rwx

1 001

2 010

3 011

4 100

5 101

6 110

7 111

umask一般是用在你初始创建一个目录或者文件的时候赋予他们的权限。

这里要说明两点：

1、针对目录来说x权限代表可以进入该目录，所以说对于这个权限初始赋值是没什么问题的;

2、针对文件的x的权限代表执行，这个风险太高，所以一般权限初始赋值必须去掉x的;

[root@www ~]# umask

0022

这四个字母代表什么?

首先说明一点，上面四个数字代表是赋值初始化准备丢弃的权限。(相对应文件来说，x权限就算没说明出来丢弃一样必须默认丢弃)

第一个0代表suid丢弃的权限;

第二个0代表本文件/目录拥有者什么权限都没丢弃(如果是文件，x权限除外，为什么看看上面解释);

第三个2代表本文件/目录的用户组丢弃了w权限(如果是文件那么它的x权限也丢弃);

第四个2代表本文件/目录的其他用户能使用的权限只有有r和x(文件除外)。

综合上面所说如果你新创建一个文件那么它的权限是：-wr-w--w-- 即是：644

如果你新创建一个目录的话那么它的权限是：dwrxw-xw-x 即是：755

总而言之是在默认状态666和777的状态上进行判断和修改，最终确定新建文件的读取权限的一种码，注意是八进制！以0开头，但是linux的shell可以不写0

不设置umask时，创建文件，相当于设置为0001的情况。

umask的默认值为0002

我们使用如下的简单的方法快速计算。

对于目录，直接使用777-umask即可，就得到了最终结果。

对于文件，先使用666-umask。

　　如果对应位上为偶数：最终权限就是这个偶数值。

　　如果上面的对应为上有奇数，就对应位+1。

上面的这个方法计算是非常方便的， 为何得到奇数要+1呢。

文件的最大权限是666，都是偶数，你得到奇数，说明你的umask有奇数啊，读为4，写为2，都是偶数，说明你有执行权限的。

就按照上面的umask=023为例，在计算其他用户权限的时候6-3=3 ，6是读写，3是写和执行，其实应该是读写权限减去读权限的得到写权限的，相当于我们多减去了一个执行权限。所以结果加1。