**父进程和子进程**

子进程是父进程的复制品

Unix/linux系统中的进程创建是这样的，理解起来有点绕。

**父进程先执行fork()系统调用，这个调用的结果是系统中多出了一个跟父进程内容完全一样的进程,这个新进程被称为子进程，当然该进程的PCB中父进程指针是指向第一个进程的。**

**前后两个进程各自有自己的地址空间，**形式上有点像把一个文件拷贝了一个副本。虽然资源也相互独立，但拷贝时父进程执行过程已生成的数据，子进程也拷了一份。说简单点像**一个执行到半路的程序突然在系统中多出了一个孪生兄弟，什么都跟自己一样，但要管自己叫老爸。**

当然这样的简单复制本身是没什么用处的。要让它发挥作用，**还需要再执行 exec( B )系统调用，这个调用可以让当前进程转而执行另一个可执行代码（一个新的程序）。**简单的说进程本来在执行A程序，一旦执行到这个调用，就转而开始执行B程序。

至此，**父子两进程就变的不一样了，**但不管它们各自执行的什么代码，其父子关系不会改变，**在父进程中可以使用子进程的进程ID（在执行fork()时的返回值中得到）来中止子进程的执行。**当然子进程也可以因为自己的执行程序结束而终止执行

**父进程和子进程先后执行的问题，**是这样的，在fork之后，**是父进程先执行，然后一个时间片到达之后就是子进程再执行了。**

**每一个子进程都有一个父进程，当进程终止或者结束的时候，都会给父进程发送一个SIGCHLD信号，系统默认是父进程忽略这个信号，如果父进程希望被告知其子进程的这种状态改变，则应该捕获这个信号，捕捉函数一般是wait函数来取得子进程ID和子进程状态。**

对于键盘上的Ctrl+按键的操作，一般是产生一个信号，然后进程捕捉这个信号。另外当然我们可以直接使用这些信号，通过kill命令，把信号发给相应的其他进程。

wait函数是父进程等待子进程结束，也就是说当子进程结束的时候会发送给父进程一个信号SIGCHID，这时候父进程通过wait函数接收到这个信号，这时候父进程就知道子进程结束了。这个正好用在shell解析器的编写里面，shell解析器作为父进程，而命令行命令作为子进程，当子进程结束的时候就会告诉父进程，这时候父进程就可以提示输入下一个命令了。

**一个进程只能为他自己或者他的子进程设置进程组ID，在他的子进程调用了exec函数之后，就不能再改变该子进程的进程组ID了。**

Linux进程管理对于电脑使用的玩家的常用软件，然后我就学习及深入的研究Linux进程管理，在这里和大家一起探讨Linux进程管理的使用方法，希望对大家有用。程序是为了完成某种任务而设计的软件，比如OpenOffice是程序。

　　什么是进程呢？**进程就是运行中的程序。一个运行着的程序，可能有多个进程。 比如 LinuxSir.Org 所用的WWW服务器是apache服务器，当管理员启动服务后，可能会有好多人来访问，也就是说许多用户来同时请求httpd服务，apache服务器将会创建有多个httpd进程来对其进行服务。**