输出流（ ）、一个标准错误流（ ）分别对应 、 和 ，第一个是只读数据流，后边两个是只写数据流，对它们的操作按照对数据流的操作方式即可。例如， 可以按照以下方式实现。

如何降权

在　　系统下，我们知道需要使用　　权限才能监听　　以下端口。但是一旦完成端口监听后，继续让程序运行在　　权限下存在安全隐患，因此最好能把权限降下来。以下是这样一个例子。

上例中有几点需要注意：

如果是通过　　获取　　权限的,运行程序的用户的　　和　　保存在环境变量　　和　　里边。如果是通过　　方式获取　　权限的，运行程序的用户的　　和　　可直接通过　　和　　方法获取。

　　和　　方法只接受　　类型的参数。

降权时必须先降　　再降　　，否则顺序反过来的话就没权限更改程序的　　了。

如何创建子进程

以下是一个创建　　子进程的例子。

上例中使用了 方法，该方法支持三个参数。第一个参数是执行文件路径，可以是执行文件的相对或绝对路径，也可以是根据 环境变量能找到的执行文件名。第二个参数中，数组中的每个成员都按顺序对应一个命令行参数。第三个参数可选，用于配置子进程的执行环境与行为。

另外，上例中虽然通过子进程对象的 和 访问子进程的输出，但通过 字段的不同配置，可以将子进程的输入输出重定向到任何数据流上，或者让子进程共享父进程的标准输入输出流，或者直接忽略子进程的输入输出。

进程间如何通讯

在 系统下，进程之间可以通过信号互相通信。以下是一个例子。

在上例中，父进程通过 方法向子进程发送 信号，子进程监听 对象的 事件响应信号。不要被 方法的名称迷惑了，该方法本质上是用来给进程发送信号的，进程收到信号后具体要做啥，完全取决于信号的种类和进程自身的代码。

另外，如果父子进程都是 进程，就可以通过 （进程间通讯）双向传递数据。以下是一个例子。

可以看到，父进程在创建子进程时，在 字段中通过 开启了一条 通道，之后就可以监听子进程对象的 事件接收来自子进程的消息，并通过 方法给子进程发送消息。在子进程这边，可以在 对象上监听 事件接收来自父进程的消息，并通过 方法向父进程发送消息。数据在传递过程中，会先在发送端使用 方法序列化，再在接收端使用 方法反序列化。

如何守护子进程

守护进程一般用于监控工作进程的运行状态，在工作进程不正常退出时重启工作进程，保障工作进程不间断运行。以下是一种实现方式。

可以看到，工作进程非正常退出时，守护进程立即重启工作进程。

小结

本章介绍了使用 管理进程时需要的 以及主要的应用场景，总结起来有以下几点：

使用 对象管理自身。

使用 模块创建和管理子进程。