**exec函数簇详解**

**作者：chw**

**============================================================================**

**由于本人能力有限，文档中难免有错误。请发消息chw@farsight.cn给发提醒！谢谢！**

**============================================================================**

1. **简介**

**我们在学习fork()函数的时候，我们知道它是用于创建一个子进程，该子进程几乎是父进程的副本，而有时我们希望子进程去执行另外的程序，exec函数族就提供了一个在进程中启动另一个程序执行的方法。它可以根据指定的文件名或目录名找到可执行文件，并用它来取代原调用进程的数据段、代码段和堆栈段，在执行完之后，原调用进程的内容除了进程号外，其他全部被新程序的内容替换了。另外，这里的可执行文件既可以是二进制文件，也可以是Linux下任何可执行脚本文件。**

**我们的exec函数簇一般使用在以下两种情况中：**

**(1)当进程认为自己不能再为系统和用户做出任何贡献时，就可以调用任何exec 函数族让自己重生。**

**(2)如果一个进程想执行另一个程序，那么它就可以调用fork函数新建一个进程，然后调用任何一个exec函数使子进程重生。**

**fork创建一个新的进程就产生了一个新的PID，exec启动一个新程序，替换原有的进程，因此这个新的被 exec 执行的进程的PID不会改变(和调用exec的进程的PID一样)。**

1. **exec函数簇简介**

**<1>头文件**

**#include <unistd.h>**

**<2>函数原型**

**extern char \*\*environ;**

**int execl(const char \*path, const char \*arg, ...);**

**int execlp(const char \*file, const char \*****arg, ...);**

**int execle(const char \*path, const char \*arg, ..., char \* const envp[]);**

**int execv(const char \*path, char \*const argv[]);**

**int execvp(const char \*file, char \*const argv[]);**

**int execve(const char \*path, char \*const argv[], char \*const envp[]);**

**返回值：成功返回0，失败返回-1。**

**参数：**

**@path 含有路径名的文件**

**@arg 可执行程序的名字...**

**arg1 相应的参数.... 最后以NULL结尾。**

**@file 可执行文件的名字**

**@environ exec定义的全局变量。可改变当前进程的环境。**

**@envp 给environ传递的改变进程环境的内容。（eviron = &envp[0]）**

**这6 个函数在函数名和使用语法的规则上都有细微的区别，下面就可执行文件查找方式、参数表传递方式及环境变量这几个方面进行比较说明。**

**① 查找方式：上表其中前4个函数的查找方式都是完整的文件目录路径，而最后2个函数（也就是以p结尾的两个函数）可以只给出文件名，系统就会自动从环境变量“$PATH”所指出的路径中进行查找。**

**② 参数传递方式：exec函数族的参数传递有两种方式，一种是逐个列举的方式，而另一种则是将所有参数整体构造成指针数组进行传递。**

**在这里参数传递方式是以函数名的第5位字母来区分的，字母为“l”（list）的表示逐个列举的方式，字母为“v”（vertor）的表示将所有参数整体构造成指针数组传递，然后将该数组的首地址当做参数传给它，数组中的最后一个指针要求是NULL。读者可以观察execl、execle、execlp的语法与execv、execve、execvp的区别。**

**③ 环境变量：exec函数族使用了系统默认的环境变量，也可以传入指定的环境变量。这里以“e”（environment）结尾的两个函数execle、execve就可以在envp[]中指定当前进程所使用的环境变量替换掉该进程继承的所以环境变量。**

|  |  |
| --- | --- |
| 后缀 | 操作能力 |
| l | 希望接收以逗号分隔的参数列表，列表以NULL指针作为结束标志 |
| v | 希望接收到一个以NULL结尾的字符串数组的指针 |
| p | 是一个以NULL结尾的字符串数组指针，函数可以DOS的PATH变量查找子程序文件 |
| e | 函数传递指定参数envp，允许改变子进程的环境，无后缀e时，子进程使用当前程序的环境 |

**实际例子详解：**

**execl("/bin/ls","ls","-l",NULL);**

**execlp("ls","ls","-l",NULL);**

**execle("ls","ls","-l",NULL,NULL);**

**------------------------------------------------------------------**

**char \*args[] = {"ls","-l",NULL};**

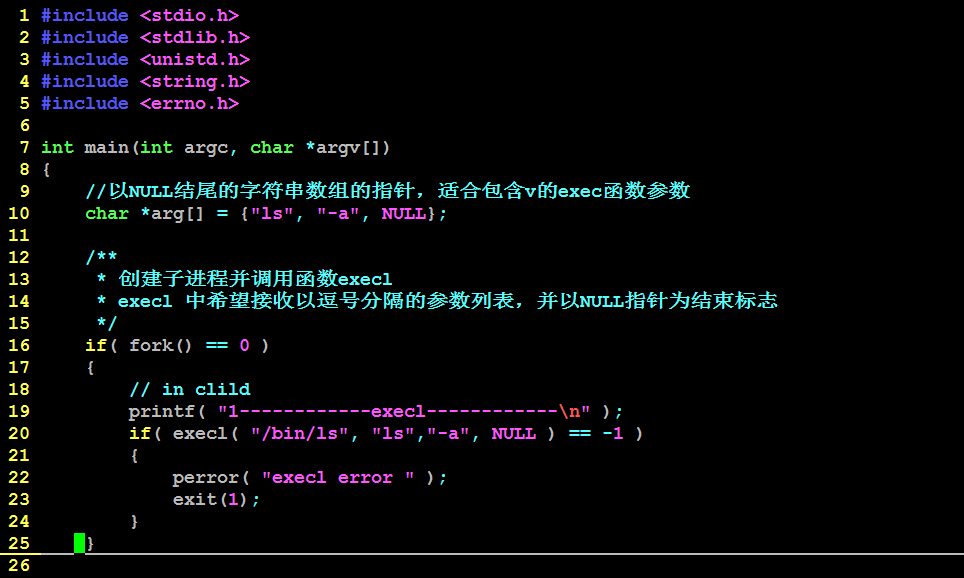
**execv("/bin/ls",args);**

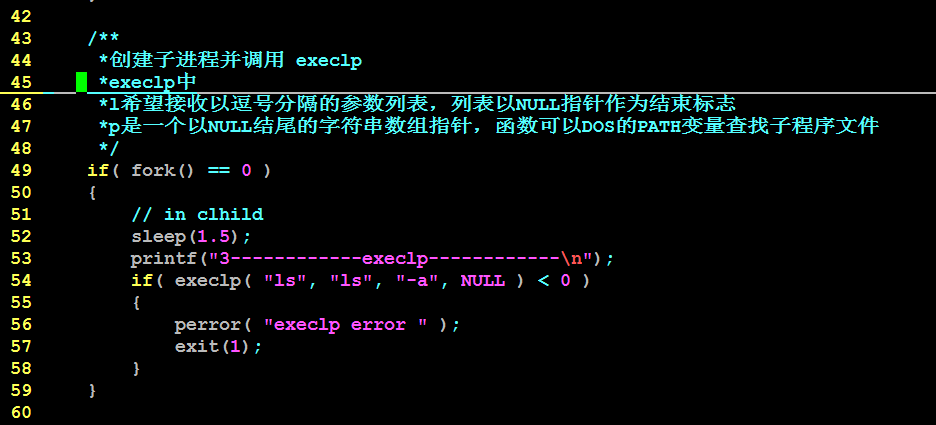
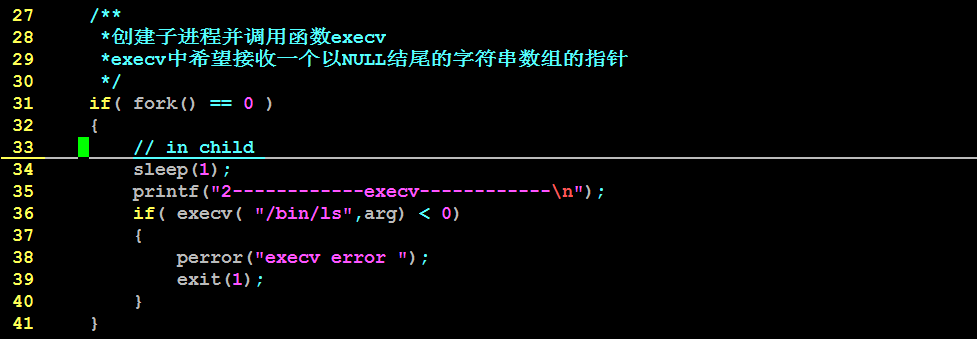
**execvp("ls",args);  //把路径添加到path环境变量中。**

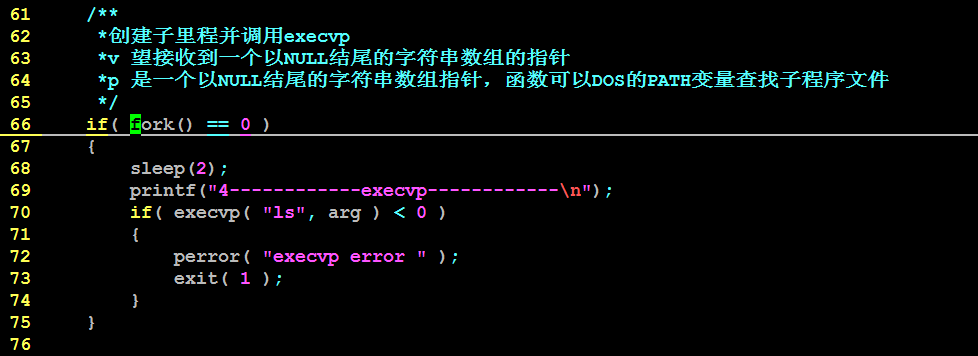
**execve("ls",args,NULL);**

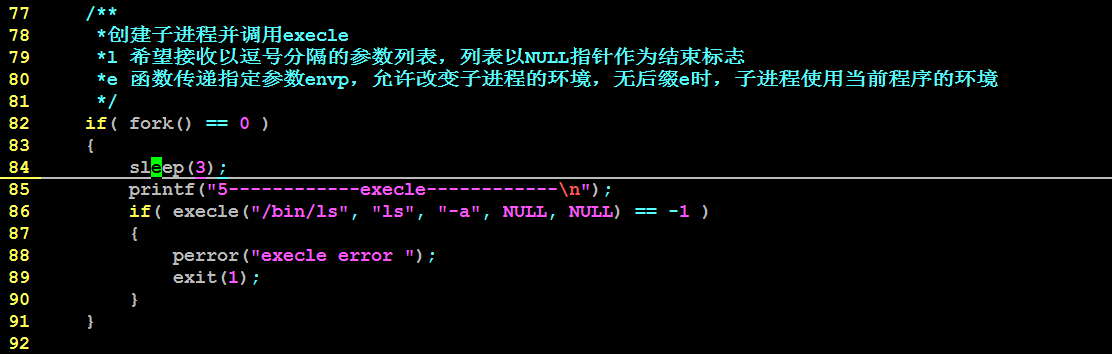
1. **代码详解**

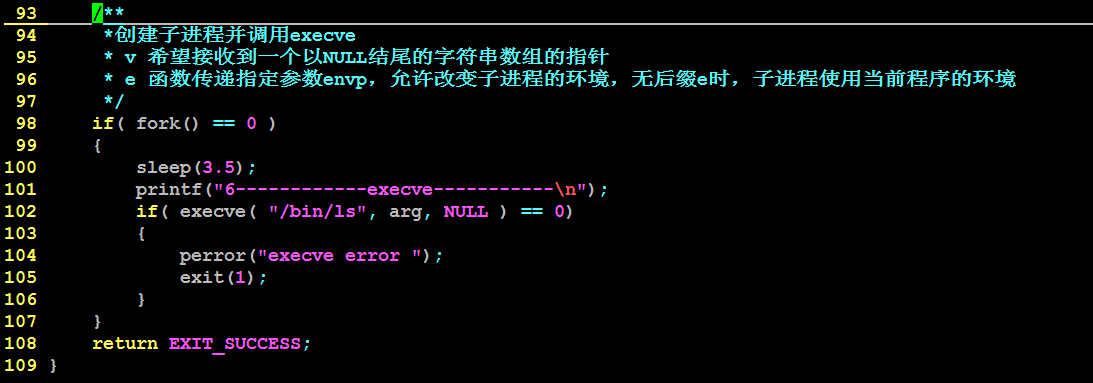
**exec.c**



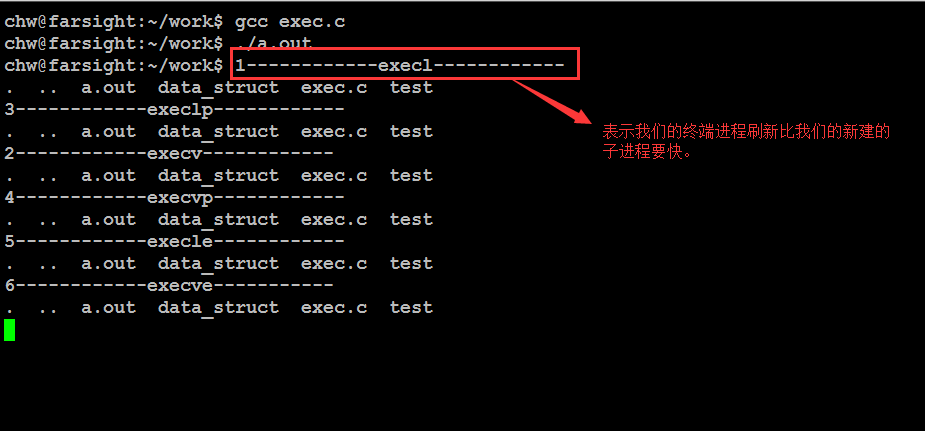








**运行结果：**



**注：由于每个子进程谁先启动不清楚。故顺序混乱属于正常现象。**