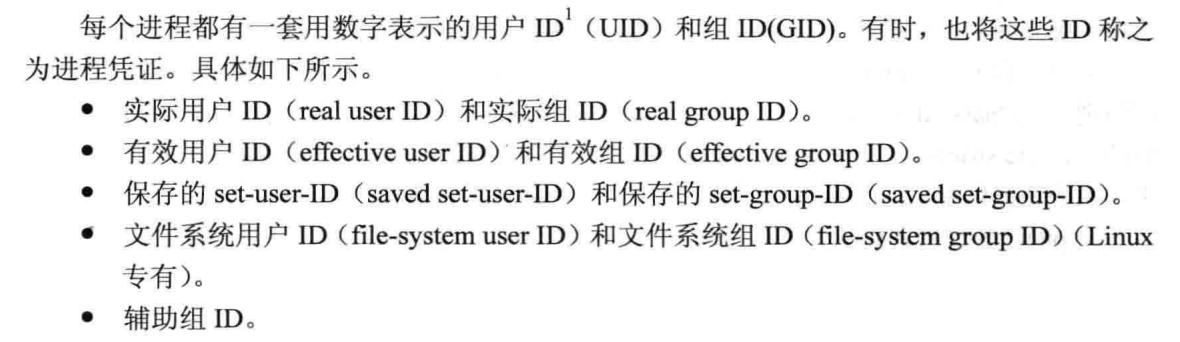
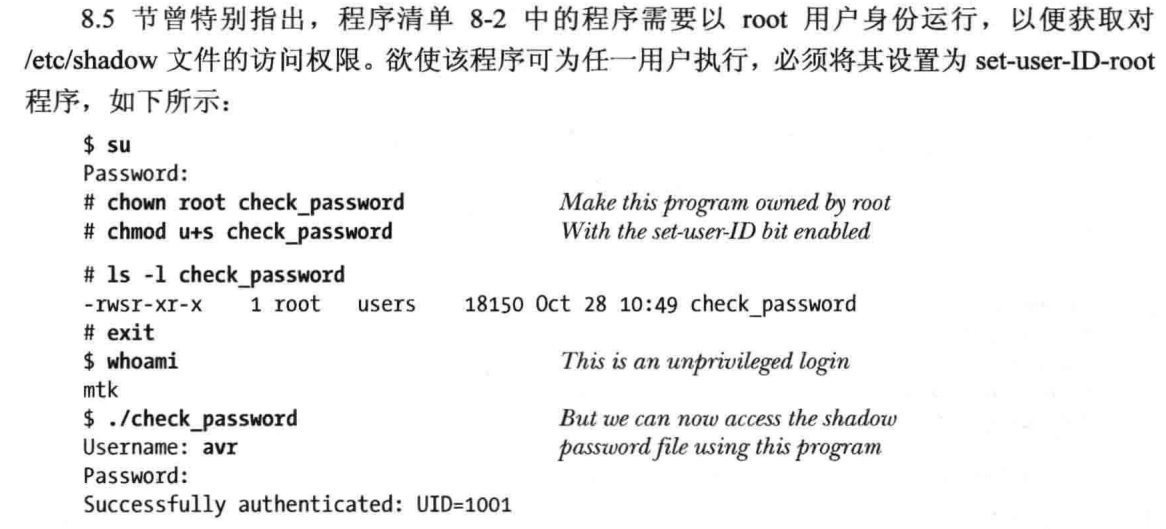
第九章

1. 进程凭证：

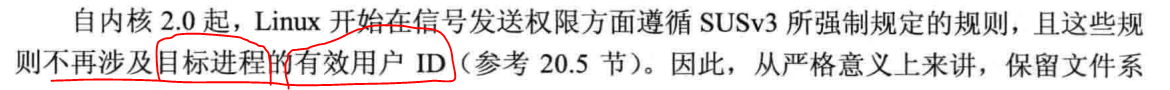


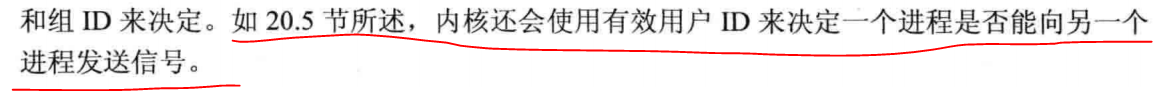
1. 实际用户ID和实际组ID：这两个确定了进程的所属，来源于登录时读取/etc/passwd文件的内容。
2. 有效用户ID和有效组ID：当进程尝试执行各种操作的时候，将结合有效用户ID和有效组ID，辅助ID来判断授予进程权限。一般而言，有效用户ID和有效组ID和实际ID相同，但是有两种方法能够让两者不一样，一种是系统调用，二是执行set-user-id程序或者set-group-id程序。
3. set-user-id程序的定义：设置了权限位的可执行程序。一般这种程序使用ls -l看到x类型变为s。当运行这种程序时，内核会将进程的有效ID设置为可执行文件的用户ID从而获得额外的权限。



对于上面的问题，文件的x设置为s时不成功的，仍旧是x标志。

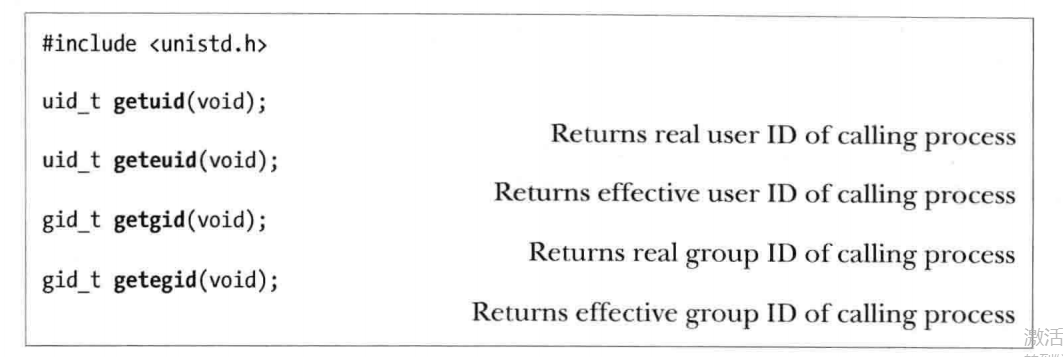
1. saved set-user-id 和 saved set-group-id的相关内容：在执行程序的时候，会将有效id复制到saved set-user-id或saved set-group-id里面。
2. 文件系统ID和组ID：一般情况下这两个和有效用户ID和有效组ID是一毛一样的，只是由于历史原因才出现文件系统ID。但是从描述里遇到了一个问题：



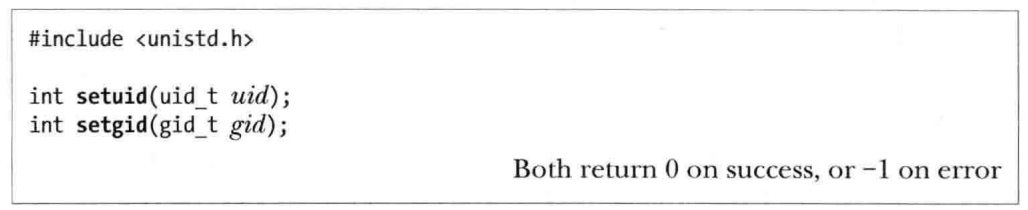


这两个描述是否有矛盾，还需要查看一下。

1. 辅助组ID：从系统组文件中获取的辅助组ID。
2. Linux系统特有的/proc/PID/status文件里面Uid，Gid各行分别代表的内容是实际，有效，保存设置，文件系统ID。
3. 获取实际和有效ID：

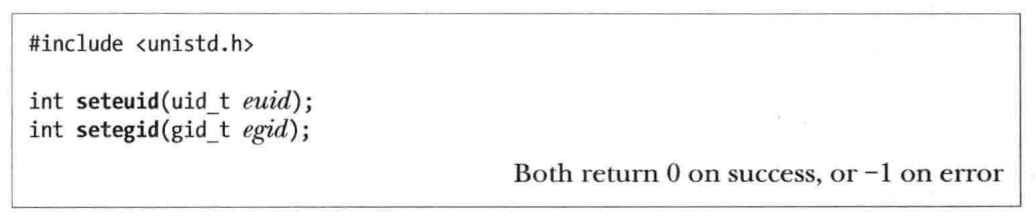


1. 修改有效ID：



非特权进程调用setuid的时候，仅能修改有效ID，而且仅当执行set-user-id程序的时候才发挥作用。

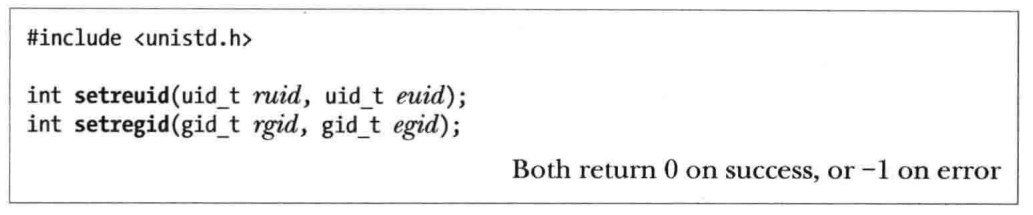
当特权用户调用时，实际用户ID，有效用户ID，saved set-user-id均改变，这是单向的，执行后没有特权。组ID则不同。



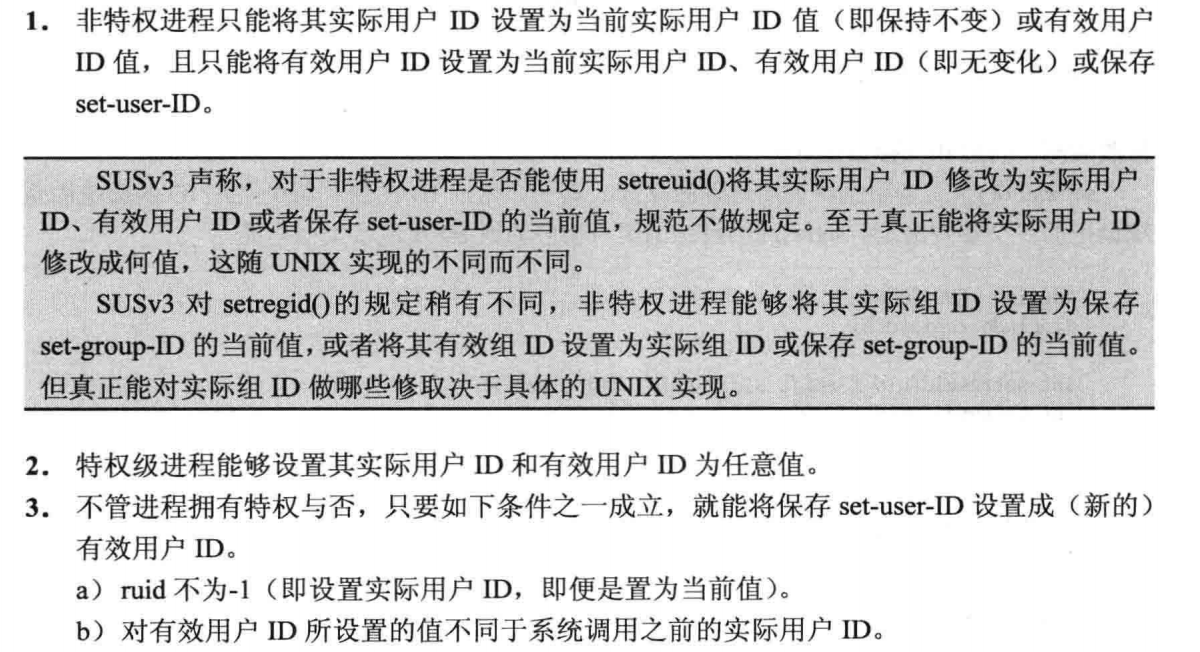
非特权进程如上效果。

特权进程可以修改为任意值。

1. 修改实际ID和有效ID：

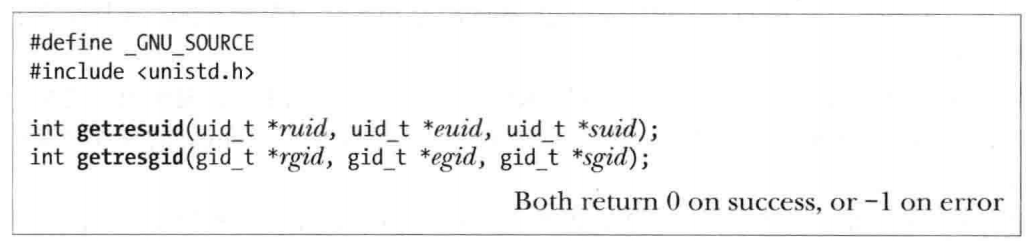


规则如下：

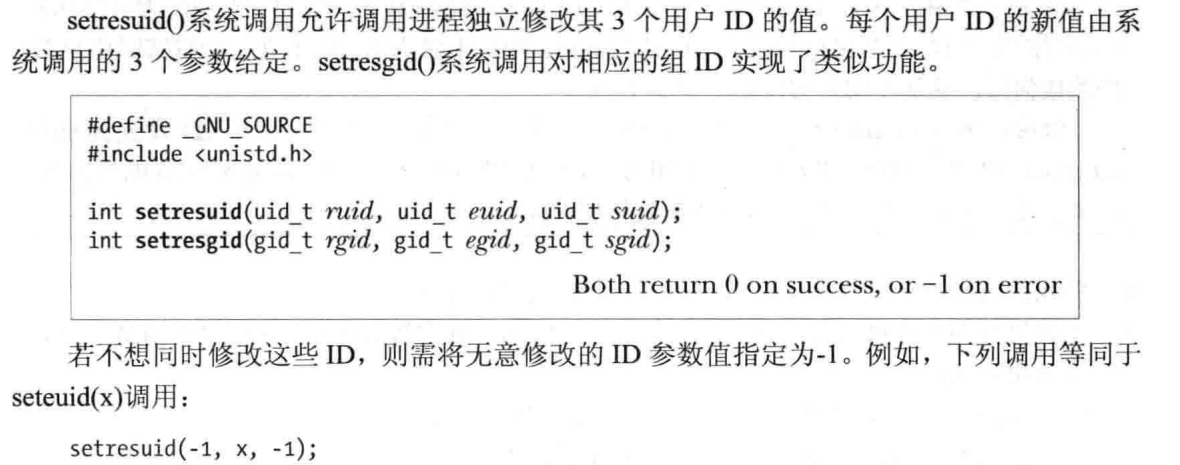


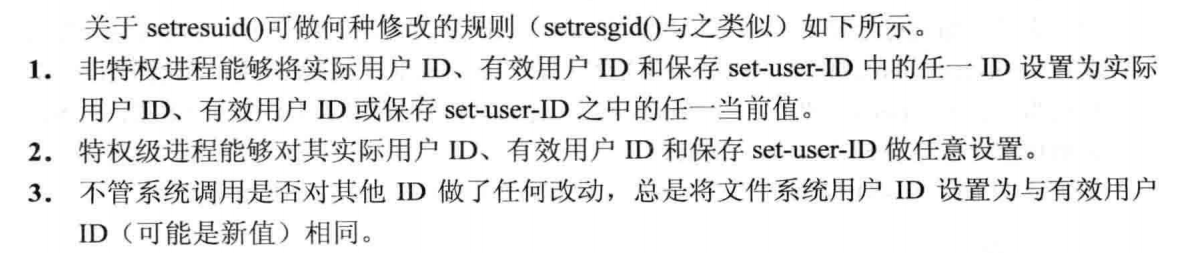
通过这个可以看到，saved set-user-id的作用是收放特权时不用为了改变有效ID而改变实际用户ID。

1. 获取实际，有效，保存的ID：

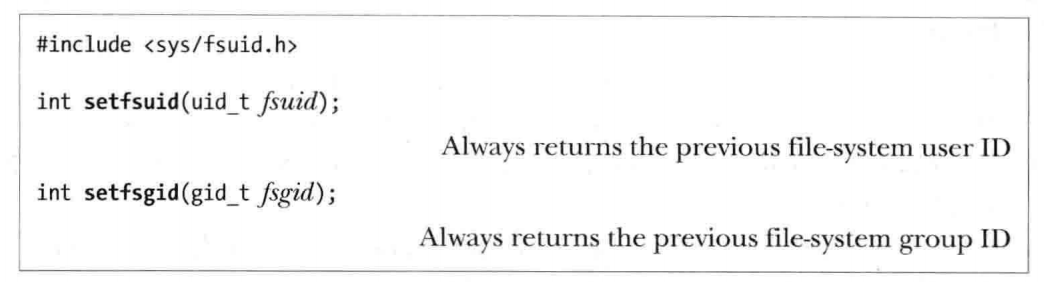


1. 修改实际，有效，保存ID：

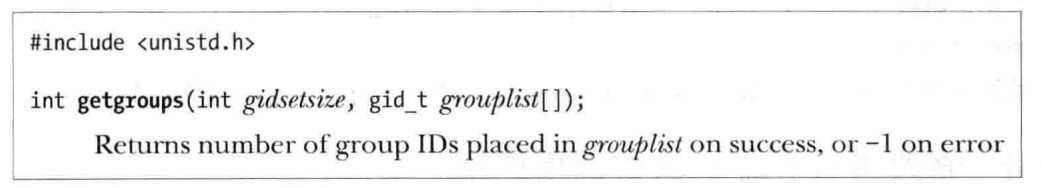


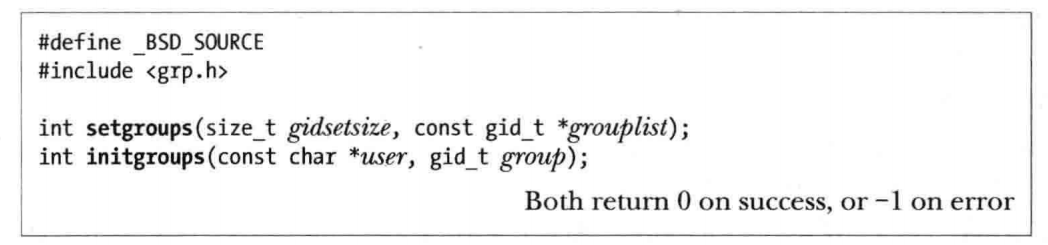


1. 获取和修改文件系统ID：



1. 获取和修改辅助组ID：





1. 总结如下

