挂载点

根目录 / 第一块硬盘根驱动器 包含虚拟目录

其他驱动器挂载在目录下

登录，主目录:分配给用户账户的目录

Pwd 当前工作目录 cd 目录 切换当前工作目录

绝对路径相对路径 / 从根目录开始 跳回主目录 cd

相对路径 基于工作目录 cd子目录 .当前目录 ..父目录 cd ../

列表ls -F目录名加上斜线 隐藏文件 -a 点号开头 -R 子目录里的文件也显示

显示附加信息 –l 总文件数 类型 权限 硬链接数 所属用户名 组名 大小 修改时间 文件名

过滤器 名字 ？一个字符 \*一个或多个 通配符 [] 多选一 !不需要

文件操作 创建空文件touch 文件名 改变已有文件修改时间touch 改变访问时间touch –a

复制文件 cp 源 新文件目录/名字 -i 提醒覆盖已存在文件 cp . 目录尾部应该增加/ 不然可能创建一个新文件而不是新目录

复制整个目录 cp –R 目录/ 新目录名 ls-Fd 只显示目录不显示内容 cp通配

自动补全文件名 tab键

链接文件 符号链接 指向另一个文件 ln –s 被链文件名 链接文件名

文件 inode编号，每个文件的唯一编号 ls –i 链接文件很小

硬链接 实际上是同一个文件 ls-l 中可看到链接数

重命名使用mov 指令 重命名相当于覆盖吧 mv 旧文件 新文件

移动 mv 文件 新目录/新文件名 -i 是否覆盖 没有改变文件inode编号

删除文件remove rm –i 提示是否删除 删除后无法找回，没有回收站

通配符删除 强制删除 –f

创建目录 makedir mkdir 目录名 一次性将各级子目录创建出来 mkdir /1/2/3/4

删除目录 rmdir 默认只能删除空目录 应该先删除文件再删除目录

删除目录和文件 rm –r /目录 -ri有提示 -rf 没有提示

查看文件 不用编辑器

类型file 编码 文本 链接 脚本 可执行

内容 cat 文本内容 -n带上行号 只文本行行号 –b 替换制表符 –T

More 每页会停下来，一次只能一页 q退出 less 可以上下翻页

部分文件 tail 文件尾 默认10行 -n 行数 -f**实时查看**

Head

进程检测 process ps pid tty终端 cpu时间 –A所有进程 用户组进程 cmdlist grplist userlist pidlist sesslist

显示线程 –L 长列表 –l uid pid ppid父进程 c cpu利用率 -f cmd 程序名扩展输出或者 –l

Stime 启动时间

使用 –l 后 内存地址addr s进程状态 运行休眠等待运行，停止 已结束 交换空间sz

2、实时检测 top 负载 时间 用户数 进程task q退出

结束进程 进程间用信号通信 信号 挂起 中断 结束运行 无条件终止 段错误 停止/暂停 恢复

Kill pid 必须是进程的属主 或者root killall 进程名 可以用通配符

检测磁盘空间

把磁盘挂载到目录下 mount 输出设备列表 设备文件名 挂载目录 文件系统类型 访问状态

手动挂载 mount –t 文件系统类型 设备 目录 vfat ntfs iso9660

卸载 unmount 目录或者设备 程序正在使用无法卸载

磁盘使用情况 df -h转换成标准格式， 尺寸用m和g

显示文件的情况 当前目录下所有文件的磁盘情况 du -c总大小 -h易读格式 kb –s参数总计

数据处理 文本文件排序 sort 文件名 默认是当成字符串 -n识别成数字 -M按月排序 时间戳

-k –t

搜索 正则 grep 模式 文件名 反向搜索 –v 显示行号 –n 知道匹配的行数 –c 多个模式-e –e

压缩数据 zip gzip bzip2 gzip 文件 通配符

归档数据 tar –a已有tar增加到另一个tar

创建新tar -c

追加文件到末尾 -r

列出tar内容 -t

删除 -d

提取文件 –x

其他选项: 输出结果 显示文件 用zip压缩 –z 用bzip2压缩 结果保存到文件f 文件名

Shell类型 默认的hsell程序 gun bash shell /bin/bash

启动shell /bin/bash dash

退出shell exit

父子shell 输入/bin/bash ps –f 看ppid

进程列表 连续命令 pwd;ls ;cd etc/;pwd 用分号分开 进程列表用括号包起来() 会生成子shell执行

嵌套进程列表((( ) ))

后台模式 sleep 10 置入后台模式 sleep 10& ps-f看到 后台进程

Jobs 后台作业信息 []中有作业号

进程列表放入后台模式()&

协程 coproc 命令 （） jobs 可以自己设置协程进程名

内建命令 外部命令 ps 找命令位置 which ps type ps 外部命令是程序，需要创建子进程

内建命令是shell子带的 type exit cd 有的命令有外部也有内建 type –a

History 输入命令的历史 使用最近的命令 !! 历史记录写在文件中，退出时保存 –a强制写入

命令别名 alias 查看别名 alias –p 比如 ll 就是ls –alF

六、环境变量 系统信息 存储临时数据 和配置信息 环境变量在内存存数据，方便访问

全局 env printenv printenv变量名 HOME echo $环境变量名

使用环境变量 可以用$ 符号 ls $HOME

局部环境变量只在进程可见 set

直接在shell中设置 变量 变量名=变量值 echo $变量名 记住变量名值没有空格，字符串用””括起来

局部导出到全局 export 变量名

删除环境变量 unset 变量名

默认环境变量 PATH HOME BASH BASHPID 等等

外部命令必须加入到path路径 不然就只能输入自己的全路径执行，目录用;分隔

Shell启动文件 .profile

环境变量持有化

数组变量 arr=(1 2 3 4 5) 数组用空格分隔 使用索引访问 echo $arr[]

显示整个数组 ech $arr[\*] 修改值 删除值unset

文件权限

/etc/passwd root账户 uid 0 500以下是系统账户 500以上用户账户

账户 密码 uid gid 备注 home目录 默认shell

/etc/shadow 保护账号密码 需要root 加密后密码 修改天数

添加新用户useradd -D按照默认值添加 默认不创建HOME目录 –m创建

--r创建系统账户 -p密码

删除用户 userdel 删除用户信息 passwd 但不删除文件 -r删除home文件

修改用户

Usermod 可以修改任何一处 –l修改登录名 L锁定 U解锁 p修改密码

passwd修改密码

chpasswd

密码过期实践

Chsh chfn chage 设置过期时间

Linux组

多个用户对文件目录 共享权限 /etc/group 组名 组密码 gid 用户列表

Usermod –G 组名 用户名 将用户添到组 创建组groupadd 组名

修改组 groupmod -g 修改gid -n 修改组名

文件权限

第一个字段，对象类型

文件 d 目录 l链接 c字符型设备（打印机） b块设备 n网络设备

三组访问权限 r读 w写 x执行 三组分别是 属主，属组 其他用户

默认文件权限

Umask 设置默认权限 显示和设置 权限码 000 -111 0-7

0无权 1执行 2写入 3写入执行 4读取 5读取执行 6读写 7全部

权限码共三组

666 777

Umask的值 全权限-umask =默认权限 如果umask=26 默认目录=777-26=751

改变权限 chmod 八进制码 文件

改变文件属主chown 新属主（uid或名字） 文件 同时改变属组 chown 主.组 文件

修改属组 chgrp 组名 文件

**共享文件:创建组 或者修改其他用户的访问权限**

每个文件单独设置 特殊权限 suid sgid 使一个目录下都属于目录属组

文件系统

Ext文件系统 虚拟目录操作硬件 定长块 索引节点表 每个文件在节点表中

文件名 大小 属主 属组 访问权限 指向硬盘块的指针 使用索引节点号引用，创建文件时分配

文件不能超过2GB

Ext2 索引节点表 创建时间修改时间 最后访问时间 32TB 碎片化 按组分配磁盘块 读取时按组找数据，不需要找整个磁盘

日志文件系统 文件修改写入临时文件（日志） 成功写入设备和索引后才删除日志，断电会读取日志

数据模式 文件写入日志

有序模式 只写索引节点 数据写入成功才删除

回写模式

Ext3

Ext4

resiserFS

JSF xfs

写时复制文件系统 cow 不会覆盖当前数据 放入另一个位置，修改完也不重写

ZFS

创建分区 fdisk 必须指定一个存储设备 P显示存储设备详细信息 n 创建分区

主分区 扩展分区（容纳主分区） l可用分区类型 w保存分区更改

分区创建文件系统 mkefs mkfs.ext3 type命令检查是否安装该文件系统

Mount 挂载分区到虚拟目录

文件系统检查和修复 fsck

逻辑卷管理 lvm 物理卷（相当于分区） 卷组 逻辑卷

安装软件

包管理

基于debian apt-get apt-cache aptitude(避免依赖缺少)

包列表 dpkg –L 包名

Search install remove

Etc/

编辑器

Vim alias vi which vim 创建/打开文件 vi 目录/文件名

操作模式 普通 插入 默认普通模式（输入的是命令） 退出插入模式esc

移动光标 方向键就好 可以用hjkl 快速移动 pagedown 最后一行 G 第一行gg

第n行 n G

冒号 底栏模式 q未修改退出 q!取消修改退出 w保存 wq 保存退出

删除 x 最好不用退格 删除行dd 删除单词 dw 删到行尾ds 撤销u 光标后追加数据a

行尾追加数据A

复制粘贴 y 复制字符 yw复制单词 y$复制到行尾 p粘贴 删除的数据也可以粘贴（剪切）

查找替换 /文本 n键继续查找

替换 :s/旧/新/ 替换所有 /g

emacs

脚本显示输出 echo

查找 find –name 文件名

十一、脚本编程

Date;who 日期时间 登录者

创建文件 文件头 #! /bin/bash 使用的shell

然后在每行写命令 可以用分号写一行

注释# 除了#！之外都不解释

保存文件，文件夹要加入到path环境变量，或者用绝对路径，需要执行权限chmod

3、显示消息 echo 和命令显示在同一行 echo –n “内容”;命令

4、变量 环境变量 set显示环境变量列表 $使用 echo “$”照样会翻译，需要转义

用户变量 名=值 写在脚本中 类型自动的 同样 是$引用

赋值的话 值也要用美元 a=$b 不用的话会解释成字符串

命令赋值给变量 a=$(命令) $(命令+表达式)

重定向输入输出 命令输出发送到文件 命令>文件 覆盖 追加>>

输入重定向 文件定向给命令 命令<文件 wc文本计数

内联输入重定向 << 标记 （eof） 数据 标记

管道 命令的输出作为另一个命令的输入 sort 命令1|命令2 ，可以配合重定向和管道

数学运算 expr 1+1 更简单$[表达式] a=$[] 仅仅整数

Bc 计算器 内联输入重定向

退出脚本 退出状态码 $? 成功 0 错误正数 未知命令 不可执行

Exit 状态码 指定状态码

结构化命令 if 命令 then 命令 fi 如果 前一个命令成功执行下个命令

If then else fi

定义条件 test 条件 if test 条件 then 缩写为if [条件] then

test后面可以是变量等 比较大小相等

比较字符串

比较文件 是否存在-e 是目录 文件 可读 可写 可执行 属于用户 创建时间

复合条件 []&&[] ||

数学表达式 (( )) ++ -- ！ \*\* >>

循环 遍历 for读取列表 读取重复 命令输出 更改文件分隔符 通配符读目录

循环处理文件数据 while until IFS环境变量 每行处理 break continue

查找可执行文件 创建多个账户

循环输出 到管道 或者重定向

和用户交互，**处理输入**

命令行参数

位置参数 $0程序名（脚本名） $1第1个命令行参数 $2 大于9需要${10}

运行时输入参数 脚本 参数

特殊变量 $#参数个数 $\* $@所有参数 for遍历

移动参数 shift 依次处理

选项 - 查找

用户输入 read 变量名 输入会被存入变量 超时 –t 等待秒数 –s隐藏用户输入

从文件读取 每次一行

文件描述符 标识打开的文件 9个 0 1 2 标准输入（键盘） 标准输出（终端） 标准错误（默认也是终端）

重定向时 < 用指定文件取代标准输入

错误重定向 让错误不出现在屏幕上 命令 描述符>输出文件test 2> file

2> 1> 临时重定向 永久重定向 exec 1>文件 之后echo全部都发到文件

创建自己的重定向 **文件描述符**

列举打开的 描述符 lsof

临时文件 启动自动删除

脚本控制

发送信号 挂起 停止 终止 无条件终止 继续运行 sigint

生成信号 ctrl+c 停止运行的进程 暂停进程 ctrl+z ps

捕获信号 陷阱 trap 命令 信号

捕获退出 trap 命令 EXIT 修改移除捕获

后台运行 命令& jobs [作业号]pid 后台仍然会输出标准输出

多个后台作业 同时使用

非控制台运行脚本 nohup 命令& 阻止退出信号 退出终端后不能标准输出，nohup.out

作业控制

Jobs $$脚本pid 查看作业pid jobs –l 减号 下一个默认作业

重启作业bg 作业号 kill

调整优先级（谦让度） -20 +19 nice –n 优先级 命令

Renice –n 10 –p pid

定时作业 at

定期脚本 corn

函数 function 函数名 {命令} 或者 函数名（）{}

使用 函数名 函数是小型脚本，有退出状态码 return 状态码只能是0-255

如果想输出一个值可以输出到变量中

传参 使用命令行参数 $# $0

变量作用域 默认都是全局 局部的话需要 local

数组的传入和返回

递归

文本处理sed gawk 处理数据流 正则

管道连接sed echo “文本” | sed

一次一行 匹配数据 命令流 sed 编辑器命令

替换 s/ 原/新/ ***不会修改原来文件***

多个命令 sed -e “命令1;命令2” 文件 或者`` 可换行

命令也可以用文件 –f

Gawk 属于编程语言，适合修改文本

Gawk ‘{命令 print $1}’ 文件名 多个命令用分号分隔

字段 $0 $1 字段分隔符 空格等 –F

Sed 替换标记 s/旧/新/标记 g 所有匹配 n数字，替换第几处 w替换写入文件 p打印原来行 转义字符 \/

默认sed应用于所有行 行寻址 行号 2，3s/旧/新/ 2，3 行 2行到末尾 2，$

文本模式过滤 /模式/s/旧/新/ 行上多条命令 行号{命令;命令}

删除行 d 3d 2,3d 2,$d 模式匹配

插入行 2i 增加新行到前面 3i/插入文本

附加行 增加到后面 3a/附加文本内容

修改行 3c/新文本

转换 处理单字符 1y\abc\edf 匹配的是单字符，不是串 所有的a会变e

打印 2，3p文本行 打印行号 2= 列出行

写入文件 文件写入行 “3w 文件名” 另一个文件 读取2，3r

正则表达式 通配符 文本 特殊字符转义 锚字符 首尾

Sed是单行 next 多行删除 N D P