5 权限管理
fchao.site

1、文件权限的匹配规则
- UID ——> GID ——>Other：按照匹配的身份具有的权限进行访问
- root 用户无视文件权限的匹配规则

2.查看文件权限（ll）
-rw-r--r--. 1 root root 0 Jul 14 14:41 zhangsan.txt
- 第一个字符-表示文件类型
  - - 表示普通文件
  - d 表示目录(Directory)
  - l 表示符号链接(Link)
  - c 表示字符设备文件(Character)
  - b 表示块设备文件(Block)
  - s 表示套接字文件(Socket)
  - p 表示管道文件(Pipe/FIFO)
- rw-：文件的拥有人权限
- r--：文件的拥有组权限
- r--：文件的其他人权限
  - rw-r--r--：剩余的九个字符表示文件或者目录的权限
  - r：表示读取权限（readable）
  - w：表示写入权限，删除也需要有w权限（writable）
  - x：表示执行权限（executable）
  - -：表示没有对应权限
- .：seLinux的安全标记，表示文件受到seLinux的保护
- 1：表示文件的硬链接数
- root：文件的UID也就是拥有人
- root：文件的GID也就是拥有组
- 0：表示文件的大小，如果为0则表示是一个空文件
- Jul 14 14:41: 文件最后一次被修改的时间
- zhangsan.txt: 文件名

权限对文件和目录的作用

	文件	目录
R读	可以读取文件的内容	可以查看目录下的文件和子目录(列出文件)
W写	修改或编辑文件的内容	可以在目录下创建或者删除文件和子目录
X执行	可以将文件作为命令或脚本执行	可以进入到目录下

目录最少需要有rx权限，表示用户可以进入到该目录下，正常的读取到文件

3.数字赋权
四位数，后三位的每一位都是拥有人、拥有组、其他人r+w+x 的和，第一位为特殊权限
- R:4 W:2 X:1

4.权限管理
- 修改文件和目录的权限 **chmod**
  - 字符方式赋权：拥有人u、拥有组g、其他人o、所有人a
    - 加减赋权
      - chmod u+r,g+w,o+x filename
      - chmod ugo+r filename
      - chmod +r filename / chmod -x filename 省略a所有人
    - 精确赋权
      - chmod u=r,g=w,o=x filename 会覆盖原有权限
  - 数字方式赋权
    - chmod 644 file.txt
- 修改所有者和所属组 **chown**
  - chown username:groupname filename
  - chown username filename 仅修改拥有人
  - chown :groupname filename 仅修改拥有组
- 仅能修改所属组 **chgrp**
  - chgrp 用户组 文件/目录

均可使用-R递归修改整个文件夹

5.umask
专门控制用户创建内容的默认权限，输入umask查询值

若文件和目录计算umask不同以目录为主
- 目录的默认权限=777 - umask=755 **目录的最大权限是777**
- 文件的默认权限=666 - umask=644 **文件的最大权限是666**
- 计算文件权限时，umask奇数位在结果加1

设置umask

可以使用条件赋值
- 只影响当前 Shell和所有子进程/子 Shell，当执行exit出当前终端或者重启时失效
  - 输入：umask 值
- 系统所有的用户都生效
  - 写入/etc/bashrc
- 指定用户生效
  - 写入~username/.bashrc

6.特殊权限
suid权限
- 对文件：文件有suid权限，执行这个文件的是以拥有人的权限来执行（不包括脚本）
- 对目录：没有意义
- 设置suid权限（suid对应的数字为4）
  - chmod u+s /usr/bin/touch
  - chmod 4655 /usr/bin/touch 添加
  - chmod 0655 /usr/bin/touch 删除
- 查看拥有人suid权限
  - 小s，说明这个栏位原来有x权限：-rwsr-xr-x
  - 大S，说明这个栏位原来没有x权限：-rwSr--r--

sgid权限
- 对文件：文件有sgid权限，执行这个文件的是以拥有组的权限来执行（不包括脚本）
- 对目录：目录有sgid的权限，那么这个目录下所创建的新的内容会继承这个目录的拥有组，也会继承sgid权限
- 设置sgid权限（sgid对应的数字为2）
  - chmod g+s /usr/bin/touch
  - chmod 2655 /usr/bin/touch 添加
  - chmod 0655 /usr/bin/touch 删除
- 查看拥有组sgid权限
  - 小s，说明这个栏位原来有x权限
  - 大S，说明这个栏位原来没有x权限

sbit/stciky权限
- 对文件：没有意义
- 对目录：如果目录有sbit的权限，那么对于这个目录下的内容来说，谁创建的文件只能谁去删除和修改
- 设置sbit权限（sbit对应的数字为1）
  - chmod o+t /opt
  - chmod 1655 /opt 添加
  - chmod 0655 /opt 删除
- 查看其他人sbit权限
  - 如果是小t，说明这个栏位原来是有x权限
  - 如果是大T，说明这个栏位原来没有x权限

7.Access Control List权限
- 设置acl权限
  - -m 修改（modified）
    - setfacl -m u:zhangsan:rw- file 修改用户zhangsan对file的权限
    - setfacl -m g:zhangsan:rw- file 修改用户组zhangsan对file的权限
    - 不指定用户/用户组，相当于chmod setfacl -m g::rw- file == chmod g=rw file
  - -R -m 递归修改acl权限
  - -d -m 目录下新建文件继承权限
  - chacl 'u::rwx,g::r-x,o::r--,u:bob:r--,m::r-x' myfile.txt 写出完整acl权限字符串进行覆盖式修改

  setfacl [选项] [规则] 文件或目录
- 删除acl权限
  - 删除指定用户的acl权限 — setfacl -x u:zhangsan file
  - 删除文件的所有acl权限 — setfacl -b file
- 查看acl权限
  - ll — 如果文件权限后面有+ 表示有acl权限
  - getfacl file
- acl权限掩码
  - setfacl -m m::rwx file 设置掩码值
  - 与运算，当 ACL 规则和 mask 都允许某个权限时，该权限才会真正生效
- acl权限的优先级 — 拥有人→用户acl权限→拥有组→组acl权限→其他人

8.sudo
- 提权命令
  - -k 强制下次sudo时验证密码
  - -u 指定用户身份，-g指定用户组
  - -l 列出当前用户的sudo提权配置
  - -b 放入后台运行 （和 & 符号一样）
  - -i 以root身份启动一个完整的登录 Shell
- 配置提权
  - 目录
    - /etc/sudoers
    - /etc/sudoers.d/
  - 编辑配置文件
    - visudo
    - sudoedit
  - 格式
    - 提权的用户 主机名=(提权到哪个用户) 提权的命令绝对路径
    - zhangsan rhel9=(ALL) NOPASSWD:ALL — ALL 所有 NOPASSWD: 某命令或全部无需密码
    - %用户组 主机名=(提权到哪个用户) 提权的命令
    - %sudo ALL=(ALL:ALL) NOPASSWD:ALL — (ALL:ALL) 中第一个 ALL 代表可以 以任意用户身份运行 第二个ALL代表可以以任意用户组身份运行
  - 别名写法（可以简化配置文件 别名必须大写）
    - 用户别名 User_Alias USERNUM=zhangsan，lisi，wangwu
    - 主机别名 Host_Alias HOSTNUM=node1，node2，node3
    - 提权到指定用户别名 Runas_Alias USEREXEC=root，user1，user2
    - 执行命令的别名 Cmnd_Alias CMDNUM=/usr/bin/touch，/usr/sbin/useradd
    - 【引用生效】USERNUM HOSTNUM=(USEREXEC) NOPASSWD:CMDNUM

    将配置项单独拆分写出再引用
  - 用 visudo 和 sudoedit 来编辑配置文件