**Linux系统**

**2019/09/17**

1. Linux由unix演变而来的开源系统
2. **支撑互联网的开源技术：**LAMP架构

*L:Linux 操作系统*

*A:Apache Web服务器*

*M:Mysql 数据库*

*P:PHP 编程语言*

1. **系统分区**
2. 分区：把大硬盘分为小的逻辑分区（主分区和扩展分区）；
3. 格式化：写入文件系统，建立数据块（block）4kb；
4. 分区设备文件名；
5. 挂载：给每个分区分配挂载点，包括：/根分区、swap分区（虚拟内存）、/boot启动分区
6. Linux输入区分大小写，而windows输入不区分大小写。Linux中所有内容以文件形式保存，包括硬件。Linux中文件没有扩展名，不能靠扩展名区分文件，所有存储设备都必须挂载之后用户才能使用

(date -d “1970-01-01 16066 days” 将16066换算成年月日)

1. linux各个目录及作用

|  |  |
| --- | --- |
| 目录 | 作用 |
| /bin/ | 存放系统命令的目录，普通用户和超级用户都可以执行 |
| /sbin/ | 保存和系统环境设置相关的命令，只有超级用户可以使用 |
| /usr/bin/ |  |
| /usr/sbin/ |  |
| /boot/ | 系统启动目录（200mb） |
| /dev/ | 保存所有硬件设备文件 |
| /etc/ | 保存配置文件 |
| /home/ | 家目录，创建每个用户时，每个用户要有一个默认登录位置，如/home/luolin/ |
| /lib/ | 系统调用的函数库位置 |
| /lost+found/ | 系统以外崩溃或者机器以外关机，产生的一些文件碎片放在这里，该目录只在每个分区中出现 |
| /media/ | 挂载目录，系统建议挂载软盘和光盘 |
| /mnt/ | 挂载目录， |
| /misc/ | 挂载目录 |
| /opt/ | 第三方安装软件保存位置 |
| /proc/ | 虚拟文件系统，保存在内存当中 |
| /sys/ | 虚拟文件系统，保存在内存当中 |
| /root/ | 超级用户家目录，普通用户家目录在/home/下 |
| /srv/ | 服务数据目录， |
| /tmp/ | 临时目录 |
| /usr/ | 系统软件资源目录，系统中安装的软件大多数保存在这里 |
| /var/ | 动态数据保存位置，主要保存缓存、日志等文件 |

1. **常用命令**

命令格式：命令 [-选项] [参数]

**6.1 文件处理命令**

**文件类型：**-二进制文件；d目录；l软连接文件

**文件权限：**r读；w写；x执行

举例：-rw-r--r--：*表示这是一个二进制文件(-)，其中所有者(rw-)具有读和写的权限，所属组(r--)具有读的权限，其他人(r--)具有读的权限*

|  |  |
| --- | --- |
| ① ls | |
| **英文原意** | list |
| **执行权限** | 所有用户 |
| **功能描述** | 显示目录文件 |
| **选项：**-a显示使用文件；-l详细信息显示；-d查看目录属性；-h以直观的形式显示；-i查看文件的i节点号 | |

**6.2 目录处理命令**

|  |  |
| --- | --- |
| ② mkdir | |
| **英文原意** | make directories |
| **执行权限** | 所有用户 |
| **功能描述** | 创建新目录 |
| mkdir /tmp/luolin | 在tmp目录下创建名为luolin的目录 |
| **可以使用递归创建多个目录：-p**  如：mkdir /tmp/LL/luolin 错误，LL目录不存在  mkdir -p /tmp/LL/luolin 正确，-p递归创建两个目录 | |

|  |  |
| --- | --- |
| ③ cd (shell内置命令) | |
| **英文原意** | Change directory |
| **执行权限** | 所有用户 |
| **功能描述** | 切换目录 |
| **cd /tmp/LL** 切换到指定目录  **cd ..** 切换到上一级目录（.表示当前目录；..表示上一级目录） | |

|  |  |
| --- | --- |
| ④ pwd | |
| **英文原意** | Print working directory |
| **执行权限** | 所有用户 |
| **功能描述** | 显示当前目录 |

|  |  |
| --- | --- |
| ⑤ rmdir | |
| **英文原意** | Remove directories |
| **执行权限** | 所有用户 |
| **功能描述** | 只能删除**空**目录 |
| **Rmdir /tmp/LL/L1** 删除L1空目录 | |

|  |  |
| --- | --- |
| ⑥ cp | |
| **英文原意** | copy |
| **执行权限** | 所有用户 |
| **功能描述** | 复制文件或者目录 |
| cp -rp [源文件、目录][目标目录]  -r复制目录（复制目录必须加上-r）  -p保留文件属性（文件修改时间不会改变）  Cp -r /tmp/LL /root/HH 将目录LL复制到root下并更改名字为HH | |

|  |  |
| --- | --- |
| ⑦ mv | |
| **英文原意** | move |
| **执行权限** | 所有用户 |
| **功能描述** | 剪切文件、或者改名 |
| mv /tmp/LL/L2 /root | 将L2目录剪切到root目录下 |
| cp -rp [源文件、目录][目标目录]  -r复制目录（复制目录必须加上-r）  -p保留文件属性（文件修改时间不会改变）  Cp -r /tmp/LL /root/HH 将目录LL复制到root下并更改名字为HH | |

|  |  |
| --- | --- |
| ⑧ rm | |
| **英文原意** | remove |
| **执行权限** | 所有用户 |
| **功能描述** | 删除文件 |
| rm -rf [源文件、目录][目标目录]  -r删除目录  -f强制执行 | |

**6.3 文件处理命令**

|  |  |
| --- | --- |
| ⑧ touch | |
| **英文原意** | touch |
| **执行权限** | 所有用户 |
| **功能描述** | 创建空文件 |
| Touch LPL 创建名为LPL的文件  Touch LPL LCK 创建名为LPL和LCK的两个文件  Touch “LPL LCK” 创建名为LPL LCK 的一个文件 | |

|  |  |
| --- | --- |
| ⑨ cat | |
| **英文原意** | cat |
| **执行权限** | 所有用户 |
| **功能描述** | 显示文件内容 |
| Cat /etc/issue 查看issue文件的内容  Cat -n /etc/issue 查看文件内容的同时，显示行号 | |

|  |  |
| --- | --- |
| ⑩ tac | |
| **英文原意** | tac |
| **执行权限** | 所有用户 |
| **功能描述** | 倒序显示文件内容 |

|  |  |
| --- | --- |
| 11 more | |
| **英文原意** | more |
| **执行权限** | 所有用户 |
| **功能描述** | 分页显示文件内容 |
| More /etc/services  空格：翻页  回车：按行翻页  按q键：退出 | |

|  |  |
| --- | --- |
| 12 less | |
| **英文原意** | more |
| **执行权限** | 所有用户 |
| **功能描述** | 分页显示文件内容(可以向上翻页或者搜索) |
| less /etc/services  空格：翻页  回车：按行翻页  按q键：退出  Pgup键：按页向上翻页  上箭头：按行向上翻  /LL 搜索LL | |

|  |  |
| --- | --- |
| 13 head | |
| **英文原意** | head |
| **执行权限** | 所有用户 |
| **功能描述** | 显示文件前面指定几行（默认10行） |
| head -n 20 /etc/services 显示前面20行内容 | |

|  |  |
| --- | --- |
| 14 tail | |
| **英文原意** | tail |
| **执行权限** | 所有用户 |
| **功能描述** | 显示文件后面几行（默认10行） |
| tail /etc/services  -n 指定行数  -f动态显示文件末尾内容 | |

|  |  |
| --- | --- |
| 15 ln | |
| **英文原意** | link |
| **执行权限** | 所有用户 |
| **功能描述** | 生成链接文件（默认为硬链接） |
| ln -s [源文件][目标文件]  -s 创建软链接  软链接：类似于windows中的快捷方式，软文件与源文件的i节点号不同  硬链接：类似于拷贝，但是硬文件与源文件可以同步更新，但是如果源文件丢失，硬链接文件仍然存在。硬链接文件与源文件的i节点号相同，硬链接不同分区，如根分区的文件不能硬链接到boot分区或者swap分区中，但是软链接可以。另外硬链接不能作用于目录，但是软链接可以 | |

**7 权限管理命令**

|  |  |
| --- | --- |
| 16 chmod | |
| **英文原意** | Change the permissions mode of a file |
| **执行权限** | 所有者和root |
| **功能描述** | 改变文件或者目录权限 |
| 语法：①chmod [u/g/o/a+-=r/w/x] [文件或者目录]  如：chmod u-r.g+w /tmp/LL 将LL的所有者减少读权限，同时将LL的所属组增加写权限  ②数字表示(r-4; w-2; x-1) Chmod 641(rw-r----w) /tmp/LL  -R 改变目录以及该目录下所以文件和目录的权限 | |

对r/w/x的补充：对某个目录具有w权限，表示可以在该目录下创建和删除文件，即使对该文件没有w权限，文件的w权限表示可以对该文件进行写内容，并不表示对该文件具有删除的权限。（区分文件和目录的权限的区别）

|  |  |
| --- | --- |
| 17 chown | |
| **英文原意** | Change owner |
| **执行权限** | root |
| **功能描述** | 改变文件或者目录的所有者 |
| 语法：chown root /tmp/LL 将LL的所有者改为root | |

|  |  |
| --- | --- |
| 18 chgrp | |
| **英文原意** | Change group |
| **执行权限** | root |
| **功能描述** | 改变文件或者目录的所属组 |
| 语法：chgrp group1 /tmp/LL 将LL的所属组改为group1 | |

|  |  |
| --- | --- |
| 19 umask | |
| **英文原意** | The user file-ceration mask |
| **执行权限** | 所有用户 |
| **功能描述** | 显示、设置文件的缺省权限 |
| 语法：umask -S  新建目录具有x（执行权限），而新建一个文件没有x，即创建了文件不具备可执行权限 | |

**8、文件搜索命令**

|  |  |
| --- | --- |
| 20 find | |
| **英文原意** | find |
| **执行权限** | 所有用户 |
| **功能描述** | 文件搜索 |
| 语法：**find [搜索范围][匹配条件]** （注意区分文件名字大小写）  **find /etc -name init**  在目录/etc中查找名为init 的文件  **find /etc -name \*init\*** 查找含有init字段的文件通配符(\*)表示任意0个或者多个字符  **find /etc -name init???** 查找以init开头的文件，通配符(?)表示匹配任意一个字符  **find /etc -iname init** 不同分大小写，查找init或者INIT  **find /etc -size +204800** 在/etc目录下查找大于100MB的文件，（linux存在文件的单位为数据块，一个数据块为512字节，也就是0.5K，100MB=102400k=204800个数据块）（+n表示大于n；-n表示小于n，=n表示等于n）  **find /home -user LUOLIN** 在/home目录下查找所有者为LUOLIN的文件或者目录  **find /etc -cmin -5** 在/etc目录下查找5分钟内被修改过的属性文件和目录  （-amin 访问时间 access  -cmin 文件属性 change  -mmin 文件内容 modify）  **find /etc -size +163840 -a -size -204800** 在/etc目录下查找大于80MB小于100MB的文件（连接符-a 表示and，-o 表示or）  **find /etc -name init -a -type f** 在/etc目录下查找名为init的文件（f:文件； d：目录； l:软链接文件）  **find /etc -name inittab -exec/-ok ls -l {} \;** 在/etc目录下查找名为inittab的文件并且展示该文件； -exec 命令1：表示对查找得到的文件执行命令1，后面必须加上{} \;  -ok 命令1 ：表示对查找得到的文件执行命令1并且会询问你是否执行  **find /etc -inum 31531 -ok rm {} \;** 在/etc目录下查找i节点为31531的文件并且删除 | |

|  |  |
| --- | --- |
| 21 locate | |
| **英文原意** | locate |
| **执行权限** | 所有用户 |
| **功能描述** | 在文件资料库中查找文件 |

|  |
| --- |
| 语法：locate 文件名  Locate -i 文件名 不区分大小写  Locate查找文件比find快，因为lunix会维护一个文件数据库，使用locat令直接从该数据库中进行查找，因此会快一点。可以使用手动updatedb来更新该数据库， |

|  |  |
| --- | --- |
| 22 which | |
| **英文原意** | which |
| **执行权限** | 所有用户 |
| **功能描述** | 搜索命令所在目录及别名信息 |
| 语法：which 命令  Which ls 查看命令ls所在目录及别名信息 | |

|  |  |
| --- | --- |
| 23 whereis | |
| **英文原意** | whereis |
| **执行权限** | 所有用户 |
| **功能描述** | 搜索命令所在目录及帮助文档路径息 |
| 语法：whereis 命令  Whereis ls 查找ls命令所在目录以及帮助文档 | |

|  |  |
| --- | --- |
| 24 grep | |
| **英文原意** | grep |
| **执行权限** | 所有用户 |
| **功能描述** | 在文件中搜寻字符串匹配的行并输出 |
| 语法：grep -iv [指定字符串][文件]  -i 不区分大小写  -v 排除指定字符串  **Grep -i multi-user /etc/inittab** 在inittab中显示与multi-user匹配的行并且输出（-i表示不区分大小写）  **Grep -v ^# /etc/inittab** 在inittab文件中不显示以#开头的行；^表示开头，^#表示以#开头，以#开头的问注释行 | |

1. **帮助命令**

|  |  |
| --- | --- |
| 25 man | |
| **英文原意** | manual |
| **执行权限** | 所有用户 |
| **功能描述** | 获得帮助信息 |
| 语法：man [命令或者配置文件]  **Man ls**  查看命令ls的帮助信息， 输入：\-l 查看具体选项的信息  **Man services** 查看配置文件services的信息  **Man passwd** passwd既是命令又是配置文件，使用man会优先将其作为命令，1：命令；5 配置文件 。使用man 5 passwd 表示将passwd作为配置文件  如果只要查看简短可以使用：**whatis 命令； apropos 配置文件**  如果只是查看命令选项：**命令 --help** | |

|  |  |
| --- | --- |
| 26 help | |
| **英文原意** | help |
| **执行权限** | 所有用户 |
| **功能描述** | 获得shell内置命令的帮助信息 |
| 语法：help shell内置命令 | |

**10、用户管理命令**

|  |  |
| --- | --- |
| 27 useradd | |
| **英文原意** | User add |
| **执行权限** | root |
| **功能描述** | 添加新用户 |
| 语法：useradd LUOLIN 添加新用户LUOIIN  如果需要为新用户设置密码可以使用：passwd 用户名  Root用户可以更改任何人的密码，且密码可以随意设置。普通用户只能更改自己的密码，且密码必须符合规范亚要求，不能使用字典中的单词 | |

|  |  |
| --- | --- |
| 28 who | |
| **英文原意** | who |
| **执行权限** | 所有用户 |
| **功能描述** | 查看登录用户信息 |
| 语法：who  结果：root（用户） tty1(本地终端) 2019-09-25(时间)  Luolin(用户) pts/1(远程终端) 2019-09-25(时间) | |

|  |  |
| --- | --- |
| 29 w | |
| **英文原意** | w |
| **执行权限** | 所有用户 |
| **功能描述** | 查看登录用户的详细信息 |
| 语法：w | |

**11、压缩/解压缩命令**

|  |  |
| --- | --- |
| 30 gzip | |
| **英文原意** | gzip |
| **执行权限** | 所有用户 |
| **功能描述** | 压缩文件（不能压缩目录，其不能保留源文件） |
| 语法：gzip 文件 | |

|  |  |
| --- | --- |
| 31 gunzip | |
| **英文原意** | Gun unzip |
| **执行权限** | 所有用户 |
| **功能描述** | 解压缩.gz的压缩文件 |
| 语法：gunzip 压缩文件 | |

|  |  |
| --- | --- |
| 32 tar | |
| **英文原意** | tar |
| **执行权限** | 所有用户 |
| **功能描述** | 打包目录 |
| 语法：tar -cvf [压缩后文件名] [目录]  -c 打包  -v 显示详细信息  -f 指定文件名  -z 打包同时压缩  -x 解包 | |

|  |  |
| --- | --- |
| 33 zip | |
| **英文原意** | zip |
| **执行权限** | 所有用户 |
| **功能描述** | 压缩文件或者目录（能保留源文件） |
| 语法：zip -r [压缩后文件名][文件或者目录]  -r 压缩目录 | |

|  |  |
| --- | --- |
| 34 unzip | |
| **英文原意** | unzip |
| **执行权限** | 所有用户 |
| **功能描述** | 解压缩.zip的压缩文件 |
| 语法：unzip 压缩文件 | |

|  |  |
| --- | --- |
| 35 bzip2 | |
| **英文原意** | Bzip2 |
| **执行权限** | 所有用户 |
| **功能描述** | 压缩文件（适合压缩大文件） |
| 语法：bzip2 -k 文件  -k 产生压缩文件后保留源文件 | |

**12、网络命令**

|  |  |
| --- | --- |
| 36 write | |
| **英文原意** | write |
| **执行权限** | 所有用户 |
| **功能描述** | 给**在线**用户发信息,以ctrl+ D保存结束 |
| 语法：write 用户名 | |

|  |  |
| --- | --- |
| 37 wall | |
| **英文原意** | Write all |
| **执行权限** | 所有用户 |
| **功能描述** | 给所有**在线**用户发信息,以ctrl+ D保存结束，发广播信息 |
| 语法：write 信息内容 | |

|  |  |
| --- | --- |
| 38 ping | |
| **英文原意** | ping |
| **执行权限** | 所有用户 |
| **功能描述** | 测试网络连通性 |
| 语法：ping -c ip地址  -c 指定发送次数 | |

|  |  |
| --- | --- |
| 39 ifconfig | |
| **英文原意** | Interface configre |
| **执行权限** | root |
| **功能描述** | 查看或者设置网卡地址 |
| 语法：ifconfig eth0 192.168.1.12 设置ip地址 | |

|  |  |
| --- | --- |
| 40 mail | |
| **英文原意** | mail |
| **执行权限** | 所有用户 |
| **功能描述** | 查看发送电子邮件（可以发送给未在线的用户） |
| 语法：mail root | |

|  |  |
| --- | --- |
| 41 last | |
| **英文原意** | last |
| **执行权限** | 所有用户 |
| **功能描述** | 列出目前与过去登入系统的用户信息 |
| 语法：mail root | |

|  |  |
| --- | --- |
| 42 lastlog | |
| **英文原意** | lastlog |
| **执行权限** | 所有用户 |
| **功能描述** | 检查用户上次登录的时间 |
| 语法：lastlog  Lastlog -u 502 只查看用户号为502的用户的登录信息 | |

|  |  |
| --- | --- |
| 43 traceroute | |
| **英文原意** | traceroute |
| **执行权限** | 所有用户 |
| **功能描述** | 显示数据包到主机之间的路径 |
| 语法：traceroute www.baidu.com | |

|  |  |
| --- | --- |
| 44 netstat | |
| **英文原意** | netstat |
| **执行权限** | 所有用户 |
| **功能描述** | 显示网络相关信息 |
| 语法：netstat [选项]  -t TCP协议  -u UDP协议  -l 监听  -r 路由  -n 显示ip地址和端口号  **Netstat -tlun** 查看本机监听的端口  **Netstat -an** 查看本机所有的网络连接  **Netstat -rn** 查看本机路由表 | |

|  |  |
| --- | --- |
| 45 setup | |
| **英文原意** | setup |
| **执行权限** | 所有用户 |
| **功能描述** | 配置网络 |
| 语法：setup | |

**13、关机和重启命令**

|  |  |
| --- | --- |
| 46 shutdown | |
| **英文原意** | shutdown |
| **执行权限** | 所有用户 |
| **功能描述** | 关机或者重启 |
| 语法：**shutdown** 选项 时间  -c 取消前一个关机命令  -h 关机  -r 重启  其他关机命令：  **Halt / poweroff / init 0**  其他重启命令：  **Reboot / init 6**  退出登录命令：  **logout** | |

**二、vim编辑器**

①输入 vi 文件名 进入命令模式

②输入i a o 进入插入模式,按esc键返回命令模式

③命令模式下输入冒号:进入编辑模式

**2.1 插入命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| a | 在光标所在字符后插入 |
| A | 在光标所在行尾插入 |
| i | 在光标所在字符前插入 |
| I | 在光标所在行行首插入 |
| o | 在光标下插入新行 |
| O | 在光标上插入新行 |

使用插入命令需要按esc返回命令模式下，才能使用插入命令

**2.2 定位命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| Set nu | 设置行号 |
| Set nonu | 取消行号 |
| gg  G | 到第一行  到最后一行 |
| nG | 到第n行 |
| :n | 到第n行 |
| $ | 移至行尾 |
| 0 | 移至行首 |

**2.3 删除命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| x | 删除光标所在字符 |
| nx | 删除光标所在处后n个字符 |
| Dd  ndd | 删除光标所在行  删除n行 |
| dG | 删除光标所在行到文件末尾的内容 |
| D | 删除光标所在处到行尾的内容 |
| :n1,n2d | 删除指定范围的行 |

**2.4 复制和剪切命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| yy | 复制当前行 |
| nyy | 复制当前行以下n行 |
| dd | 剪切当前行 |
| ndd | 剪切当前行以下n行 |
| p/P | 粘贴在当前行上或者下 |

**2.5 替换和取消命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| r | 取代光标所在处字符 |
| R | 从光标所在行开始替换字符，按esc结束 |
| u | 取消上一步操作 |

**2.6 搜索和替换字符串**

|  |  |
| --- | --- |
| 命令 | 作用 |
| /string | 搜索指定字符串  如果忽略大小写使用：set ic  取消忽略大小写：set unic |
| n | 搜索指定字符串的下一出现位置 |
| %s/old/new/g | 全文替换指定字符串%s表示全文 |
| :n1,n2s/old/new/g | 在一定范围能替换指定字符串 |

**2.7 保存和退出命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| w | 保存修改 |
| :w new\_filename | 另存为指定文件 |
| :wq | 保存修改并退出 |
| ZZ | 快捷键，保存修改并退出 |
| :wq! | 强行保存修改并且退出（文件所有者及root可使用） |

**2.8 其他命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| :r 文件名 | 将文件导入到光标所在位置 |
| : r !命令 | 把命令所执行的结果导入到光标所在位置  如：:r !date 将date命令执行的结果导入到光标所在位置 |
| :map 快捷键 触发命令 |  |
| :n1,n2s/^/字符/g | 将n1到n2行行首加上字符 |
| :n1,n2s/^字符//g | 将n1到n2行行首的字符删除 |
| :ab myname luolin | 将myname 替换为luolin |

**3.1 软件包管理**

软件包：源代码包和二进制包（RPM包、系统默认包）

**3.2 rpm包的依赖性**

①树形依赖

②环形依赖

③模块依赖

**3.3 rpm包的安装和升级**

对于未安装的包需要使用包全名，而安装好之后，只需要使用包名即可，因为安装的包放在/var/lib/rpm下，因此不需要写路径

|  |  |
| --- | --- |
| 命令 | 作用 |
| **rpm -ivh 包全名 安装命令**  -i install 安装  -v verbose 显示详细信息  -h hash 显示进度  --nodeps 不检测依赖性  **rpm -Uvh 包全名 升级命令**  **rpm -e 包名 卸载 卸载命令**  **-e erase** | Rpm包安装  (rpm包存在centos镜像文件中的package下，安装时，首先创建挂载点(mkdir /mnt/cdrom)，然后将镜像文件进行挂载（mont /dev/cdrom /mot/cdrom）) |

使用rpm命令手动会导致依赖性问题，十分复杂。因此一般使用yum进行安装，但是查询需要使用rpm命令

**3.4 rpm包的查询**

|  |  |
| --- | --- |
| 命令 | 作用 |
| rpm -q 包名  -q query | 查询命令 |
| rpm -qa | 查询所有已经安装的rpm包 |
| rpm -qi | 查询软件信息 |
| rpm -qip 包全名 | 查询未安装包 |
| rpm -ql 包名 | 查询软件包安装位置 |
| rpm -qlp 包全名 | 查询未安装包位置 |
| rpm -qf 系统文件名 | 查询系统文件名属于哪个软件包(-f file) |
| rpm -qR 包名 | 查询软件包的依赖性 |
| rpm -qRp 包全名 | 查询未安装软件包的依赖性 |

**3.5 rpm包校验和文件提取**

|  |  |
| --- | --- |
| 命令 | 作用 |
| rpm -V 已安装的包名  -V verify | 校验指定RPM包中的文件 |
| rpm2cpio 包全名 | \ cpio -idv .文件绝地路径  cpio : 是一个标准工具，用于创建软件档案文件和从档案文件中提取文件  rpm2cpio ：将rpm包转换为cpio格式 | rpm包中文件提取 |

**3.6 rpm 包管理 -yum在线管理**

**①ip地址配置和网络yum源**

|  |  |
| --- | --- |
| 命令 | 作用 |
| yum list | 查询所有可以软件包列表 |
| yum search 关键字 | 搜索服务器上所有和关键字相关的包 |
| yum -y install 包名 | 安装  -y 表示自动回答yes |
| yum -y update 包名 | 升级软件  -y 表示自动回答yes |
| yum -y remove 包名 | 卸载（依赖包也会卸载） |
| yum grouplist | 列出所有可用的软件组列表 |
| yum groupinstall 软件组名 | 安装软件组名 |
| Yum groupremove 软件组名 | 卸载指定软件组 |

**②光盘yum源搭建**

首先进行光谱挂载：mount /dev/sr0 /mnt/cdrom

然后是网络yum源失效（替换后缀名repo）

修改光盘yum源文件(vi CentOS-Media.repo)

**3.7 源码包与rpm包的安装区别：**源码包安装在用户自定义目录，而rpm包安装在默认目录，对于rpm包可以使用service，因为service可以去默认路径下寻找rpm包。

**4.1 用户配置文件**

①用户信息文件 /etc/passwd

②影子文件/etc/shadow

③组信息文件/etc/group和组密码文件/etc/gshadow

**4.3 用户管理相关文件**

**1、用户家目录**

普通用户：/home/用户名 ,所有者和所属组都是此用户，权限是700

超级用户：/root/ , 所有者和所属组都是root用户，权限为500

1. **用户的默认邮箱位置：**/var/spool/mail/用户名/
2. **用户模板目录：**/etc/skel/ ，每添加一个新的目录就会把该目录下的模板复制到用户的家目录下，使用ls -a可以显示隐藏文件

**4.4 用户管理命令**

|  |  |
| --- | --- |
| 命令 | 作用 |
| useradd 用户名 | 创建一个用户 |
| passwd [选项]用户名  -S 查看用户密码状态  -l 锁定用户状态  -u 解锁用户 | 添加或更改密码  root 可以更改任何用户的密码，  普通用户只能更改自己的密码，直接输入passwd回车即可 |
| usermod [选项]用户名  -c 用户说明 ：修改用户说明  -G root :把用户加入到root组  -L :锁定用户  -U ：解锁用户 | 修改已经存在的用户 |
| chage [选项]用户名  -l: 列出用户详细密码状态  chage -d 0 用户名：将密码修改日期归0 ，可以使得用户一登录就要修改密码 | 修改用户密码状态 |
| Userdel [选项]用户名  -r 删除用户的同时删除用户家目录 | 删除用户 |
| id 用户名 | 查看用户的id |
| su [选项]用户名  - :连带用户的环境变量一起切换  -c 命令：仅执行一次命令，不用切换用户身份  su - root -c “useradd luolin1” :不切换root但是利用root的身份执行添加用户命令 | 切换用户身份 |
| env | 查看当前用户环境 |

**5、权限管理**

**5.1 ACL权限**

ACL权限用于解决身份不够用的情况（身份：所有者，所属组，其他人），即忽略身份权限，给某一个人分配权限。

|  |  |
| --- | --- |
| 命令 | 功能 |
| **getfacl 文件名** | 查看acl权限 |
| **setfacl -m u:用户名:权限 文件**  **如：setfacl -m u:luolin:rx /project** | 设置用户对文件的acl权限  设置用户luolin对project的权限为rx |
| **Setfacl -m g:组名:权限 文件** | 设置组的acl权限 |
| **Setfacl -m m:rx 文件**  （最大有效权限mask是用来指定最大有效权限的，我们对用户赋予acl权限时，需要与mask权限通过与运算得到的权限，才是真正的权限。） | 给文件设定最高mask权限 |
| **setfacl -x u:用户名 文件名**  **setfacl -x g:组名 文件名** | 删除指定用户或者组的acl权限 |
| **setfacl -b 文件包** | 删除文件所有的acl权限 |
| **setfacl -m u:用户名:权限 -P 文件名** | 递归设置文件的acl命令，即对该目录设置acl权限的同时，也递归地对该目录的子目录设置acl权限 |
| **setfacl -m d:u:用户名:权限 文件名** | 设置目录的默认acl权限，那么在该目录下创建新的目录都会继承父目录的acl权限（只针对新创建的目录） |

**5.2 SetUID功能**

SetUID只针对可执行的二进制文件（如命令：passwd），当程序拥有SUID权限后，普通用户在执行该程序时，可以短暂获得root的所拥有的权限，但是程序结束，用户的权限也随之消失。另外该用户首先对该程序具有可执行权限(x)。

|  |  |
| --- | --- |
| 命令 | 功能 |
| **chmod 4755 文件名**  **chmod u+s 文件名** | 改变文件的权限，4表示对所有者增加SUID权限 |
| **chmod 755 文件名** | 删除SUID权限 |

SetGID用户在执行该程序时，用户的所属组临时变为该程序文件的属组，可以针对二进制文件也可以针对目录。或者普通用户在该目录下创建的文件的所属组会变为该目录的属组。

|  |  |
| --- | --- |
| 命令 | 功能 |
| **chmod 2755 文件名**  **chmod g+s 文件名** | 改变文件的权限，2表示对所属组增加GUID权限 |
| **chmod 755 文件名** | 删除GUID权限 |

**Sticky BIT：**只能针对目录，同时普通用户对该目录拥有w和r权限，当目录拥有SBIT粘着位后，普通用户只能删除自己建立的文件，不能删除其他用户建立的文件

**5.3 文件系统属性chattr权限**

|  |  |
| --- | --- |
| **命令** | **功能** |
| chattr [+-=][选项]文件或目录名  + 加权限  - 减权限  = 等于某权限 |  |
|  |  |

**5.4 系统命令权限sudo权限**

**6、文件系统管理**

1、主分区总共只能有4个，扩展分区只能有一个，并且扩展分区作为主分区的一种，因此主分区和扩展分区总共最多有4个。扩展分区不能存储数据和格式化，必须再划分为逻辑分区才能使用。逻辑分区的编号从5开始。

1. **文件系统：**

**①ext2**文件系统，最大支持16TB的分区和最大2TB的文件

**②ext3**文件系统：最大区别就是带用日志功能

**③ext4**文件系统：最大支持1EB文件系统和16TB文件（1E=1024P=1024\*1024TB）

1. **文件系统采用命令**

|  |  |
| --- | --- |
| 命令 | 功能 |
| df [选项][挂载点]  -a 显示所有文件系统信息  -h 使用习惯单位显示大小  -T 显示文件系统类型  -m 以MB为单位显示容量  -k 以KB为单位显示容量（默认方式） | 文件系统查看命令 |
| du [选项][目录或文件名]  -a 显示每一个子文件的磁盘占用量，  -h 使用习惯单位显示大小  -s 统计总占用量，而不列出子目录和子文件的占用量 | 统计目录或文件大小 |

du和df的区别：df从文件系统考虑，除了统计文件占用的空间外，还要统计命令和程序占用的空间，而du命令是面向文件的，只会计算文件或者目录占用的空间。

|  |  |
| --- | --- |
| 命令 | 功能 |
| fsck [选项] 分区设备文件名 -a 不用显示用户提示，自动修复文件系统  -y 自动修复 | 文件系统修复命令 |
| dumpe2fs 分区设备文件名 | 显示磁盘状态命令 |
| mount [-l]  -l 会显示卷标名称 | 查询系统中已经挂载的设备 |
| mount 设备文件名 挂载点  光盘的设备文件为：/dev/cdrom或者/dev/sr0 | 将设备进行挂载（光盘） |
| umont 设备文件名或者挂载点 | 卸载命令（需要在光盘外进行卸载） |
| Mount -t vfat 设备名 挂载点 | 挂载u盘 |

Linux默认不支持ntfs文件系统

1. **fdisk手动分区**

|  |  |
| --- | --- |
| **命令** | **功能** |
| **fdisk -l** | 查看硬盘分区 |
| **第一步fdisk /dev/sdb** | 手动分区 |
| **第二步mkfs -t ext4 /dev/sdb1** | 格式化分区，写入文件系统ext4 |
| **第三步创建挂载点** |  |

**7、Shell编程**

**1、概述：**命令解释器，是一个强大的编程语言，shell解释执行的脚本语言，在shell中可以直接调用linux命令。

**2、shell脚本的执行方式**

|  |  |
| --- | --- |
| **命令** | **功能** |
| echo [选项][输出内容]  -e 支持反斜线控制的字符转换（即可以使用‘\’转义字符） | 输出命令 |
| echo -e "\e[1;31m abcd \e[0m" | 颜色输出 |

3、shell脚本必须以：#!/bin/bash开头

1. 程序的执行：①可以通过绝对路径来执行；②通过bash 调用执行脚本：bash hello.sh
2. Windows上编辑的shell脚本不能在linux上运行，因为windows上编辑的文本的换行符与linux中的换行符不匹配。可以使用:dos2unix 文件名将windows格式的shell转换为linux格式的shell脚本，同样可以使用unix2dos命令。
3. **历史命令和命令补全**

|  |  |
| --- | --- |
| **命令** | **功能** |
| history [选项][历史命令保存文件]  -c 清空历史命令  -w 把缓存中的历史命令写入历史命令保存文件（bash\_history） | 查看历史命令  历史命令默认保存1000条，可以通过环境变量配置文件/etc/profile中进行修改  **使用!n：**可以重新执行第n条命令  **使用!!：**可以重新执行上一条命令  **使用!字符串：**重新执行最后一条以字符串开头的命令 |

使用tab键进行命令或者文件补齐

1. **命令别名**

|  |  |
| --- | --- |
| **命令** | **功能** |
| alias | 重新命令别名 |
| alias 命令别名=’命令1’ | 给命令1其别名，临时生效，要想永久生效，可以在/root/.bashrc中写入相应的命令，这样就能永久生效 |
| unalias 别名 | 删除别名 |

命令优先级：通过绝对路径的命令＞命令别名＞bash内部命令（如cd命令）＞环境变量定义目录找到的第一个命令

1. **输入输出重定向**

**8.1 标准输入输出**

|  |  |  |  |
| --- | --- | --- | --- |
| **设备** | **设备文件名** | **文件描述符（代号）** | **类型** |
| 键盘 | /dev/stdin | 0 | 标准输入 |
| 显示器 | /dev/stdout | 1 | 标准输出 |
| 显示器 | /dev/stderr | 2 | 标准错误输出 |

**8.2输出重定向：**将输出内容重定向到其他文件的过程：

|  |  |
| --- | --- |
| 格式 | 功能 |
| **命令 > 文件**  如：ls > log | 将命令的执行结果保存到文件中，注意会覆盖文件中之前的内容 |
| **命令 >> 文件**  如：ls > log | 将命令的执行结果保存到文件中，且不会覆盖文件中之前的内容，新内容追加到文件末尾 |
| **错误命令 2> 文件**  lst 2> log\_err（lst为错误命令，会得到错误结果） | 将命令的错误输出结果保存到文件中，注意会覆盖文件中之前的内容 |
| **错误命令 2>> 文件**  如：lst 2>> log\_err | 将命令的错误输出结果保存到文件中，且不会覆盖文件中之前的内容，新内容追加到文件末尾 |
| **命令 &>文件 或**  **命令 > 文件 2>&1** | 以覆盖的方式，将正确输出和错误输出的结果保存到同一个文件当中 |
| **命令 &>>文件 或**  **命令 >> 文件 2>&1** | 以追加的方式，将正确输出和错误输出的结果保存到同一个文件当中 |
| **命令 >> 文件a 2>>文件b** | 将正确的结果追加到文件a当中，而把错误的结果追加到文件b当中 |
| **命令 &>/dev/null** | 相当于将命令的执行结果丢入到垃圾箱当中，仅仅是执行命令即可 |

**8.3 输入重定向：将输入方式变为其他方式**

|  |  |
| --- | --- |
| **命令** | **功能** |
| wc [选项][文件名]  -c 统计字节数  -w 统计单词数  -l 统计行数 | 统计输入的字符数量 |
| wc < 文件 | 将文件内容作为输入，使用wc对文件内容进行统计 |

**9 多命令顺序执行和管道符、通配符**

**9.1 多命令顺序执行：**

|  |  |  |
| --- | --- | --- |
| **多命令执行符** | **格式** | **作用** |
| ; | 命令1;命令2 | 多个命令顺序执行，命令之间没有任何逻辑联系 |
| && | 命令1&&命令2 | 逻辑与：  当命令1正确执行，命令2才执行，否则命令2不执行 |
| || | 命令1||命令2 | 逻辑或：  命令1不正确执行，命令2才会执行；  当命令1正确执行，命令2不会执行 |
|  | 命令 && echo “yes” || echo “no” | 可以判断命令是否执行 |

|  |  |  |
| --- | --- | --- |
| **管道符** | **格式** | **作用** |
| | | 命令1 | 命令2 | 命令1的正确输出作为命令2的操作对象，如果命令1报错，则命令2不会执行 |
| 举例 | netstat -an | grep “搜索内容” | 使用netstat -an可以查看网络连接状态，然后将得到的结果再使用grep进行关键字搜索 |

|  |  |
| --- | --- |
| **通配符** | **作用** |
| ? | 匹配任意一个字符 |
| \* | 匹配任何内容 |
| [] | 匹配括号中的任意一个字符，例如[abc]代表一定匹配一个字符，a或b或c |
| [a-z] | 表示匹配a到z之间的任意一个字符 |
| [^0-9] | 表示匹配一个不是数字的字符 |
|  |  |
| **其他字符** | **作用** |
| ‘’ 单引号 | 单引号中的特殊字符无特殊含义，仅仅只是一个字符 |
| “”双引号 | 双引号中的特殊符号没有特殊含义，但是”$”，”\”表示调用变量值和转义字符 |
| $() | 表示先执行括号中的命令， |
| $ | 调用变量的值 |
| \ | 转移符 |

**10 bash 的变量**

**变量分类**

|  |  |
| --- | --- |
| 用户自定义变量（本地变量） |  |
| 环境变量 | 保存和系统操作环境相关的数据，变量名称不能更改，但是变量值可以更改，也可以自己添加环境变量 |
| 位置参数变量 | 用来向脚本当中传递参数或数据的，变量不能自定义，变量作用固定 |
| 预定义变量 | Bash中已经定义好的变量，变量不能自定义，变量作用也是固定的 |

**10.1 用户自定义变量**

**用户自定义变量只在当前的shell中生效**

①在bash中，变量的默认类型都是字符串类型，如果需要进行数值运算，则必须指定变量为数值类型

②变量用等号连接，等号两边不能有空格。

③变量的值如果有空格，需要使用单引号和双引号包括起来

④变量中可以使用“\”转义符

⑤可以将命令的执行结果作为变量的值，可以使用$(命令)来赋值

⑥环境变量命令建议大小，便于区分

|  |  |
| --- | --- |
| **变量定义：** | 如：name=”luo lin” |
| **变量调用：** | 使用$符，如：echo $name； |
| **变量叠加：** | name=123  name=”$name”456  name=${name}789 |
| **set命令** | 查看系统的所有变量 |
| **unset 变量** | 删除变量 |

**10.2 环境变量**

**环境变量会在当前shell和这个shell的所有子shell当中生效。如果把环境变量写入相应的配置文件，那么这个环境变量就会在所有的shell中生效，环境变量相当于全区变量，而自定义变量相当于局部变量。**

|  |  |
| --- | --- |
| export 变量名=变量值 | 环境变量的声明 |
| env | 查看环境变量命令(set是查看所有命令) |
| unset 变量名 | 删除变量 |

**PATH路径：**系统查找命令的路径

**PS1变量：**定义了用户提示符变量，如默认PS1=[\u@\h \W]\$得到的提示符格式为：[root@localhost ~]， 用户可以修改该提示符的格式

**10.3 位置参数变量**

|  |  |
| --- | --- |
| **位置参数变量** | **作用** |
| **$n** | n为数字，$0表示命令本身,$1-$9代表第一个到第九个参数，10以上的参数需要使用大括号包含，如${10} |
| **$\*** | 代表命令行中的所有参数，并且把所有参数看成一个整体 |
| **$@** | 代表命令行中的所有参数，且把每一个参数区分对待 |
| **$#** | 代表命令行中所有参数的个数 |

**10.4 预定义变量**

|  |  |
| --- | --- |
| **预定义变量** | **作用** |
| **$?** | 最后一次执行的命令的结果返回状态，如果值为0，表示上一次命令正确执行，如果值为非0 表示上一次命令没有正确执行 |
| **$$** | 当前进程的进程号(pid) |
| **$!** | 后台运行的最后一个进程的进程号(pid ) |

**接受键盘输入**

|  |  |
| --- | --- |
| **命令** | **功能** |
| read [选项][变量名]  **-p “提示信息”:** 在等待read输入时，会输出提示信息  **-t 秒数:** read命令会一直等待用户输入，可以设定等待时间  **-n 字符数：**read命令只接受指定的字符数，  **-s：** 隐藏输入数据，适用于机密信息的输入 | 接受键盘输入，并保存到变量名中 |

**10.5 bash运算符**

**10.5.1 数值运算与运算符**

|  |  |
| --- | --- |
| **命令** | **作用** |
| declare [+/-][选项] 变量名  -i 将变量声明为整数型(integer)  -x 将变量声明为环境变量(但是常用export进行声明)  -p 显示指定变量被声明的类型 | 声明变量类型  如：a=1  b=2  declare -i c=$a+$b |

除了上述方法外，还可以通过以下形式：

**d=$(($a+$b)) / d=$[$a+$b]**

**10.5.2 变量测试与内容替换**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量置换方式** | **没有设置变量y** | **变量y为空值** | **变量y设置值** |
| x=${y-新值} | x=新值 | x=空值 | x=$y |

**10.6 环境变量配置文件**

|  |  |
| --- | --- |
| **命令** | **作用** |
| source 配置文件  或者：. 配置文件 | 在不用重新登录的情况下，强制让配置文件生效 |

**环境变量配置文件包括：**

|  |  |  |
| --- | --- | --- |
| 1 | /etc/profile | 对所有用户生效(etc目录下的，对所有用户生效) |
| 2 | /etc/profile.d/\*.sh | 对所有用户生效 |
| 3 | ~/.bash\_profile | 对当前用户生效 |
| 4 | ~/.bashrc | 对当前用户生效 |
| 5 | /etc/bashrc | 对所有用户生效 |

**执行顺序为：**1→3/2→4→5。如果想要环境变量永久生效，可以把环境变量写入上述的配置文件当中，但是注意：执行顺序靠后的文件中的变量会覆盖全面的变量

**其他环境变量配置文件：**

|  |  |  |
| --- | --- | --- |
| 1 | ~/.bash\_logout | 注销时生效的环境变量配置文件 |
| 2 | ~/bash\_history | 历史命令的保存文件 |
| 3 | /etc/issue | 本地终端登录信息 |
| 4 | /etc/issue.net | 远程终端登录信息（只能写纯文本信息） |
| 5 | /etc/motd | 不管是本地还是远程登录，都可以显示登录信息（登录后的信息） |

**三、 Shell编程**

**1 正则表达式**

通配符（?/\*/[]）用来在系统中匹配**文件名**。属于完全匹配，如：ls/find等。正则表达式用来在文件中匹配符合条件的**字符串**，属于包含匹配，如grep等。

|  |  |
| --- | --- |
| **元字符** | **作用** |
| \* | 前一个字符匹配0次或任意多次 |
| . | 匹配除换行符以外的任意**一个**字符 |
| ^ | 匹配行首，如：^hello 表示匹配以hello开头的行 |
| $ | 匹配行尾，如：hello$ 表示匹配以hello结尾的行 |
| [] | 匹配括号中的任意**一个**字符，如：[0-9][a-z]表示匹配一位数字和一个字母构成的两位字符 |
| [^] | 匹配除了括号以外的任意**一个**字符 |
| \ | 转义字符，用于用于取消特殊符号的含义 |
| \{n\} | 表示其前面的字符出现**恰好n次**，如：[0-9]\{4\}表示匹配4位数字 |
| \{n,\} | 表示其前面的字符出现**不少于n次**，如：[0-9]\{4,\}表示匹配4位或4位以上的数字 |
| \{n,m\} | 表示其前面的字符出现**至少于n次，最多m次**，如：[0-9]\{4,8\}表示匹配4位到8位数字 |

**2 字符截取命令**

|  |  |
| --- | --- |
| **命令** | **作用** |
| **cut [选项][文件名]**  **-f 列号**  **-d 分隔符**  一般情况下，cut和grep同时使用，cut的使用局限是不能作用于以空格作为分隔符的行 | 提取第几列  按照指定分割符分割列，默认分隔符为制表符，（即tab键）如果采用其他符号作为分隔符，则需要使用-d来说明，如：-d “:” 表示分隔符为”:” |
|  |  |
| **printf ‘输出类型输出格式’ 输出内容**  **输出类型包括：**  **%ns :** 输出n个字符的字符串  **%ni:** 输出n个数字的整数  **%m.nf:** 输出n位小数，m-n位整数的浮点数，总共位数为m位  输出格式包括：  \a:输出警告声音  \b:输出退格键  \f：清除屏幕  \n:换行  \r:回车  \t:水平输出退格符  \v:垂直输出退格键 | 格式化输出内容  如：printf ‘%s %s %s\n’ 1 2 3 4 5 6  输出为：1 2 3  4 5 6 |
|  |  |
| **awk ‘条件1{动作1} 条件2{动作2}...’ 文件名**  条件：一般为关系表达式  动作：格式化输出，流程控制语句 | 使用awk可以使用$来指定列，因此忽略制表符的类型，即使是空格也可以使用awk来提取相应的列。  如：①awk ‘{printf $1 “\t” $3}’ 文件名。表示将文件中第一列($1)和第三列($3)打印，并且中间加上tab键  ②awk ‘BEGIN{printf “this is a shell\n”} {printf $2 “\t” $6 “\n”} ’ 文件名 。其中BEGIN表示在所有数据读取之前，先执行随后的动作  ③awk ‘BEGIN{FS=”:”} {printf $2 “\t” $6 “\n”} ’ 文件名。FS=”:”表示分隔符为“:”，注意需要加上BEGIN，否则第一行数据将不会被正确处理。  ④awk ‘END{printf “this is a shell\n”} {printf $2 “\t” $6 “\n”} ’ 文件名 。其中END表示在所有数据处理完之后，在执行END后面的动作。  ⑤cat student | grep -v ID | awk '$4>=90{printf $2 "\t" $4"\n"}' 。打印成绩大于等于90分的学生姓名和成绩 |
|  |  |

**sed命令 ：**sed主要是用来将数据进行选取、替换、删除、新增的命令。类似于vi或vim。但是vi只能作用于文件，而sed命令可以使用管道符来对某个命令执行的结果进行操作。

|  |  |
| --- | --- |
| **命令** | **作用** |
| **sed [选项]‘[动作]’文件名**  **选项：**  -n:默认sed命令会把所有数据都输出到屏幕，使用-n选项表示只把结果sed命令处理的行输出到屏幕  -e:允许对输入数据应用多条sed命令编辑  -i:使用sed修改的结果直接修改到读取的数据文件，而不是打印到屏幕上  **动作：**  a\: 追加，在当前行后添加一行或者多行  c\: 行替换，用c后面的字符串替换原数据行  i\: 插入，在当前行前插入一行或多行  d: 删除  p: 打印  s: 子串替换，s/旧子串/新字串/g | 操作文件  ①sed -n ‘2p’ 文件名：表示只打印文件中第二行内容  ②sed ‘2,4d’ 文件名 ：表示删除第2行到第4行的数据（不会更改源文件）  ③sed ‘2a hello’ 文件名：表示在文件第二行后面追加hello  ④sed ‘2i hello \ world’ 文件：表示在文件第2行前插入两行数据  ⑤sed ‘2c No such person’ student：表示替换第二行数据  ⑥sed ‘3s/90/100/g’ 文件：表示把文件中第3行中的90替换会100  ⑦sed -i ‘3s/90/100/g’ 文件：表示把文件中第3行中的90替换会100，并且原文件会被改变，不会打印到屏幕上  八sed -e ‘s/Li/La/g ; s/Jia/Ta/g’ 文件：表示同时把Li替换为La把Jia替换为Ta，由于s前没有行号，表示对整个文档进行替换 |

**3字符处理命令**

|  |  |
| --- | --- |
| **命令** | **作用** |
| **sort [选项]文件名**  **选项：**  **-f** 忽略大小写  -**n** 以数值型进行排序  **-r** 反向排序  **-t** 指定分割符，默认是制表符  **-k** n[,m] 按照指定的字段范围进行排序，从n字段开始，m字段结束（默认到行尾） | **排序命令**   1. sort /etc/passwd 排序 2. sort -t “:” -k 3,3 文件名：表示对文件中第3个字段进行排序 |
| **wc [选项]文件名**  **-l** 只统计行数  **-w** 只统计单词数  **-m** 只统计字符数 | **统计命令** |

**4 条件判断命令**

|  |  |
| --- | --- |
| **命令** | **作用** |
| **test -e 文件或[ -e 文件 ]**  **-e** 判断文件是否存在  **-d** 判断该文件是否为目录且是否存在  **-f** 判断该文件是否问普通文件且是否存在  **-r** 判断文件是否存在且是否有读权限  **-w** 判断文件是否存在且是否有写权限  **-x** 判断文件是否存在且是否有执行权限 | 判断文件是否存在  注意中括号里面空格不能省略 |
| **[ 文件1 -nt 文件2 ]** | 判断文件1的修改时间是否比文件2新 |
| **[ 文件1 -ot 文件2 ]** | 判断文件1的修改时间是否比文件2旧 |
| **[ 文件1 -ef 文件2 ]** | 判断文件1和文件2的inode是否一致，可以认为判断两个文件是否为同一文件 |
| **[ 整数1 -eq 整数2 ]**  **-ne 不相等**  **-gt 大于**  **-it 小于**  **-ge 大于等于**  **-le 小于等于** | 判断两个数是否相等 |
| **[ -z 字符串 ]** | 判断字符串是否为空（为空返回真） |
| **[ -n 字符串 ]** | 判断字符串是否为非空 |
| **[ 字符串1==字符串2 ]** | 判断两个字符串是否相等 |
| **[ 字符串1！=字符串2 ]** | 判断两个字符串是否不等 |
| **[ 判断1 -a 判断2 ]** | 两个判断都为真则返回真 |
| **[ 判断1 -o 判断2 ]** | 两个判断至少有一个为真就为真 |
| **[ ! 判断 ]** | 取反，判断为真时返回假 |

**5 流程控制语句**

|  |  |
| --- | --- |
| **if语句**  **if开头，fi结尾** | **①单分支语句：**  if [条件判断式] ; then 动作 fi  或：if [条件判断式]  then  程序  fi  **②多分支语句：**  if [ 判断式 ]  then  条件成立，执行程序1  else  条件不成立，执行程序2  fi  **③多分支语句**  if [ 条件判断式1 ]  then  判断式1成立，执行程序1  elif [ 判断式2 ]  then  判断式2成立，执行程序2  exit 4 (**相对于break，退出判断语句，并且返回4**)  。。。。  else  执行程序n  fi |
| **case语句** | case $变量名 in  “值1”)  如果变量值等于值1，执行程序1  ;;  “值2”)  如果变量值为值2，执行程序2  ;;  其他分支  \*)  如果变量的值没有匹配任何值，则执行本处程序  ;;  esac |
| **for循环** | **语法一**  for 变量in 值1 值2 值3 .....值n （循环n次，第i次将值i赋给变量）  do  程序  done  **语法二**  for((初始值;循环控制条件;变量变化))  do  程序  done |
| **while循环与until循环** | while [ 条件判断式 ]  do  程序(包含条件的改变，防止死循环)  done  untill [ 条件判断式 ]  do  程序  done |

**12 进程管理**

进程管理主要的工作是判断服务器的健康状态，另外就是查看系统的中所有的进程以及杀死进程

**进程状态（STAT）:**R 运行；S 睡眠；T 停止状态；s 包含子进程；+ 位于后台

|  |  |
| --- | --- |
| **命令** | **功能** |
| ps aux | 查看系统中所有进程（使用BSD操作系统格式） |
| ps -le | 查看系统中所有进程（使用Linux操作系统格式） |
| top [选项]  **-d** 秒数 指定top命令每隔几秒更新，默认是3秒  **?或h** 显示交互模式的帮助  **P** 以CPU使用率排序，默认项  **M** 以内存使用率排序  **N** 以PID排序  **q** 退出top | 查看系统健康状态 |
| **pstree** [选项]  **-p** 显示进程pid  **-u** 显示进程所属用户 | 查看进程树 |
| **kill -l** | 查看可用的进程信号 |
| **kill pid** | 正常终止 |
| **kill -1 pid** | 重启 |
| **kill -9 pid** | 强制终止 |
| **kill -15 pid** | 正常终止（默认） |
| **killall [选项][信号]进程名**  **选项：**  **-i** 交互式，询问是否杀死某个进程  **-I** 忽略进程名大小 | 按照进程名杀死进程 |
| **pkill [选项][信号] 进程名**  **选项：**  **-t 终端号** 按照终端号踢出用户 | 按照进程名终止进程 |
| **命令 &（命令仍然执行）**  **或 按ctrl+z放入后台（命令停止执行）** | 把进程放入后台  如：tar -zcf etc.tar.gz /etc & 将打包进程放入后台 |
| **vmstat [刷新延时 刷新次数]** | 监控系统资源  如：vmstat 1 3 |
| **dmesg** | 开机时内核检测信息 |
| **free [-b|-k|-m|-g]**  **-b** 以字节数为单位显示  **-k** 以KB为单位显示（默认）  **-m** 以MB为单位显示  **-g** 以GB为单位显示 | 查看内存使用状态 |
| **cat /proc/cpuinfo** | 查看CPU信息 |
| **uptime** | 显示系统启动的时间和平均负载，类似于w命令 |
| **uname [选项]**  **-a** 查看系统所有相关信息  **-r** 查看内核版本  **-s** 查看内核名称 | 查看内核相关信息 |

**crond服务管理与访问控制**

|  |  |
| --- | --- |
| **命令** | **作用** |
| **service crond restart** | 启动crond服务 |
| **crontab [选项]**  **-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 | **通过crontab -e 加入vim的编辑界面**  **定时任务编写格式为：**  **\* \* \* \* \* 任务** |

**日志**

**日志服务：rsyslogd**

|  |  |
| --- | --- |
| **命令** | **功能** |
| **vim /etc/inittab** | **查看默认运行级别** |

**其他：**

|  |  |
| --- | --- |
| **uniq** [选项] 文本  选项：  **-c** 去除重复行并且显示重复次数 | 去除重复行 |
|  |  |

**宏的录制**

按q+宏名称开始录制宏，再一次按q结束录制

按@宏名称使用宏

**如果使用rm -f 不小心删除了某个文件，该如何恢复？**

使用rm删除文件时，只是删除了文件的目录索引节点并没有删除文件描述符，此时对于文件系统不可见，但是对于打开它的进程仍然看见，可以尝试通过文件描述符来重新读取文件，然后使用重定向来恢复文件。步骤：使用ps -aux查看该文件的进程，使用cat /proc/进程号/fd/文件描述符查看恢复数据，使用i/o重定向来恢复文件。