1.GUI: 图形用户界面

CLI： 命令行界面

2.Shell

Shell是一个程序，接收由键盘输入的命令并将其传递给操作系统来执行。**Bash是Shell的一种实现，是Linux中默认的Shell。**

3.终端

打开终端后，会出现类似提示符：

[me@linuxbox ~]$

这是Shell提示符，出现在Shell已经准备好接收输入的时候。提示符格式通常包括 **用户名@计算机名**

如果提示符的最后一个字符**是#而非$**，表示该终端会话具有**超级用户权限**。要么是以root用户登录的，要么终端仿真器提供了超级用户权限。

大部分Linux能够记住最近输入的1000个命令。

4.命令

date 显示时间和日期

cal 显示当前月份的日历

df 查看磁盘的可用空间

free 查看可用内存容量

exit 关闭终端窗口

5.路径名

绝对路径从根目录开始。系统的大多数系统程序安装在/usr/bin目录。

相对路径：**.代表当前工作目录**，..代表父目录。**在大多数情况下，可以忽略./** 。这部分是隐含的，所以cd bin等同于cd ./bin

以点号开头的文件名是隐藏的，ls命令不会列出这些文件，需要用ls -a。创建账户时，主目录中会放置一些用于配置账户的隐藏文件。用户主目录用~表示。

在Linux中，每个用户的主目录通常是以用户名命名的文件夹，例如，如果用户的用户名为“john”，则该用户的主目录可能是“/home/john”。  
主目录是每个用户登录到Linux系统后默认进入的工作目录。当用户登录时，系统会自动将用户的当前工作目录设置为该用户的主目录。这样，用户就可以在自己的主目录下创建、修改和管理自己的文件和文件夹。

Linux中的文件名与命令区分大小写。Linux中没有”文件扩展名“的概念，文件内容或用途是由其他方式来决定的

6.便捷写法

cd 将当前工作目录改为用户主目录/home/用户名

cd -将当前工作目录换回前一个工作目录

cd ~user\_name 将当前工作目录更改为用户user\_name的主目录，如cd ~bob会切换到bob用户的主目录

1. ls命令

ls可以指定要显示内容的目录：

ls /usr #查看/usr目录中的内容

8.选项与参数

大多数命令之后通常会跟随一个或多个能够修改命令行为的选项，接着是一个或多个参数，用于表明操作对象，大多数命令类似于：

command -options arguments

即： **命令 选项 参数(表明操作对象)**

大部分命令使用的选项是在单个字符前加上连字符，如-l。很多命令也支持长选项，即在单词前加两个连字符。而且不少命令还允许将多个短选项串在一起使用，如ls -lt将l和t串在一起。t用于依照文件修改时间对输出结果排序。

加上长选项--reverse，即ls -lt --reverse，降序排列输出结果。

常用的ls命令选项：

-a, --all 列出所有隐藏文件，隐藏文件包括.(当前工作目录)、..(父目录)、以.开头的文件等

-A, --almost-all 与-a类似，但不列出.(当前工作目录)和..(父目录)。

-d, --directory 与-l选项一起使用，查看目录的详细信息，而不是查看目录的内容

-F, --classify 在每个列出的名称后面加上类型指示符，如/表示目录名

-l 以长格式显示输出结果

-h， --human-readable 在长格式(-l)的输出结果中，不再以字节为单位显示文件大小，而是以人类可读的显示文件大小

-r, --reverse 以降序显示输出结果。默认ls按照字母升序显示

-S 按照文件大小排序输出结果

-t 按照修改日期排序输出结果

9.利用file命令确定文件类型

可以用file命令来确定文件类型

**file filename** #输出文件内容的简要描述

使用less命令查看文本文件

**less filename**

less可以查看人类可读的文本文件。很多包含系统设置的文件采用的都是文本格式。

在less中：使用Q可退出less命令。g可以移动到文本文件开头，G可移动到文本文件末尾。左右键可以前翻后翻。/characters向前搜索指定的字符串。n重复上一次搜索。h显示帮助信息。

在终端界面，双击可复制文件名，单击中键可以粘贴到命令中。**reset**命令可以恢复终端中混乱的字符

文本是字符与数字之间的一对一映射关系

1. 长列表

ls -l后会列出文件的属性，如：

**drwxr-xr-x 2 kmc kmc 4896 3月 27 20:15 文档**

第一个字母d表示**文件类型**：

- 普通文件(txt等) d 目录文件 p 管理文件 l链接文件

b 块设备文件 c字符设备文件 s套接字文件

之后的九个字符rwxr-xr-x表示**文件权限**，三个一组。第一组rwx表示文件的所有者的权限，第二组r-x表示所有者所在的组的权限，第三组r-x表示其他用户的权限

文件权限： r 读权限 w写权限 x可执行权限 -无权限

第一组的rwx表示所有者有读的权限、写的权限、执行的权限，第二组表示组内其他成员有读权限、执行权限

之后的数字2对于普通文件来说是链接数，对于目录文件来说是第一级子目录数

之后的kmc和kmc分别表示用户名和组名

之后的数字表示文件的大小，以字节B为单位

之后的时间表示最后修改时间

最后表示文件名

11.链接文件

 lrwxrwxrwx 1 root root 11 2018-08-11 07:34 libc.so.6 -> libc-2.6.so

该条目的第一个字母是**l**，而且看起来像有2个文件名。这种特殊的文件叫作符号链接(**软链接**)。

假设安装了2.6版本的文件foo，其文件名为foo-2.6，然后创建一个指向foo-2.6的符号链接foo。**这意味着程序打开foo文件时，实际上打开的是foo-2.6文件。**

12.通配符

Shell要用到大量的文件名，因此有一种特殊字符，帮助快速指定一组文件名。这组特殊字符叫作**通配符**。使用通配符的过程也叫**通配符匹配**。

**\*** 匹配任意多个字符

**?** 匹配任意单个字符

**[characters]**匹配属于字符集合characters中的任意单个字符

**[!characters]**匹配不属于字符集合characters中的任意单个字符

**[[:class:]]** 匹配属于字符类class中的任意单个字符,如下：

**[:almum:]** 匹配任意单个字母数字字符

**[:alpha:]**匹配任意单个字母

**[:digit:]**匹配任意单个数字

**[:lower:]**匹配任意单个小写字母

**[:upper:]**匹配任意单个大写字母

通配符示例：

\* 所有文件，不能匹配以.开头的文件(隐藏文件)。

g\* 以g开头的任意文件

b\*.txt 以b开头，扩展名为.txt的所有文件

Data???以Data开头并紧接3个字符的文件

[abc]\* 以abc中的任意字符开头的文件

BACKUP.[0-9][0-9][0-9] 以BACKUP开头并紧接3个数字的文件

[[:upper:]]\* 以单个大写字母开头的文件

[![:digit:]]\* 不以数字开头的文件

\*[[:lower:]123]以小写字母或1、2、3中任意数字结尾的文件

**mkdir** 创建目录，可以**同时创建多个**目录：

mkdir dir1 dir2 dir3

cp 复制文件和目录

cp item1 item2 #复制单个文件item1到item2

cp item... directory #**将多个文件或目录**复制到目录directory，该目录必须事先存在

cp命令的选项：

-a, --archive 复制文件和目录及其包括所有权与权限在内的所有属性。通常副本采用执行复制操作的**用户的默认属性**。

-i, --interactive 在覆盖原有文件之前，提示用户确认。否则不会警告

-r, --recursive 递归复制目录及其内容。复制目录时需要(或-a选项)

-u, --update 只复制目标目录中不存在或比目标目录中现有文件更新的文件，能够跳过不必要的复制

-v, --verbose 在复制时显示相关信息

**mv 移动和重命名文件**，会移动该目录下的所有文件

**rm 删除文件和目录**，可删除多个

-i, --interactive 删除已有文件前提示用户确认

-r, --recursive **递归删除目录**

-f,--force 忽略不存在的文件，不提示。

-v, --verbose 在执行删除时显示相关操作

1. ln——创建硬链接和符号链接

ln命令可用于创建硬链接或符号链接，该命令有2种不同的用法。

**创建硬链接： ln file link**

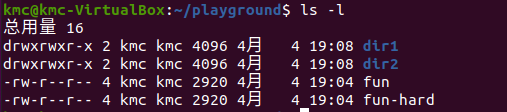
**创建符号链接： ln -s item link** (其中，item可以是文件或目录)

在默认情况下，每个文件只有一个**inode号**，只有一个硬链接(原文件名)。创建硬链接时就为文件额外创建了一个文件名。硬链接不能引用目录，不能用在不同的文件系统中。每个inode节点**只在一个文件系统内有唯一的标识**，不同文件系统之间的inode节点是独立的。因此，如果需要创建硬链接，必须在同一个文件系统中进行。删除硬链接时，**消失的只是链接本身而非文件内容**，所有硬链接都消失时文件才会被删除。

符号链接：

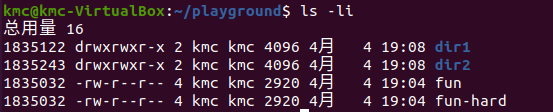
创建一种特殊类型的文件，内含一段指向被引用文件或目录的文本指针，与快捷方式类似。删除符号链接时只会删除链接而非文件。文件不存在而符号链接存在时，链接为**无效链接**，ls命令中会用**不同的颜色(红色)**来显示无效链接。

在ls -l显示中，



文件权限后面的数字，如fun显示4表示该文件当前的硬链接数量，这里的fun-hard也是同一个文件的硬链接。创建硬链接，实际上创建的是名称部分，全都指向相同的数据部分，系统为i节点(inode)分配了一系列磁盘块，然后将i节点与名称部分关联在一起。每个文件都是一个i节点。

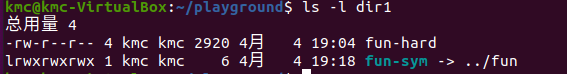
ls -i可以显示文件指向的inode



14.创建**符号链接**

ln -s item link 其中，item可以用相对路径和绝对路径来表示。相对路径是相对于link的路径。如：

ln -s ../fun dir1/fun-sym #在dir1目录中创建符号链接fun-sym，该符号链接与父目录的fun相连。



这里符号链接文件的大小为6，表示字符串../fun的长度而非所指向的文件大小。

符号链接可以引用目录

rm删除链接不会影响目标文件

15.命令相关

**type** 命令 #显示命令类型

type ls #显示ls的类型

**which** 显示可执行文件的位置，只适用于可执行文件。

which ls #显示ls的位置即/bin/ls

对于"cd"命令而言，它并不是一个可执行的程序或者脚本，而是一个shell内置命令。

**help** 命令 #获取shell内建命令的帮助信息

help cd

命令 --help #显示用法信息

mkdir --help

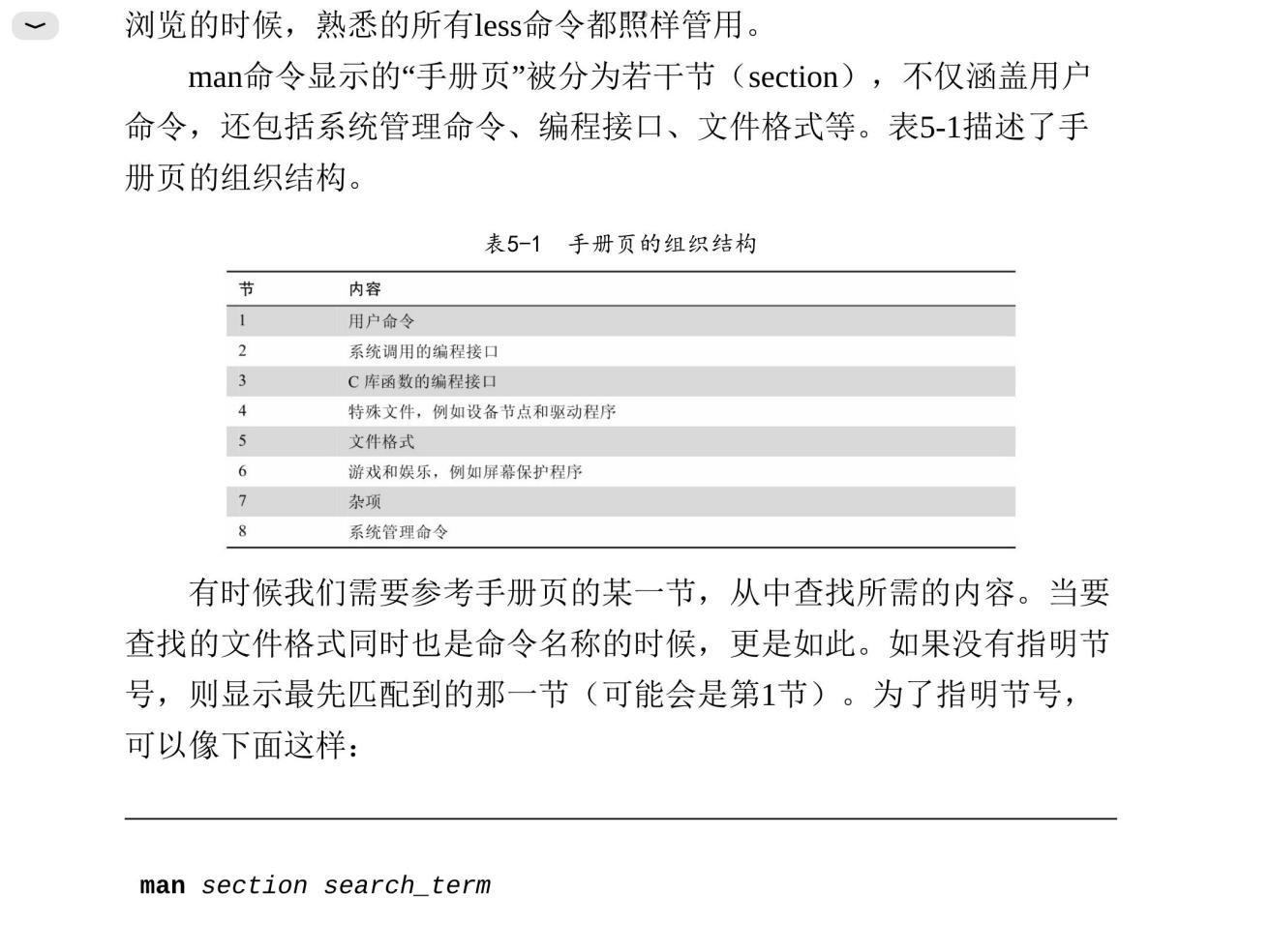
**man** #显示命令的手册页

大多数用于命令行的程序会提供一份叫作手册或手册页的正式文档，man可以浏览这种文档

man *program* #program是待浏览的手册页对应的命令名称

man ls #显示ls的手册页

在大多数Linux系统中，man命令使用less命令显示手册页，less命令一样能用。man显示的手册页能被分为**若干节，每节有不同内容**。



可以阅读某一节的内容：

man 节数 阅读的命令

如： man 5 passwd #显示passwd手册页中的文件格式

**apropos** 显示合适的命令清单，可根据关键字在手册页列表搜索可能的匹配项。

apropos partition

**whatis** 显示手册页的简述

会显示匹配指定关键字的手册页名称和单行描述

**info**

info是GNU项目为其程序所写的手册页的替代品。info页使用info阅读器显示。

info程序读取info文件，该文件按照树形结构组织成各个单独的节点，每个节点包含一个主题。info文件包含的超链接可以在节点之间跳转。

**文档文件**

很多软件包有自己的文档文件，被存放在/usr/share/doc目录中，可用**less**查看。

**.gz扩展名文件**表示它们经过gzip压缩，gzip软件包中的特殊命令**zless**可以显示.gz文档文件的内容

16.使用**alias**创建自己的命令

使用**;**作为分隔符，可以在命令行中**一次性输入多个命令**:

command1;command2;command3...

如： cd /usr; ls; cd -

使用alias将上述命令序列变成新命令，需要先为新命令创建别名(foo)，创建别名之间最好**检查一下别名是否被占用**： type foo;

alias命令的写法： **alias 名称=’命令内容’** #名称与等号之间**不能**有空格

如： alias foo=’cd /usr; ls; cd -’

之后自创的命令可以应用在任何地方，可用type查看命令内容

**unalias**可以删除别名：

unalias foo;

使用已有的命令名称作为别名可以为已有命令添加选项

可以用**不加参数的alias**获得系统中定义的所有别名。当Shell会话结束时，别名会随之消失

17.重定向

ls命令一般情况下将运行结果和状态/错误信息分别发送到stdout(标准输出)、stdeer(标准错误)的特殊文件。默认情况下标准输出和标准错误**与显示器屏幕相关联**，**并不会保存为磁盘文件**。

许多程序从**stdin(标准输入)**中获取输入，标准输入默认与键盘相关。

重定向可以修改输入输出来源。

18.标准输出重定向

将标准输出重定向到其它文件，可以用重定向操作符>， >后加文件名

**ls -l /usr/bin > ls-output.txt** #将/usr/bin的长列表输出到ls-output.txt文件中，会**重写**该文件

将重定向的标准输出**追加**到文件尾部，而非重写，可以用**>>**实现：

19.标准错误重定向

标准错误重定向缺少专门的重定向操作符，需要引用其文件描述符。**标准输入、标准输出、标准错误的文件描述符分别为0、1、2。**

重定向标准错误：

ls -l /bin/usr **2>** ls-error.txt #在>前加上文件描述符2，表示标准错误重定向

20.将标准输出和标准错误重定向到同一个文件

法一：ls -l /bin/usr **>** ls-output.txt **2>&1**

在这里执行了两次重定向，先将标准输出重定向到ls-output.txt，再用2>&1将文件描述符2(标准错误)重定向到文件描述符1(标准输出).**这里的&用来表明1是文件描述符，表示标准输出，不加&会表示文件1**。

法二：ls -l /bin/usr **&>** ls-output.txt

使用&>同时将标准输出和标准错误重定向。追加重定向时使用&>>

21.丢弃不需要的输出结果

将输出结果重定向到名为**/dev/null**的特殊文件，该文件是系统文件，被称作**位桶**。

22.**标准输入**重定向

cat 文件名1 文件名2

读取一个或多个文件并将其复制到标准输出 (即**拼接**多个文件内容)，可以用于重定向输出：

cat movie.mepg.0\* > movie.mpeg #通配符会按顺序扩展，这条命令将所有以0\*作为后缀的文件**拼接成1个文件**

如果只输入cat，**没有参数**则cat会从**标准输入读取。**输入完成后按下**Ctrl+D**表示**EOF**，告诉cat到达标准输入的文件末尾。

cat > lazy\_dog.txt #重定向输出到lazy\_dog.txt

重定向标准输入：

cat < lazy\_dog.txt #从读取文件作为输入

23.管道

**Shell的管道**利用了命令能够从标准输入读取数据并发送到标准输出的能力。管道**操作符|**可以将一个命令的**标准输出传给**另一个命令的**标准输入**。

command1 | command2

将command1的输出发送给**less**，可以用less打开该输出

管道可以放入**多个**命令：

ls /bin /usr/bin | sort | less #输出列表，之后将输出的列表先排序，再用less打开该排序后的输出

uniq 删除**连续的**重复行，不连续的无影响

通常与sort命令配合使用，可以从标准输入或单个文件名参数中获取有序的数据列表，默认**删除所有重复行**。

ls /bin /usr/bin | sort | uniq | less

使用uniq的-d选项可以查看重复行

**wc** 统计文件中**换行符、单词、字节**的数量

**wc 文件名** #输出**3个数字**，分别代表文件中的行数、单词数、字节数。

wc的**-l**选项限制**只输出文件行数**。

grep 用于在文件中查找文本模式

**grep 模式 文件** 输出与模式匹配的行，模式是**正则表达式**

如 ls |grep kmc

grep的选项有：

**-i** 在搜索时忽略字母的大小写(默认区分大小写)

**-v** 只输出不匹配指定模式的行

**head/tail**

只得到命令**输出结果的前几行或后几行**，用head和tail命令，默认为前10/后10行。行数可以通过**-n**选项来调整。

**head/tail -n 5 文件名** #输出文件内容的前5或后5行

tail -f 文件名 #-f选项可以**实时**查看文件，即输出与文件内容**同步更新**。按**ctrl-c**停止查看

tee 文件名 #读取标准输入并将输出结果**写入标准输出和文件中**，即**读取一个输入，输出到两个位置**。

24.Shell

再输入命令按下enter键之后，执行命令前Bash会执行一些**文本替换操作**，如\*有多个含义，这一处理过程被称为**扩展**。

**echo 文本** #将输入的文本参数**输出到标准输出**

echo D\* #D\*会自动通配到所有以D开头的文件名

**波浪线~**扩展：

默认~是指同名用户的主目录

**~用户名** #如果系统中存在该用户，则扩展为该用户的主目录，如~foo 会扩展为/home/foo

算术扩展

**$((表达式))** 表示算术表达式

可以将Shell当作计算器使用： echo $((2+2)) #输出4

花括号扩展

可以按照**花括号中的模式**创建多个文本字符串。在花括号前后可以添加**前导和后继**。

echo Front-{A,B,C}-Back

输出： Front-A-Back Front-B-Back Front-C-Back

花括号本身可以是逗号分隔的字符串列表，也可以是整数区间或单个字符，不能包含未经引用的空白字符。

整数区间写法： **{1..5}** #[1, 5]区间,用..连接起始和终止范围。

整数也可以用0填充： echo{01..15} #生成01, 02, .., 15

**{Z..A}**  #字母**降序**排列

花括号也可以降序：echo a{A{1, 2}, B{3, 4}}b #生成aA1b,...aB4b

利用花括号扩展：mkdir {2007..2009}-{01..12} #建立多个目录

25.**参数**扩展

在Shell脚本中要比直接在命令行中更实用，用于查看变量。

echo **$**USER #查看**变量**USER中的内容

printenv | less #**printenv**查看可用的变量列表

26.命令**替换**

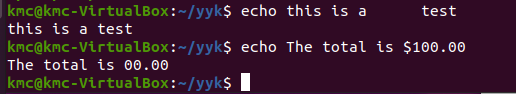
允许使用命令的输出作为**扩展输入**，无需得到具体输出结果

ls -l $(which cp) #无需知道cp的路径，就能得知cp的详细信息。$用于获取变量的值

在旧版Shell中，用’’代替$()

27.引用

用于**控制扩展**



Shell通过单词分割，**删除**echo命令参数中**多余的空白字符**；同时如果**$1**未被定义，则扩展为**空字符串**

(1 双引号

放入双引号中的文本会被视为普通字符，但是参数扩展、算术扩展、命令参数仍可用。

如果有一个名为two words.txt的文件， ls -l two word.txt会将其视为两个独立的参数。使用””可以阻止单词分割：

ls -l “two word.txt” #可以访问到该文件

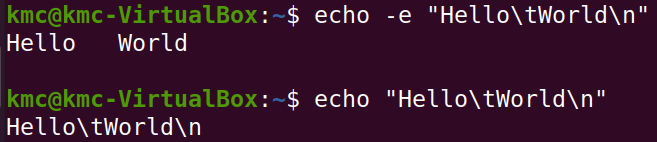
(2 单引号

想要禁止所有扩展，可以使用单引号。这不会计算算术表达式等结果。

28.转义字符

只引用单个字符，可以在该字符前加上一个反斜线\，与C语言类似。

echo的-e选项能够解释转义序列。



sleep 秒数 #能够暂停一段时间(以秒为单位)

29.光标移动

Ctrl-A 移动光标到行首

Ctrl-E 移动光标到行尾

Ctrl-F 将光标向右移动一个字符，相当于右方向键

Ctrl-B 将光标向左移动一个字符，相当于左方向键

Alt-F 将光标向前移动一个单词

Alt-B 将光标向后移动一个单词

Ctrl-L 清除屏幕，相当于clear命令

Ctrl-D 删除光标处的字符

Ctrl-T 将光标处的字符与其之前的字符对调

Alt-T 将光标处的单词与其之前的单词对调

Alt-L 将从光标处到单词结尾的字符变为小写

Alt-U 将从光标处到单词结尾的字符变为大写

剪切和粘贴，被剪切的文本保存在名为kill-ring的缓冲区(内存中的一块临时存储区域)

Ctrl-K 剪切从光标处到行尾的文本

Ctrl-U 剪切从光标处到行首的文本

Ctrl-Y 粘贴文本

查看命令的历史记录：

history | less

调用历史记录中的命令： !编号 #调用相应编号的历史记录

可以用Ctrl-R后搜索输入的历史命令，从当前命令行向上搜索。

Ctrl-O 执行历史记录列表中的当前命令并移动到下一条命令。

script file #可以记录整个Shell会话并将其保存在file文件中。

30.权限

文件或目录如果属于某个用户，**该用户(属主)**拥有**访问权**。用户可以属于由一个或多个用户组成的组**(属组)**，属组被文件和目录的**属主**授予访问权。属主还可以为**所有用户**授权。

查看用户身份信息： id命令

创建用户时，会为用户分配一个**用户ID(UID)**数字，uid又被映射为用户名。用户会得到一个**属组ID(gid)**，还可以归属于其他属组。Ubuntu中UID从1000开始。

属主定义在**/etc/passwd**中，属组定义在/etc/group中。在创建用户和属组时，这些文件随/etc/shadow一起改动，**/etc/shadow保存了用户的密码**，/etc/passwd定义了用户名、uid、gid、用户真实姓名、主目录、登录Shell。

文件权限(分为对文件和对目录来说)：



**目录的x权限**允许用户**访问该目录中的文件**。如果用户没有x权限，即使该用户具有文件的r权限，也无法访问该目录中的文件。

1. chomd 修改文件模式

只有**文件属主或超级用户**才能用chmod修改权限，有两种方式：**八进制表示法和符号表示法**。

(1 **八进制表示法**

**每个8进制位**表示**3个二进制**数位。因此文件模式---表示0，rwx表示7.

chomd 600 foo.txt #将属主的权限设置为可读取和写入，同时**去除**属组和其他用户的权限。600分别表示ugo的每一位权限。

(2 符号表示法

符号由三个部分组成：**改动影响对象、执行的操作、设置的权限。**

改动影响对象可通过字符**u、g、o、a**来指定。u表示属主，g表示属组，o表示其他用户，a表示all，包括属主、属组、其他用户。如果没有指定字符，则使用all。

操作符**+表示添加权限，-表示去除权限，=表示只赋予指定权限。**同时**r、w、x**分别表示读写执行权限。如u+x表示为属主添加执行权限。+x表示为所有用户添加执行权限。u+x,go=rx 设置多组权限时，彼此之间用逗号分隔。

1. **umask**设置**默认权限**

umask可设置文件创建时的默认权限。

**不带参数**的umask可以查看当前的**掩码值**，它是掩码的8进制表示。第一个8进制位不考虑。**一个掩码位**表示**3个二进制位**，相应二进制位位置为**1**的权限会被**去除**。

默认情况下，新建文件的权限是666，目录的权限是777。

umask不能控制执行权限。

umask 0000 #设置掩码值为0000

33.改变用户身份

su 允许扮演其他用户，以该用户的ID启动新的Shell会话或执行单个命令。

sudo 允许超级用户设置名为/etc/sudoer的配置文件、执行特定命令。

34.

35.

36.

37.

38.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.

60.