**IMG_256Linux**

**—————————————————————————————————**

**1.Linux文件与目录结构**

**1.1 Linux文件**

Linux系统中一切皆文件，在根目录之下的既可以是目录，也可以是文件

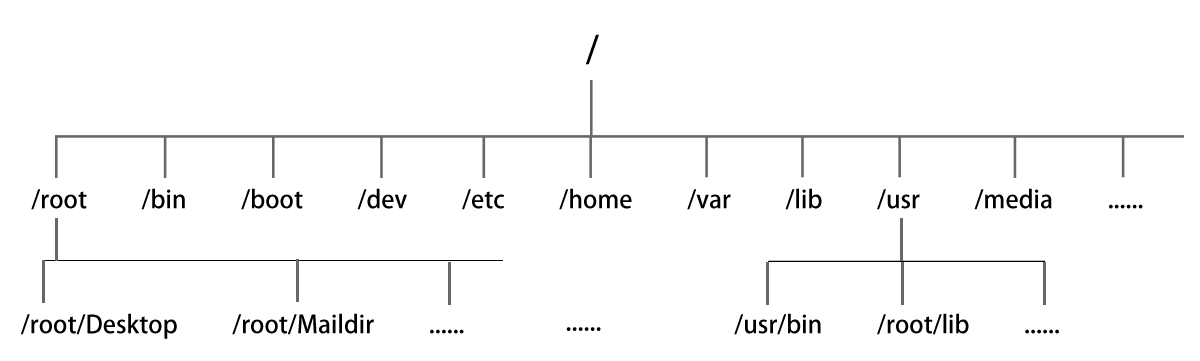
**1.2 Linux目录结构**

文件系统的最顶层是由根目录开始的，系统使用 / 来表示根目录。而每一个目录中又可以包含子目录文件。如此反复就可以构成一个庞大的文件系统。

. ：代表当前的目录，也可以使用 ./ 来表示；

.. ：代表上一层目录，也可以 ../ 来代表。

如果一个目录或文件名以一个点 . 开始，表示这个目录或文件是一个隐藏目录或文件(如：.bashrc)。即以默认方式查找时，不显示该目录或文件。

如左图

/bin Binaries (二进制文件) 的缩写，这个目录存放着最经常使用的命令

/sbin s是Super User的意思，这里存放的是系统管理员使用的系统管理程序

/home 存放普通用户的主目录，每个用户一般有一个以该用户账号命名的目录

/root 系统管理员的目录

/lib Library(库) 的缩写,系统开机所需要最基本的动态连接共享库，作用类似Windows的DLL文件。几乎所有应用程序都需要这些共享库

/lost+found 该目录一般为空，系统非法关机后存放文件

/etc Etcetera(等等) 的缩写,所有系统管理所需要的配置文件和子目录

/usr unix shared resources(共享资源) 的缩写，存放很多用户的应用程序和文件，类似于windows的program files目录

/boot 系统启动的核心文件，包括一些连接文件和镜像文件

/proc Processes(进程) 的缩写，/proc是一种伪文件系统（也即虚拟文件系统），存储的是当前内核运行状态的一系列特殊文件，该目录是在内存中的虚拟目录，是系统内存的映射，可以直接访问该目录来获取系统信息

/srv Service缩写，存放一些服务启动后需要提取的数据

/sys 文件系统

/tmp 存放临时文件

/dev Device(设备) 的缩写,类似于win的设备管理器，把所有硬件用文件形式存储

/media 系统识别的设备会被挂载在这个目录

/mnt 让用户临时挂载别的文件系统

/opt optional(可选) 的缩写,主机额外安装软件的目录

/var variable(变量) 的缩写，存放不断扩充的东西，用户习惯将经常被修改的目录存放在这个目录下，包括各种日志文件

/usr/src： 内核源代码默认的放置目录。

/run： 是一个临时文件系统，存储系统启动以来的信息。当系统重启时，这个目录下的文件应该被删掉或清除。如果你的系统上有 /var/run 目录，应该让它指向 run。

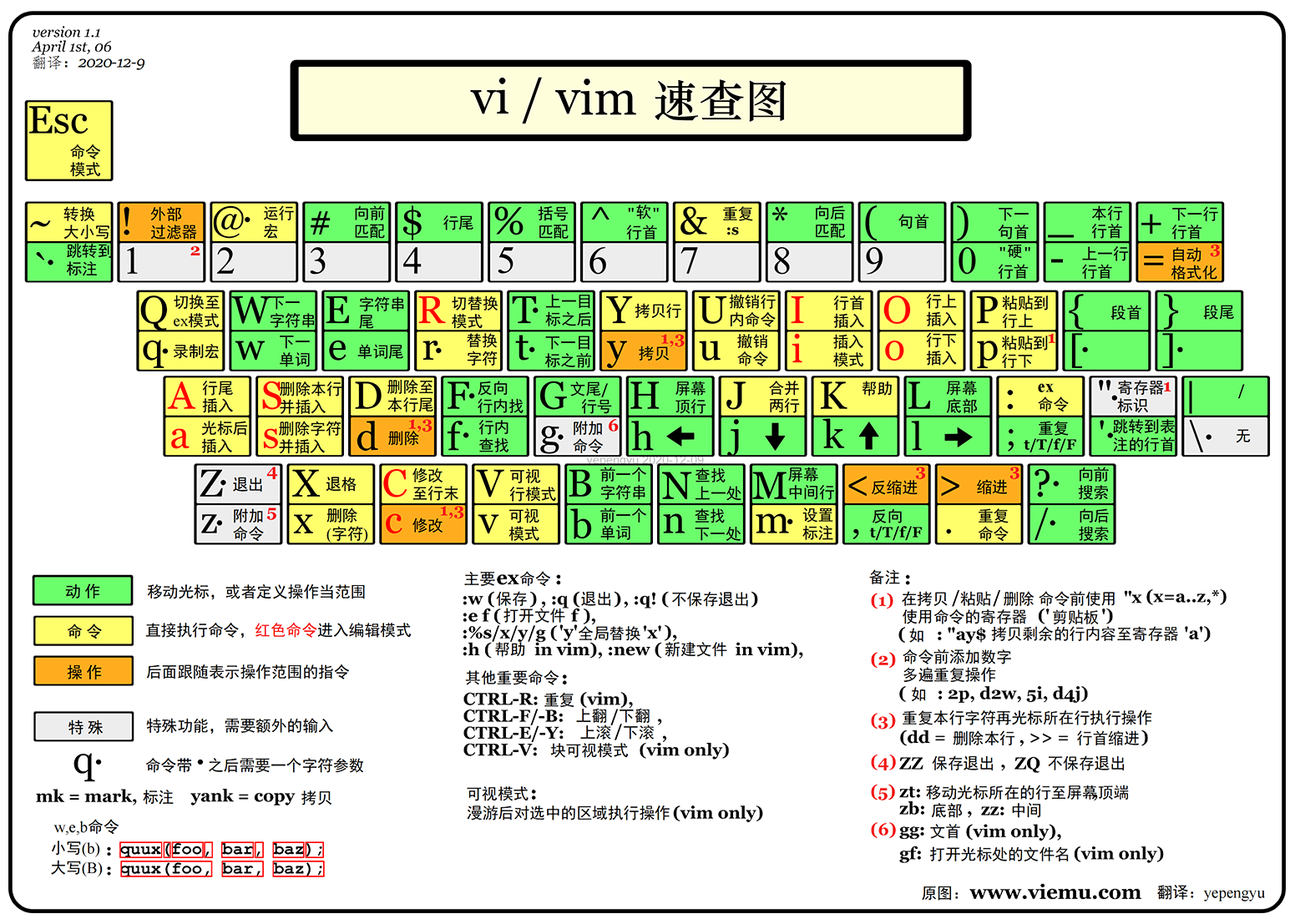
1. VI//VIM编辑器

2.1解释

VI是Unix操作系统和类Unix操作系统中最通用的文本编辑器。

VIM编辑器是从VI发展出来的一个性能更强大的文本编辑器。可以主动的以字体颜色辨别语法的正确性，方便程序设计。

VIM与VI编辑器完全兼容。

基本上 vi/vim 共分为三种模式，命令模式（Command Mode）、输入模式（Insert Mode）和命令行模式（Command-Line Mode）。

2.2 Vim的使用

输入 vim 文件路径 来使用vim编辑器打开文件。

**2.2.1 一般模式**

进入vim的初始模式

|  |  |
| --- | --- |
| **光标移动** | **功能描述** |
| h 或 向左箭头键 | 光标向左 |
| j 或 向下箭头键 | 光标向下 |
| k 或 向上箭头键 | 光标向上 |
| l 或 向右箭头键 | 光标向右 |
| 在键盘上 hjkl 是排列在一起的  如果想要进行多次移动的话，输入：（数字）（方向键）  例：如向下移动 30 行，可以使用 "30j" 或 "30↓" 的组合按键 | |
| b | 回退到光标上一个位置 |
| w | 移动到下一个词头 |
| e | 移动到当前词尾 |
| (数字)gg | (n代表页）移动到文件头行 |
| Ctrl + f | 屏幕『向下』移动一页，相当于 [Page Down]按键 (常用) |
| Ctrl + b | 屏幕『向上』移动一页，相当于 [Page Up] 按键 (常用) |
| Ctrl + d | 屏幕『向下』移动半页 |
| Ctrl + u | 屏幕『向上』移动半页 |
| + | 光标移动到非空格符的下一行 |
| - | 光标移动到非空格符的上一行 |
| ^ 或 数字键0 | 这是数字『 0 』：移动到这一行的最前面字符处 (常用) |
| $ | 移动到这一行的最后面字符处(常用) |
| H | 光标移动到这个屏幕的最上方那一行的第一个字符 |
| M | 光标移动到这个屏幕的中央那一行的第一个字符 |
| L | 光标移动到这个屏幕的最下方那一行的第一个字符 |
| (n)G | n 为数字。移动到这个档案的第 n 行，无n则移动到这个档案的最后一行。 |
| (n)gg | 移动到这个第n页的第一行 (常用) |
| (n)<Enter> | 光标向下移动 n 行(常用) |
| (n)<space> | 光标会向右移动这一行的 n 个字符。 |

|  |  |
| --- | --- |
| **功能语法** | **功能描述** |
| --复制操作，关键字（y），输入完之后按<enter> | |
| y(n)h 或 y(n)l | 复制当前光标开始向 h:左 或 l:右 的n个字符 |
| (n)yy | 复制当前光标开始向下的n行（不写n则复制光标当前行1行) |
| y(n)y 或 (n1),(n2)y | 复制一段(从第几行到第几行) 可配合G，^，$等关键字使用 |
| y$ | 复制光标所在字符到行末内容 |
| y^ 或 y0 | 复制光标所在字符到行首内容 |
| y(n)w | 无n复制光标所在字符到当前单词末尾的内容（先到当前词开头）  有n向后复制n个单词 |
| --剪切操作，关键字（x）|| 其余操作方法与复制操作同理 | |
| (n)x | 无n删除字符(相当于del) 有n删除n个字符 |
| (n)Shift+x | 无n剪切并回到光标上一个（相当于Backspace）有n删除n个字符 |
| --删除操作，关键字（d）|| 操作方法与复制操作同理 | |
| (n)p | 光标后的行粘贴,有n则重复粘贴n次 |
| (n)P | 光标前 |
| u | 撤销上一步 |
| ctrl+r | 反撤销（重做上一个动作） |
| Ctrl+Z 或 Ctrl+C | 退出vim |

|  |  |
| --- | --- |
| **搜索替换** | **功能描述** |
| /(查找内容) | 从光标所在行向下查找 |
| ?(查找内容) | 从光标所在行向上查找 |
| n | 下一个 |
| N | 上一个 |
| ：[addr](n1),(n2)s/(字)/(字)/[option] | 表示将前面的(字)替换为后面的(字)。  检索范围[addr]："n1,n2":表示从第n1行到n2行，"%":表示整个文件,同"1,$"，".,$":表示从当前行到文件尾，省略时表示当前行  s:表示替换操作,其为substitute的缩写  操作类型[option] : g全局替换，c进行确认，p逐行显示(Ctrl + L恢复屏幕)，i不分大小写，省略时仅对每行第一个匹配串进行替换 |

2.2.2 编辑模式

编辑模式中跟普通操作文本的操作方法一样

|  |  |
| --- | --- |
| **从一般模式进入编辑模式** | **功能描述** |
| i , I | 进入输入模式(Insert mode)：  i 为『从目前光标所在处输入』， I 为『在目前所在行的第一个非空格符处开始输入』。 (常用) |
| a , A | 进入输入模式(Insert mode)：  a 为『从目前光标所在的下一个字符处开始输入』， A 为『从光标所在行的最后一个字符处开始输入』。(常用) |
| o , O | 进入输入模式(Insert mode)：  这是英文字母 o 的大小写。o 为在目前光标所在的下一行处输入新的一行； O 为在目前光标所在的上一行处输入新的一行！(常用) |
| r , R | 进入取代模式(Replace mode)：  r 只会取代光标所在的那一个字符一次；R会一直取代光标所在的文字，直到按下 ESC 为止；(常用) |
| [ Esc ] | 退出编辑模式。 |

2.2.3 指令行模式

输入 “:” 进入指令行模式

|  |  |
| --- | --- |
| 功能 | **功能描述** |
| :w | 将编辑的数据写入硬盘档案中 |
| :w! | 若文件属性为『只读』时，强制写入该档案。  能不能写入与用户对该档案的档案权限有关。 |
| :q | 离开 vi |
| :q! | 修改过档案后不想储存，使用 ! 为强制离开不储存档案。 |
| :wq | 储存后离开，若为 :wq! 则为强制储存后离开 (常用) |
| :w [filename] | 将编辑的数据储存成另一个档案（类似另存新档） |
| :r [filename] | 在编辑的数据中，读入另一个档案的数据。亦即将 『filename』 这个档案内容加到游标所在行后面 |
| 下面设置的参数都只是临时生效，一旦关闭文件再打开，又需要重新输入。如果想要永久生效，需要手动建立 vim 的配置文件 ==“~/vimrc”== ，把你需要的参数写入配置文件就永久生效了。 | |
| :set （no)nu | (不)显示行数字 |
| syntax on(off) | 是否依据语法显示相关颜色帮助 |
| set (no)hlsearch | 设置是否将查找的字符串高亮显示 |
| set (no)showmode | 设置是否在左下角显示如“-INSERT-” 之类的状态栏 |
| set (no)list | 设置是否显示隐藏字符（tab 键用 "^I" 表示，回车符用 "$" 键表示） |
| Windows 下回车符在 Linux 中是用 “^M$” 显示的，而不是 “$” 。这样会导致 Windows 下编辑的程序脚本（如 shell 脚本），无法在 Linux 中执行。这时可以通过命令 ==「dos2unix」==，把 windows 格式转换为 Linux 格式。同样可以使用 ==「unix2dos」== ，把 Linux 格式转换为 Windows 格式。这两个命令默认没有安装，需要手动安装才能使用。 | |

1. **系统管理**

**3.1进程与服务**

计算机中，一个正在执行的程序或命令，被叫做“进程”（process）。

启动之后一只存在、常驻内存的进程，一般被称作“服务”（service）。

**3.2 service服务管理**

**3.2.1 基本语法**

service 服务名 start | stop |· restart | status (Centos6)

systemctl start | stop | restart | status 服务名 (Centos7)

**3.2.2 操作**

案例:(centos7改成systemctl)

（1）查看网络服务的状态service network status

（2）停止网络服务service network stop

（3）启动网络服务service network start

（4）重启网络服务service network restart

查看服务的方法：/usr/lib/systemd/system

**3.2.3** 设置后台服务的自启动配置

//centOs7

**systemctl list-unit-files （功能描述：查看服务开机启动状态）**

**systemctl disable service\_name （功能描述：关掉指定服务的自动启动）**

**systemctl enable service\_name （功能描述：开启指定服务的自动启动）**

//centOs6

**chkconfig （功能描述：查看所有服务器自启配置）**

**chkconfig 服务名 off （功能描述：关掉指定服务的自动启动）**

**chkconfig 服务名 on （功能描述：开启指定服务的自动启动）**

**chkconfig 服务名 --list （功能描述：查看服务开机启动状态**

**3.2.4 系统运行级别**

开机→BIOS→/boot→init进程→运行级别→运行对应服务

Linux系统有7种运行级别(runlevel):(常用的是级别3和5)

·**运行级别0**:系统停机状态，系统默认运行级别不能设为0,否则不能正常启动

·**运行级别1**:单用户工作状态，root权限，用于系统维护，禁止远程登陆

·**运行级别2**:多用户状态(没有NFS),不支持网络

·**运行级别3**:完全的多用户状态(有NFS),登陆后进入控制台命令行模式

·**运行级别4**:系统未使用，保留

·**运行级别5**:X11控制台，登陆后进入图形GUI模式

·**运行级别6**:系统正常关闭并重启，默认运行级别不能设为6,否则不能正常启动

**CentOS7 的运行级别简化为:**

multi-user.target 等价于原运行级别 3（多用户有网，无图形界面）

graphical.target 等价于原运行级别 5（多用户有网，有图形界面）

查看当前运行级别： systemctl get-default

修改当前运行级别：

systemctl set-default TARGET.target （这里 TARGET 取 multi-user 或者 graphical）

**3.3 关机重启（man shutdown 查看关机帮助文档）**

（1）sync （功能描述：将数据由内存同步到硬盘中）

（2）halt （功能描述：停机，关闭系统，但不断电）

（3）poweroff （功能描述：关机，断电）

（3）reboot （功能描述：就是重启，等同于 shutdown -r now）

（4）shutdown [选项] 时间

-H 相当于--halt，停机

-r -r=reboot 重启

now 立刻关机

时间 等待多久后关机（时间单位是分钟）。

例子： shutdown –h now 立马关机

shutdown –h 20:25 系统会在今天20:25关机

shutdown –h +10 十分钟后关机

shutdown –r now 系统立马重启

shutdown –r +10 系统十分钟后重启

shutdown -c 取消关机

1. **帮助命令**

**4.1 获取帮助信息**

**4.1.1 man获取帮助信息**

man [命令或配置文件]

|  |  |
| --- | --- |
| 信息 | 功能 |
| NA M E | 命令的名称和单行描述 |
| SY NO PS IS | 怎样使用命令 |
| D ESC R I PT IO N | 命令功能的深入讨论 |
| EXA M P L ES | 怎样使用命令的例子 |
| S E E A LSO | 相关主题（通常是手册页） |

**4.1.2 help获取内置命令帮助信息**

help 命令（ 功能描述：获得 shell 内置命令的帮助信息）

**4.1.3 tab使用**

[Tab] ## 接在一串指令的第一个字的后面，则为『命令补全』

[Tab] ## 接在一串指令的第二个字以后时，则为『文件补齐』

1. **文件目录类**

**5.1 pwd 显示当前工作目录的绝对路径**

**5.2 ls 列出目录的内容**

|  |  |
| --- | --- |
| 选项 | 功能 |
| -a | 全部的文件，连同隐藏档( 开头为 . 的文件) 一起列出来(常用) |
| -l | 长数据串列出，包含文件的属性与权限等等数据；(常用)等价于“ll ” |

**5.3 cd 切换目录**

|  |  |
| --- | --- |
| 参数 | 功能 |
| cd 绝对路径 | 切换路径 |
| cd 相对路径 | 切换路径 |
| cd ~或者 cd | 回到 自己的家 目录 |
| cd - | 回到上一次所在 目录 |
| cd .. | 回到当前 目录的上一级 目录 |
| cd - P | 跳转到实际物理路径，而非快捷方式路径 |

**5.4 mkdir 创建一个新的目录**

mkdir [选项] 要创建的 目录

mkdir -p xiyou/dssz/meihouwang (创建多层目录）

**5.5 rmdir 删除一个空的目录**

**5.6 touch 创建空文件**

**5.7 cp 复制文件或目录**

**cp [选项] source dest （功能描述：复制source文件到dest）**

-a：相当於 -pdr 的意思，至於 pdr 请参考下列说明；(常用)

-d：若来源档为链接档的属性(link file)，则复制链接档属性而非文件本身；

-f：为强制(force)的意思，若目标文件已经存在且无法开启，则移除后再尝试一次；

-i：若目标档(destination)已经存在时，在覆盖时会先询问动作的进行(常用)

-l：进行硬式链接(hard link)的链接档创建，而非复制文件本身；

-p：连同文件的属性一起复制过去，而非使用默认属性(备份常用)；

-r：递归持续复制，用於目录的复制行为；(常用)

-s：复制成为符号链接档 (symbolic link)，亦即『捷径』文件；

-u：若 destination 比 source 旧才升级 destination ！

**5.8 rm 删除文件或目录**

rm [选项] deleteFile

-f ：就是 force 的意思，忽略不存在的文件，不会出现警告信息；

-i ：互动模式，在删除前会询问使用者是否动作

-r ：递归删除（危险）

**5.9 mv移动文件与目录或重命名**

（1）mv oldNameFile newNameFile （功能描述：重命名）

（2）mv /temp/movefile /targetFolde （功能描述：移动文件）

-f ：就是 force 的意思，忽略不存在的文件，不会出现警告信息；

-i ：互动模式，在删除前会询问使用者是否动作

-r ：递归删除（危险）

**5.10 Linux 链接概念**

Linux 链接分两种，一种被称为硬链接（Hard Link），另一种被称为符号链接（Symbolic Link）。默认情况下，ln 命令产生硬链接。

**硬链接**：硬连接指通过索引节点来进行连接。在 Linux 的文件系统中，保存在磁盘分区中的文件不管是什么类型都给它分配一个编号，称为索引节点号(Inode Index)。在 Linux 中，多个文件名指向同一索引节点是存在的。硬连接的作用是允许一个文件拥有多个有效路径名，这样用户就可以建立硬连接到重要文件，以防止“误删”的功能。也就是说，文件真正删除的条件是与之相关的所有硬连接文件均被删除。

· ln f1 f2 #创建f1的一个硬连接文件f2

****软连接**：**另外一种连接称之为符号连接（Symbolic Link），也叫软连接。软链接文件有类似于 Windows 的快捷方式。

· ln -s f1 f3 #创建f1的一个符号连接文件f3

**5.11 Linux 压缩解压**

**5.11.1 gzip/gunzip压缩**

**gzip[选项]XXX.zip将要压缩的内容 （功能描述：只能压缩文件, 不保留原来的文件）**

**gunzip[选项]XXX.gz （功能描述：解压缩文件）**

**5.11.2 zip/unzip压缩**

**zip[选项]XXX.zip将要压缩的内容 （功能描述：压缩文件和目录(选项-r)的命令）**

**unzip[选项]XXX.zip （功能描述：解压缩文件)(-d 目录 指定存放位置）**

**5.11.3 tar打包**

**tar[选项]XXX.tar.gz将要打包进去的内容**

**-c 产生 .tar 打包文件**

**-v 显示详细信息**

**-f 指定压缩后的文件名**

**-z 打包同时压缩**

**-x 解包.ta r 文件**

**-C 解压到指定 目录**

1. **文件查看**

6.1 cat

一般显示较小的文件

cat [选项] 要查看的文件

-A ：相当於 -vET 的整合选项，可列出一些特殊字符而不是空白而已；

-b ：列出行号，仅针对非空白行做行号显示，空白行不标行号！

-E ：将结尾的断行字节 $ 显示出来；

-n ：列印出行号，连同空白行也会有行号，与 -b 的选项不同；

-T ：将 [tab] 按键以 ^I 显示出来；

-v ：列出一些看不出来的特殊字符

6.2 tac

与cat相反，从文件最后一行开始显示

6.3 more 文件内容分屏查看器

more 指令是一个基于 VI 编辑器的文本过滤器，它以全屏幕的方式按页显示文本文件的内容。

space 翻下一页

Enter 下一行

q 立刻离开

Ctrl+F 向下滚动一屏

Ctrl+B 返回上屏

= 输出当前行号

:f 输出文件名和当前行号

/字串 向下搜索

?字串 向上搜索

6.4 less 分屏显示文件内

less 指令用来分屏查看文件 内容，它的功能与 more 指令类似，但是比 more 指令更加强大，支持各种显示终端。less 指令在显示文件 内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容，对于显示大型文件具有较高的效率。

space 翻下一页

Enter 下一行

q 立刻离开

Ctrl+F 向下滚动一屏

Ctrl+B 返回上屏

= 输出当前行号

:f 输出文件名和当前行号

g 到开头

G 到末尾

/字串 向下搜索(n下 N上)

?字串 向上搜索

6.5 echo 输出内容到控制台

echo [选项] [输出内容]

-e ：支持反斜线控制的字符转换

\\ 输出\本身

\n 换行符

\t 制表符（Tab键)

6.6 head显示文件头部内容

head 用于显示文件的开头部分内容，默认情况下 head 指令显示文件的前 10 行内容。

head 文件 （ 功能描述：查看文件头10行内容）

head -n 5 文件 （ 功能描述：查看文件头5行内容，5可以是任意行数）

6.7 tail 输出文件尾部内容

tail 文件 （ 功能描述：查看文件尾部10行内容）

tail -n 5 文件 （ 功能描述：查看文件尾部5行内容，5可以是任意行数）

tail -f 文件 （ 功能描述：实时追踪该文档的所有更新）

6.8 >输出重定向和>>追加

（1 ）ls -l > 文件 （ 功能描述：列表的内容写入文件 a.txt 中（ 覆盖写））

（2 ）ls -al >> 文件 （ 功能描述：列表的内容追加到文件 aa.txt 的末尾）

（3 ）cat 文件 1 > 文件 2 （ 功能描述：将文件 1 的内容覆盖到文件 2）

（4 ）echo “ 内容” >> 文件

6.9 history 查看已经执行过历史命令

1. 时间日期类

7.1 data

7.1.1 基本语法

date [OPTION]... [+FORMAT]

**可选参数**

-d, --date=STRING：通过字符串显示时间格式，字符串不能是'now'。

-f, --file=DATEFILE：类似于--date; 一次从DATEFILE处理一行。

-I[FMT], --iso-8601[=FMT]：按照 ISO 8601 格式输出时间，FMT 可以为'date'(默认)，'hours'，'minutes'，'seconds'，'ns'。 可用于设置日期和时间的精度，例如：2006-08-14T02:34:56-0600。

-R, --rfc-2822 ： 按照 RFC 5322 格式输出时间和日期，例如: Mon, 14 Aug 2006 02:34:56 -0600。

--rfc-3339=FMT：按照 RFC 3339 格式输出，FMT 可以为'date', 'seconds','ns'中的一个，可用于设置日期和时间的精度， 例如：2006-08-14 02:34:56-06:00。

-r, --reference=FILE：显示文件的上次修改时间。

-s, --set=STRING：根据字符串设置系统时间。

-u, --utc, --universal：显示或设置协调世界时(UTC)。

--help：显示帮助信息。

--version：输出版本信息。

**FORMAT 参数**

在显示方面，使用者可以设定欲显示的格式 ，格式设定为一个加号后接数个标记，其中可用的标记列表如下：

%% 输出字符 %

%a 星期几的缩写 (Sun..Sat)

%A 星期的完整名称(Sunday..Saturday)。

%b 缩写的月份名称（例如，Jan）

%B 完整的月份名称（例如，January）

%c 本地日期和时间（例如，Thu Mar 3 23:05:25 2005）

%C 世纪，和%Y类似，但是省略后两位（例如，20）

%d 日 (01..31)

%D 日期，等价于%m/%d/%y

%e 一月中的一天，格式使用空格填充，等价于%\_d

%F 完整的日期；等价于 %Y-%m-%d

%g ISO 标准计数周的年份的最后两位数字

%G ISO 标准计数周的年份，通常只对%V有用

%h 等价于 %b

%H 小时 (00..23)

%I 小时 (01..12)

%j 一年中的第几天 (001..366)

%k 小时，使用空格填充 ( 0..23); 等价于 %\_H

%l 小时, 使用空格填充 ( 1..12); 等价于 %\_I

%m 月份 (01..12)

%M 分钟 (00..59)

%n 新的一行，换行符

%N 纳秒 (000000000..999999999)

%p 用于表示当地的AM或PM，如果未知则为空白

%P 类似 %p, 但是是小写的

%r 本地的 12 小时制时间(例如 11:11:04 PM)

%R 24 小时制 的小时与分钟; 等价于 %H:%M

%s 自 1970-01-01 00:00:00 UTC 到现在的秒数

%S 秒 (00..60)

%t 插入水平制表符 tab

%T 时间; 等价于 %H:%M:%S

%u 一周中的一天 (1..7); 1 表示星期一

%U 一年中的第几周，周日作为一周的起始 (00..53)

%V ISO 标准计数周，该方法将周一作为一周的起始 (01..53)

%w 一周中的一天（0..6），0代表星期天

%W 一年中的第几周，周一作为一周的起始（00..53）

%x 本地的日期格式（例如，12/31/99）

%X 本地的日期格式（例如，23:13:48）

%y 年份后两位数字 (00..99)

%Y 年

%z +hhmm 格式的数值化时区格式（例如，-0400）

%:z +hh:mm 格式的数值化时区格式（例如，-04:00）

%::z +hh:mm:ss格式的数值化时区格式（例如，-04:00:00）

%:::z 数值化时区格式，相比上一个格式增加':'以显示必要的精度（例如，-04，+05:30）

%Z 时区缩写 （如 EDT）

7.1.2 显示当前时间

date "+%Y-%m-%d %H :%M:%S" （ 功能描述：显示年月 日时分秒）

7.1.3 显示非当前时间

date -d ' 1 days ago' （ 功能描述：显示前一天时间）

date -d '- 1 days ago' （ 功能描述：显示明天时间）

7.1.4 设置系统时间

date -s 字符串时间

例：[root@admin ~]# date -s "2017-06-19 20:52:18"

7.2 查看日历

cal [选项] （功能描述：不加选项，显示本月 日历）

选项可填年月

1. 用户/用户组管理

8.1用户管理

8.1.1添加用户

useradd 选项 用户名

useradd -g 组名 用户名 （功能描述：添加新用户到某个组）

选项:

-c comment 指定一段注释性描述。

-d 目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录。

-g 用户组 指定用户所属的用户组。

-G 用户组，用户组 指定用户所属的附加组。

-s Shell文件 指定用户的登录Shell。

-u 用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号。

8.1.2 usermod 修改用户

usermod 选项 用户名

usermod -g 用户组 用户名

常用的选项包括-c, -d, -m, -g, -G, -s, -u以及-o等，这些选项的意义与useradd命令中的选项一样，可以为用户指定新的资源值。另外，有些系统可以使用选项：(-l 新用户名)指定一个新的账号，即将原来的用户名改为新的用户名。

8.1.3 userdel 删除用户

userdel 用户名 （ 功能描述：删除用户但保存用户主 目录）

userdel -r 用户名 （ 功能描述：用户和用户主 目录，都删除）

8.1.4设置用户密码

passwd 用户名 （ 功能描述：设置用户密码）

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

8.1.5 id 查看用户是否存在

8.1.6 查看已创建用户

cat /etc/passwd

8.1.7 su切换用户

su 用户名称 （ 功能描述：切换用户，只能获得用户的执行权限，不能获得环境变量）

su - 用户名称 （ 功能描述：切换到用户并获得该用户的环境变量及执行权限）

8.1.8 who 查看登录用户信息

（1 ）whoami （ 功能描述：显示 自身用户名称）

（2 ）who am i （ 功能描述：显示登录用户的用户名以及登陆时间）

8.1.9 sudo 设置普通用户具有 root 权限

若想命令提前生效，要修改 /etc/sudoers 文件，找到下面一行(91 行) ，在 root 下面添加一行。

8.2用户组管理

8.2.1 groupadd 新增组

8.2.2 groupdel 删除组

8.2.3 groupmod 修改组

groupmod -n 新组名 老组名

8.2.4 查看已创建组

cat /ete/group

8.3文件权限

Linux系统是一种典型的多用户系统，不同的用户处于不同的地位，拥有不同的权限。为了保护系统的安全性，Linux系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。在Linux中我们可以使用ll或者ls-l命令来显示一个文件的属性以及文件所属的用户和组。

从左到右的10个字符表示，如图7-1所示,如果没有权限，就会出现减号[-]而已。



0位表示类型：

-代表文件 d代表目录 l代表连接文档

[r]代表可读(read) [w]代表可写(write) [x]代表执行(execute)

8.3.1 chmod修改权限

chmod[{ugoa}{+-=}{rwx}]文件或目录

chmod[mode=421][文件或目录]

R=4 w=2 x=1 rwx=4+2+1=7

8.3.2 chown改变所有者

chown[选项][最终用户][文件或目录]

8.3.3 chgrp更改文件属组

chgrp [-R] 属组名 文件名

1. 搜索查找

9.1 find查找文件或目录

find [搜索范围] [选项]

常用：-name pattern：按文件名查找，支持使用通配符 \* 和 ?。

-type type：按文件类型查找，可以是 f（普通文件）、d（目录）、l（符号链接）等。

-size [+-]size[cwbkMG]：按文件大小查找，支持使用 + 或 - 表示大于或小于指定大小，单位可以是 c（字节）、w（字数）、b（块数）、k（KB）、M（MB）或 G（GB）。

-mtime days：按修改时间查找，支持使用 + 或 - 表示在指定天数前或后，days 是一个整数表示天数。

-user username：按文件所有者查找。

-group groupname：按文件所属组查找。

find 命令中用于时间的参数如下：

-amin n：查找在 n 分钟内被访问过的文件。

-atime n：查找在 n\*24 小时内被访问过的文件。

-cmin n：查找在 n 分钟内状态发生变化的文件（例如权限）。

-ctime n：查找在 n\*24 小时内状态发生变化的文件（例如权限）。

-mmin n：查找在 n 分钟内被修改过的文件。

-mtime n：查找在 n\*24 小时内被修改过的文件。

9.2 locate 快速定位文件路径

locate 指令利用事先建立的系统中所有文件名称及路径的 locate 数据库实现快速定位给定的文件。Locate 指令无需遍历整个文件系统，查询速度较快。为了保证查询结果的准确度，管理员必须定期更新 locate 时刻。

由于 locate 指令基于数据库进行查询，所以第一次运行前，必须使用 updatedb 指令创建 locate 数据库。

1. Updatedb (2)locate 搜索文件

9.3 grep 过滤查找及“ |”管道符

管道符，“ |” ，表示将前一个命令的处理结果输出传递给后面的命令处理

grep 选项 查找内容 源文件

例: ls | grep -n test

1. 磁盘管理

10.1 du 查看文件和目录占用的磁盘空间

du: disk usage 磁盘 占用情况

du [选项] 目录/文件 （功能描述：显示目录下每个子目录的磁盘使用情况）

-a ：列出所有的文件与目录容量，因为默认仅统计目录底下的文件量而已。

-h ：以人们较易读的容量格式 (G/M) 显示；

-s ：列出总量而已，而不列出每个各别的目录占用容量；

-S ：不包括子目录下的总计，与 -s 有点差别。

-k ：以 KBytes 列出容量显示；

-m ：以 MBytes 列出容量显示；

10.2 df 查看磁盘空间占用空余情况

df: disk free 空余磁盘df 选项 [目录或文件名]（ 功能描述：列出文件系统的整体磁盘使用量，检查文件系统的磁盘空间占用)用情况）

-a ：列出所有的文件系统，包括系统特有的 /proc 等文件系统；

-k ：以 KBytes 的容量显示各文件系统；

-m ：以 MBytes 的容量显示各文件系统；

-h ：以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示；

-H ：以 M=1000K 取代 M=1024K 的进位方式；

-T ：显示文件系统类型, 连同该 partition 的 filesystem 名称 (例如 ext3) 也列出；

-i ：不用硬盘容量，而以 inode 的数量来显示

10.3 lsblk 查看设备挂载情况

lsblk [-f查看详细的设备挂载情况，显示文件系统信息]（ 功能描述：查看设备挂载情况）

10.4 mount/umount 挂载/卸载

不作详细记录

10.5 fdisk 分区

fdisk -l （功能描述：查看磁盘分区详情）

fdisk 硬盘设备名 （功能描述：对新增硬盘进行分区操作）

功能说明

（1 ）Linux 分区

Device ：分区序列

Boot ：引导

Start ：从X磁柱开始

End ：到Y磁柱结束

Blocks ：容量

Id ：分区类型ID

System ：分区类型

（2 ）分区操作按键说明

m ：显示命令列表

p ：显示当前磁盘分区

n ：新增分区

w ：写入分区信息并退出

q ：不保存分区信息直接退出

1. 进程管理

11.1 ps 查看当前系统进程状态

ps:process status 进程状态

ps aux | grep xxx （ 功能描述：查看系统中所有进程）

ps -ef | grep xxx （ 功能描述：可以查看子父进程之间的关系）

a 列出带有终端的所有用户的进程

x 列出当前用户的所有进程，包括没有终端的进程

u 面向用户友好的显示风格

-e 列出所有进程

-u 列出某个用户关联的所有进程

-f 显示完整格式的进程列表

**ps aux 显示信息说明**

USER ：该进程是由哪个用户产生的

PID ：进程的 ID 号

%CPU ：该进程 占用 CPU 资源的百分比， 占用越高，进程越耗费资源；

%MEM ：该进程 占用物理内存的百分比， 占用越高，进程越耗费资源；

VSZ ：该进程 占用虚拟内存的大小，单位 KB；

RSS ：该进程 占用实际物理内存的大小，单位 KB；

TTY ：该进程是在哪个终端中运行的。对于 CentOS 来说，tty 1 是图形化终端，

tty2-tty6 是本地的字符界面终端。pts/0-255 代表虚拟终端。

STAT：进程状态。常见的状态有：R ：运行状态、S ：睡眠状态、T ：暂停状态、

Z ：僵尸状态、s ：包含子进程、l ：多线程、+ ：前台显示

START：该进程的启动时间

TIME: 该进程占用CPU 的运算时间，注意不是系统时间

COMMAND: 产生此进程的命令名

**ps-ef 显示信息说明**

UID: 用户ID

PID: 进程 ID

PPID: 父进程ID

C:CPU 用于计算执行优先级的因子。数值越大，表明进程是CPU 密集型运算，

执行优先级会降低；数值越小，表明进程是I/O密集型运算，执行优先级会提高

STIME: 进程启动的时间

TTY: 完整的终端名称

TIME:CPU 时间

CMD: 启动进程所用的命令和参数

11.2 kill 终止进程

kill [-9 表示强迫程序立即停止] 进程号 （ 功能描述：通过进程号杀死进程）

killall 进程名称 （ 功能描述：通过进程名称杀死进程，也支持通配符，这在系统因负载过大而变得很慢时很有用）

11.3 pstree 查看进程树

pstree [-p显示进程的PID -u显示进程所属用户]

11.4 top 实时监控系统进程状态

top [选项]

-d 秒数 指定 top 命令每隔几秒更新。默认是 3 秒在 top 命令的交互模式当 中可以执行的命令。

-i 使 top 不显示任何闲置或者僵死进程。

-p 通过指定监控进程 I D 来仅仅监控某个进程的状态。

**操作说明:**

P 以 CPU 使用率排序，默认就是此项

M 以内存的使用率排序

N 以 PID 排序

q 退出 top

**查询结果字段解释:**

第一行信息为任务队列信息

|  |  |
| --- | --- |
| 内容 | 说明 |
| 12 :26 :46 | 系统当前时间 |
| u p 1 day, 13 :32 | 系统的运行时间，本机已经运行 1天  13小时 32分钟 |
| 2 users | 当前登录了两个用户 |
| load average : 0.00, 0.00, 0.00 | 系统在之前 1 分钟，5 分钟， 15 分钟的平均负 载 。一般认为小于 1 时，负载较小 。如果大于 1 ，系统已经超出负荷。 |

第二行为进程信息

|  |  |
| --- | --- |
| Tas ks : 95 tota l | 系统中的进程总数 |
| 1 running | 正在运行的进程数 |
| 94 sleeping | 睡眠的进程 |
| 0 stopped | 正在停止的进程 |
| 0 zombie | 僵尸进程。如果不是 0 ，需要手工检查僵尸进程 |

第三行为 CPU 信息

|  |  |
| --- | --- |
| Cpu(s): 0. 1%us | 用户模式 占用的 CPU百分比 |
| 0. 1%sy | 系统模式 占用的 CPU百分比 |
| 0.0%ni | 改变过优先级的用户进程 占用的 CPU百分比 |
| 99.7%id | 空闲 CPU的 CPU百分比 |
| 0. 1%wa | 等待输入/输出的进程的 占用 CPU百分比 |
| 0.0%hi | 硬中断请求服务 占用的 CPU百分比 |
| 0. 1%si | 软中断请求服务 占用的 CPU百分比 |
| 0.0%st | st（Steal time）虚拟时间百分比。就是当有虚拟机时，虚拟 CPU 等待实际 CPU 的时间百分比。 |

第四行为物理内存信息

|  |  |
| --- | --- |
| Mem: 625344k total | 物理内存的总量，单位 KB |
| 571504k used | 已经使用的物理内存数量 |
| 53840k free | 空闲的物理内存数量，我们使用的是虚拟机，总 共只分配了 628MB 内存，所以只有 53MB 的空闲内存了. |
| 65800k buffers | 作为缓冲的内存数量 |

第五行为交换分区（ swap ）信息

|  |  |
| --- | --- |
| Swap: 524280k total | 交换分区（虚拟内存）的总大小 |
| 0k used | 已经使用的交互分区的大小 |
| 524280k free | 空闲交换分区的大小 |
| 409280k cached | 作为缓存的交互分区的大小 |

11.5 netstat 显示网络状态和端口占用信息

netstat -anp | grep 进程号 （功能描述：查看该进程网络信息）

netstat -nlp | grep 端口号 （功能描述：查看网络端口号占用情况）

-a 显示所有正在监听（listen ）和未监听的套接字（ socket）

-n 拒绝显示别名，能显示数字的全部转化成数字

-l 仅列出在监听的服务状态

-p 表示显示哪个进程在调用

11.6 crontab 系统定时任务

crontab [-e编辑crontab定时任务 -l查询crontab任务 -r删除当前用户所有的crontab任务]

（1 ）进入 crontab 编辑界面。会打开 vim 编辑你的工作。

\* \* \* \* \* 执行的任务

|  |  |  |
| --- | --- | --- |
| 项目 | 含义 | 范围 |
| 第一个“\* ” | 一小时当中的第几分钟 | 0-59 |
| 第二个“\* ” | 一天当中的第几小时 | 0-23 |
| 第三个“\* ” | 一个月当中的第几天 | 1-31 |
| 第四个“\* ” | 一年当中的第几月 | 1- 12 |
| 第五个“\* ” | 一周当中的星期几 | 0-7 （ 0 和 7 都 代 表 星 期  日） |

（2 ）特殊符号

|  |  |
| --- | --- |
| 特殊符号 | 含义 |
| \* | 代表任何时间。 比如第一个“\* ”就代表一小时中每分钟  都执行一次的意思。 |
| , | 代表不连续的时间。 比如“0 8, 12, 16 \* \* \* 命令 ”，就代表  在每天的 8 点 0 分，12 点 0 分，16 点 0 分都执行一次命令 |
| - | 代表连续的时间范围。 比如“0 5 \* \* 1-6 命令 ”，代表在  周一到周六的凌晨 5 点 0 分执行命令 |
| \*/n | 代表每隔多久执行一次。 比如“\*/ 10 \* \* \* \* 命令 ”，代  表每隔 10 分钟就执行一遍命令 |

（3 ）特定时间执行命令

|  |  |
| --- | --- |
| 时间 | 含义 |
| 45 22 \* \* \* 命令 | 每天 22 点 45 分执行命令 |
| 0 17 \* \* 1 命令 | 每周 1 的 17 点 0 分执行命令 |
| 0 5 1, 15 \* \* 命令 | 每月 1 号和 15 号的凌晨 5 点 0 分执行命令 |
| 40 4 \* \* 1-5 命令 | 每周一到周五的凌晨 4 点 40 分执行命令 |
| \*/ 10 4 \* \* \* 命令 | 每天的凌晨 4 点，每隔 10 分钟执行一次命令 |
| 0 0 1, 15 \* 1 命令 | 每月 1 号和 15 号，每周 1 的 0 点 0 分都会执行命令 。注 意：星期几和几号最好不要同时出现，因为他们定义的都  是天。非常容易让管理员混乱。 |

1. RPM软件包管理

12.1 RPM 查询命令

rpm -qa （功能描述：查询所安装的所有 rpm 软件包）

12.2 RPM 卸载命令

（1）rpm -e RPM软件包 （功能描述：卸载软件包并检查依赖）

（2）rpm -e --nodeps 软件包（功能描述：卸载软件时不检查依赖。使用该软件包的软件之后可能就不能正常工作了。）

12.3 RPM 安装命令

rpm -ivh RPM 包全名

-i install ，安装

-v --verbose ，显示详细信息

-h --hash ，进度条

--nodeps 安装前不检查依赖

1. YUM自动处理依赖性关系安装

13.1 yum 基础

yum [选项][参数]

|  |  |
| --- | --- |
| 选项 | 功能 |
| -y | 对所有提问都回答“yes” |
| 参数 | 功能 |
| install | 安装rpm软件包 |
| update | 更新rpm软件包 |
| check-update | 检查是否有可用的更新rpm软件包 |
| remove | 删除指定的rpm软件包 |
| list | 显示软件包信息 |
| clean | 清理yum过期的缓存 |
| deplist | 显示yum软件包的所有依赖关系 |

13.2 yum 修改下载源