****一、Set UID****   
当s这个标志出现在文件所有者的x权限上时，此时就被称为Set UID，简程SUID。

例如:



在文件所有者的x权限上都出现了s权限，那么这个特殊权限的特殊性的作用是什么呢？

1、SUID权限仅对二进制程序有效。

2、执行者对于该程序需要具有x的可执行权限。

3、本权限仅在执行该程序的过程中有效。

4、执行者将具有该程序拥有者的权限。

SUID的关键作用体现在第四句，SUID的目的就是：让本来没有相应权限的用户运行这个程序时，可以访问他没有权限访问的资源。

例如：

1. 大家都知道su这个命令，无论任何人，执行了它都将具有root权限，因为他的拥有者是root，而且这个权限仅在执行的su命令中有效，不管是文件拥有者，文件拥有者所属组，还是其他人，都是具有x权限的，所以都可以执行该程序，执行之后就将具有该程序拥有者的权限，即root的权限，这也就是su命令能够切换用户权限的实现原理。

在su命令执行期间, 是可以获得root用户的权限的, 假如将su kill掉, 则直接退出root用户, 不再拥有root用户的权限了.

2、系统中的用户密码是保存在/etc/shadow, 同组用户和其他人是没有写权限的,



关键是要把密码写入到/etc/shadow中。我们知道，除了root用户能修改密码外，用户自己同样也能修改密码，为什么没有写入权限，还能修改密码，就是因为这个SUID功能。从截图2可以看到，passwd的文件拥有者，文件拥有者所属组，还是其他人，都是具有x权限的，所以都可以执行passwd，当执行了之后获得root的权限也就能进行密码的修改。

注：这个SUID只能运行在二进制的程序上（系统中的一些命令），不能用在脚本上，同样也不能放到目录上，放上也是无效的。

****二、Set GID****   
当s出现在文件的所属用户组x权限位置上时，就是SGID，相对与SUID来说，SGID还是比较少见的。   
那么SGID的功能是什么呢？和SUID一样，只是SGID是获得该程序所属用户组的权限。   
这相SGID有几点需要我们注意：

1、SGID对二进制程序有用；

2、程序执行者对于该程序来说需具备x的权限；

3、SGID主要用在目录上；

理解了SUID，我想SGID也很容易理解。如果用户在此目录下具有w权限的话，若使用者在此目录下建立新文件，则新文件的群组与此目录的群组相同。

**三、Sticky Bit**

这个就是针对others来设置的了，和上面两个一样，只是功能不同而已。

SBIT（Sticky Bit）目前只针对目录有效，对于目录的作用是：当用户在该目录下建立文件或目录时，仅有自己与 root才有权力删除。

最具有代表的就是/tmp目录，任何人都可以在/tmp内增加、修改文件（因为权限全是rwx），但仅有该文件/目录建立者与 root能够删除自己的目录或文件。

注：SBIT对文件不起作用。

四、SUID/SGID/SBIT权限设置

和rwx权限一样，s、t也有两种设置方法：

1、文字法 ：SUID: u+s ，SGID: g+s，SBIT: o+t

2、数字法：将原来的三位数扩展为四位数即可，SUID为4，SGID为2，SBIT为1，把它们放在权限数字的最开头。例如设置SUID，可以写成4777，设置SGID可以写成，2777

文章参考: https://blog.csdn.net/imkelt/article/details/53054309