

linux启动后，会默认打开3个文件描述符，

分别是：标准输入standard input 0,正确输出standard output 1,错误输出：error output 2

以后打开文件后。新增文件绑定描述符 可以依次增加。 一条shell命令执行，都会继承父进程的文件描述符。因此，所有运行的shell命令，都会有默认3个文件描述符。

文件输入输出由追踪为一个给定的进程所有打开文件的整数句柄来完成。这些数字值就是文件描述符。最为人们所知的文件描述符是 *stdin*, *stdout* 和 *stderr*, 文件描述符的数字分别是0, 1和2。这些数字和各自的设备是保留的。一个命令执行前，先会准备好所有输入输出，默认分别绑定 (*stdin*,*stdout*,*stderr*)，如果这个时候出现错误，命令将终止，不会执行。

1. 标准输入 (*stdin*)：代码为 0，使用 < 或 << ; */dev/stdin -> /proc/self/fd/0* 0代表：/dev/stdin
2. 标准输出 (*stdout*)：代码为 1，使用 > 或 >> ; */dev/stdout -> /proc/self/fd/1* 1代表：/dev/stdout
3. 标准错误输出(*stderr*)：代码为 2，使用 2> 或 2>> ; */dev/stderr -> /proc/self/fd/2* 2代表：/dev/stderr

说明： *stdin*, *stdout*, *stderr* 都是 FILE \* 类型， *STDIN\_FILENO*, *STDOUT\_FILENO*,*STDERR\_FILENO* 是它们对应的系统文件描述符， *STDIN\_FILENO* 默认指向 输入设备（键盘）， *STDOUT\_FILENO*和 *STDERR\_FILENO* 默认指向 输出设备（显示器）

*stdin*等属于标准I/O，高级的输入输出函数。在<stdio.h>。

*STDIN\_FILENO*等是文件描述符，是非负整数，一般定义为0, 1, 2，直接调用系统调用，在<unistd.h>。

```
#define STDIN_FILENO 0
#define STDOUT_FILENO 1
#define STDERR_FILENO 2
```

对于*stdin*等可以使用*fileno()*函数(用来取得参数stream指定的文件流所使用的文件描述符)来取得该文件流对应的文件描述符。

```
fileno(stdin)
= STDIN_FILENO = 0
```

```
fileno(stdout)
```

```
= STDOUT_FILENO = 1
```

```
fileno(stderr)
```

```
= STDERR_FILENO = 2
```