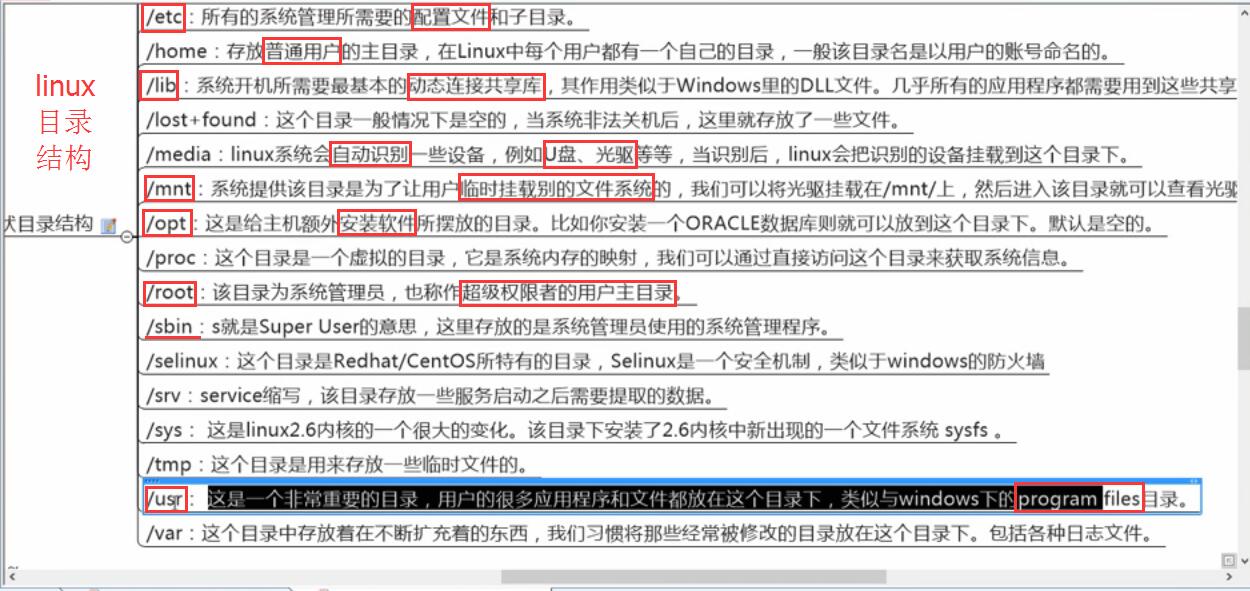
1、Hadoop介绍：

解决两大问题：1.大数据存储

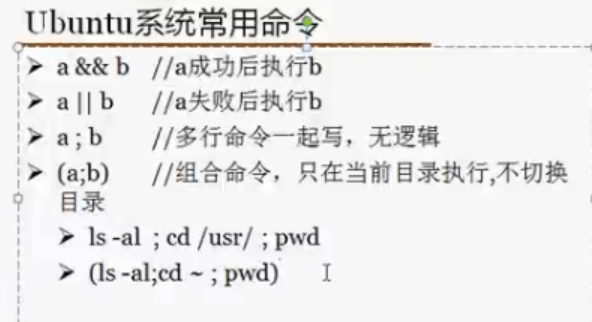
2.大数据运算

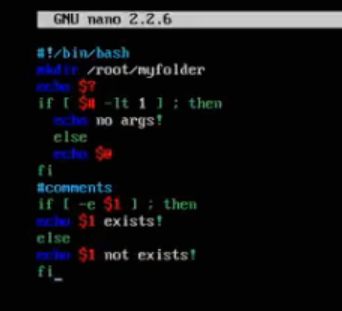
1. 安装虚拟机软件VMWare
2. Ubuntu
3. Linux目录结构

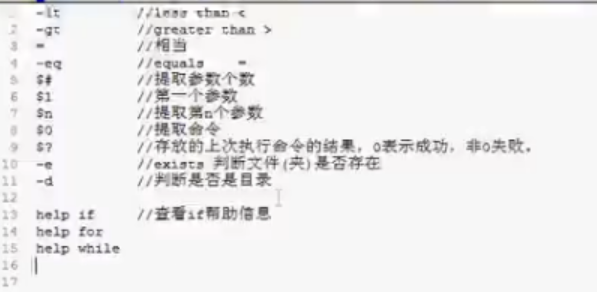
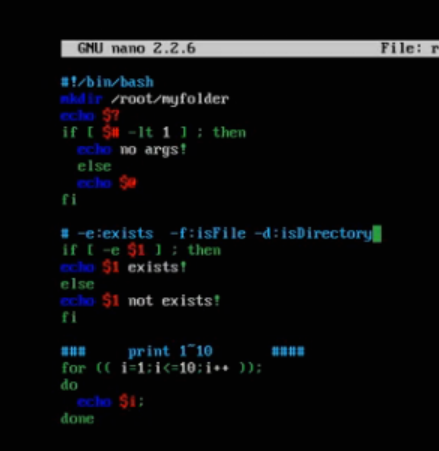


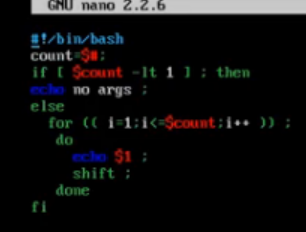
1. Ubuntu基本命令：

|  |  |  |
| --- | --- | --- |
| 操作 | Linux | Windows |
| 进入目录 | cd | cd |
| 在VMWare下安装列出当前目录内容 | ls 列出全部  ls-l 列表方式  ls-al 包含显示开头  ls-alh 可读的人性化的 | dir |
| 显示当前目录在哪里 | pwd |  |
| 清屏 | clear | cls |
| 进入当前用户主目录 | cd ~ |  |
| 动态提取当前用户名（脚本编程时使用） | whoami |  |
| 修改root根用户密码 | sudo passwd |  |
| 切换光标（光标就能显示了） | Alt+Ctrl |  |
| 查看ip | Ifconfig | Ipconfig |
| 创建目录 | mkdir | mkdir |
| 当前目录 | . |  |
| 进入上级目录 | cd .. |  |
| 删除目录 | rmdir download/ |  |
| 删除目录（循环强制删除） | rm -r -f（-f强制-r递归） |  |
| 创建文件 | touch a.txt |  |
| 拷贝 | cp a.txt b.txt  cp a.txt ~/downloads(目录不存在不能拷贝需要先创建) |  |
| 移动（剪切） | mv a.txt ../ |  |
| 重命名 | mv a.txt aa.txt |  |
| 输出一段文本 | echo | echo |
| 查看文件内容 | cat(例：cat index.html)  cat -n a.txt 带行号输出内容 | type |
| 重定向（追加） | >>(例：echo aaa >>a.txt  将aaa写入a.txt) | >>> |
| 内容覆盖（覆盖） | >（例：echo bbb>a.txt  内容bbb覆盖了aaa） | >> |
| 显示帮助文档 | man |  |
| 分页查看（一屏一屏显示） | man ls |more按d/回车翻下页 |  |
| 管道（前边信息输出作为后边信息输入） | xxx | yyy  ls-al | grep hostname 显示当前文件夹在名为hostname的文件（grep意思是过滤）  ls -al |grep a.txt 显示a.txt信息  ls -al |grep d 显示所有文件夹  ls -al |grep 4096 显示大小为4096的文件  ls -al |grep ubuntu 显示文件中包含字符Ubuntu的文件  ls |grep pig |cp `xargs` ~ /download  将当前文件夹中包含pig字符串的所有文件拷贝到download文件夹中 |  |
| 分页查看 | more |  |
| 显示前十条信息 | man ls |head -10  或者head a.txt(默认10行)  head -5 a.txt 显示a内容前五行  head -n 5 a.txt 显示a内容前五行  head -c 5 a.txt 显示前三字节 |  |
| 显示后十条信息 | man ls |tail -10  tail -n 5 a.txt 显示a内容后五行  tail -c 5 a.txt 显示后三字节 |  |
| 下载网页 | wget url（wget <http://www.baidu.com>） |  |
| 以管理员身份重启系统 | sudo reboot |  |
| 创建目录树 | mkdir -p a/b/c |  |
| 查看整个目录中的所有文件及子目录（递归） | find . |  |
| 以树形方式展现目录结构 | tree |  |
| 打开新的会话（虚拟机死了的时候可以用） | Ctrl + Alt + F5  （F123456都行） |  |
| 命令行编辑器 | nano  nano a.txt  底部有帮助快捷键  Ctrl+O 写入输出即保存  Ctrl+K 剪切一行  Ctrl+U 恢复一行  Ctrl+X 退出  Home 定位光标行首  End 定位光标行尾  Pgup 上翻  Pgdn 下翻 |  |
| 查看主机名 | hostname |  |
| 环境变量  （Windows不区分大小写Linux区分大小写） | echo $PATH  echo ${PATH}  echo “$ {PATH}” |  |
| 查看命令所在目录（在path下） | which xxx  which ifconfig |  |
| 查询命令xxx的binary、source、manual文件 | whereis xxx  whereis ping |  |
|  | source |  |
| 查看Linux名称 | uname -a  或uname --all |  |
| 查看文件信息包括文件类型等 | file a.txt |  |
| 归档文件 | tar -x extract 抽取归档文件  tar -v verbose 详细信息  tar -z filter 过滤  tar -f 指定文件 |  |
| 压缩/解压 | gzip  gzip -d 解压  gzip -1 速度优先压缩(压得越快越好)  gzip -9 质量优先压缩(压得越小越好) |  |
| 解压文件 | gunzip -c -l -f |  |
| 查找 | find. | xargs grep -ri “ibm”  在当前文件夹查找含ibm字符串的文件并且将结果用一行输出（xargs意思是将结果返回为一行；-ri是含xx字符串） |  |
| 指令套指令 | ` `用反引号 echo `hostname`  或者用 $( ) echo $(hostname)  可以嵌套 $(xxx $(xxx)) |  |
| 挂载 | mount  sudo mount /dev/cdrom /mnt/cdrom |  |
| 卸载解除挂载 | umount  sudo umount /dev/cdrom  或者sudo umount /mnt/cdrom |  |
| 连接 | ln //创建连接 link -s:符号连接  //默认是硬连接  ln target link\_m=name  //硬连接，相当于实时镜像，删除文件时不会删除连接文件  //硬连接不允许对目录创建  ln -s target link\_name  //符号连接，相当于快捷方式（指向某文件的地址）  //可以对文件和目录创建 |  |
| 备份 | cp xxx xxx.cak //文件xxx备份 |  |
| 进入物理地址 | cd -p xxx //进入xxx所在的物理地址  pwd -p xxx //进入xxx所在的物理地址 |  |
| 启动一个进程在后台运行 | xxx & eclipse & |  |
| 作业 | jobs |  |
| 杀死作业 | kill %n kill加作业编号 |  |
| 后台程序 | bg |  |
| 前台程序 | fg |  |
| 显示进程信息（任务管理器） | ps -Af -A:all == -e  -f: full output fomat  -o:用户自定义显示哪些列  ps -Ao uid pid ppid(注意是小写) |  |
| 切断内容 | cut -c N-M  -c : 按照字符数计算  N-M:从第n个字符到第m个字符  n- ：从第n个到行尾  -m ：从行首到第m个  n :第n个字符 |  |
| 显示Linux分区内部定义的命令 | help |  |
| 帮助 | xxx --help  man xxx |  |
| 分区 | fdisk  sudo fdisk -l 查看分区表  sudo fdisk /dev/sda 管理分区 |  |
|  | df diskfree |  |
|  | dd |  |
| 提取文件名 | basename /bin/ping 结果为：ping |  |
| 提取目录名 | dirname /bin/ping 结果为：/bin |  |
| 后退 | cd - |  |
| 后退n步骤 | cd -n |  |
| 导入环境变量(可用三目运算符) | export xxx=/bin  export xxx ${xxx : -$ xxx2}  export myvar ${myvar : -$myvar1}  //导入一个环境变量myvar，如果存在myvar就取myvar，不存在就取myvar1 |  |
| 添加环境变量 | export PATH=${PATH}:.  把当前目录添加到PATH环境变量中，注意要把原本的环境变量加上，用冒号隔开 |  |
| 查看所有环境变量 | env |  |
|  | $env other //空格区分  ${env} //{ }界限  “$env” //” “界限  echo ‘${env}’ //原样输出  echo `pwd` //按脚本执行  export env=${env:-$env2} //是取env，否取env2 |  |
| 修改权限 | chmod  chmod u+x //给user增加执行权限  chmod ug+x//给user和group增加执行权限  chmod ugo+x//给owner、user和owner增加执行权限。相当于chmod a+x  chmod u+rw //给u加读写权限  chmod ug-rw //给ug减少读写权限  chmod 777 a.txt //纯数字操作所有权限 |  |
| 权限拥有者修改 | chown -R ubuntu：ubuntu xxx  sudo chown -R root：root a.txt |  |
| 添加用户 | adduser --home /home/ubuntu2  //添加用户xxx addgroup  --uid 3000  ubuntu2 |  |
| 添加用户 | useradd //是OS底层使用的添加用户指令，推荐使用  adduser命令 |  |
| 删除用户 | deluser //删除用户  sudo deluser --remove-home ubuntu2 //删除  用户同时删除目录  sudo deluser --remove-home --remove-all-files ubuntu2  //删除用户同时删除home目录 |  |
| 提取第几个参数 | if[“$1”=”--conf”]  run.sh------>内容：hello $0 $1 $2!  ./run.sh aaa bbb----->输出：hello ./run.sh aaa bbb! |  |
| if | -lt //less than 即<  -gt //greater than 即>  = //相等  -eq //equals 即=  $# //提取参数个数  $1 //第一个参数  $n //提取第n个参数  $0 //提取命令  $? //存放的是上次执行命令的结果,0:成功,非0：失败  $@ //输出所有参数  -e //exists 判断文件（夹）是否存在  -d //判断是否是目录  help if //查看if帮助信息  help for //for帮助  help while //while帮助 |  |
| if例子  （中括号开始和结尾一定要带**空格**） | if[ $# -gt 1 ] //if参数个数>1  if[ -e xxx ] //xxx是否存在  if[ -d xxx ] //xxx是否是目录  if [ $# -lt 1 ];then  ……  else  ……  fi |  |
| for循环 | for (( xxx;xxx;xxx )) ;  do  ……;  done  例：//输出1~10  for (( i=1;i<=10;i++ )) ;  do  echo $i;  done |  |
| if/for例子 | //循环输出所有参数  count=$# ;  if( $count -lt 1 ) ;then  echo no args ;  else  for(( i=1;i<=count;i++ )) ;  do  echo $1 ;  shift ;  done  fi |  |
| 左移 | shift //往左窜一个 |  |
| 常用命令 | a && b //a成功后执行b  a || b //a失败后执行b  a ; b //多行命令一起写，无逻辑  (a;b) //组合命令，只在当前目录执行，不切换目录  例：//多行命令一起写  ls -al ;cd/user/ ; pwd  (ls -al;cd ~ ; pwd) |  |









文件类型：

（1）文件夹：d

（2）链接：l

（3）文件：-



用户类型：

1. owner
2. group
3. other

权限：

1. r 读
2. w 写
3. x 执行

例： d r w x r - x r - -

二进制： 1 1 1 1 0 1 1 0 0

值为7 5 4

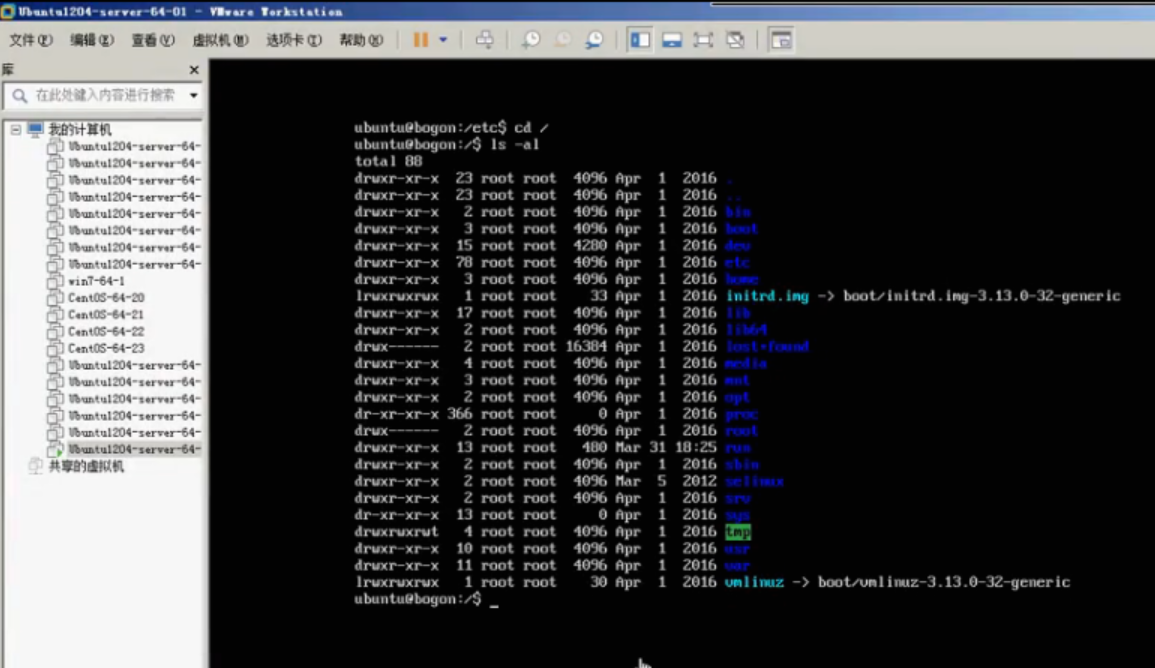
某一文件夹的权限是：owner用户可以读写和执行

和owner一组的用户可以读和执行不能写

其他用户只能读不能写和执行

如上图

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 文件夹类型 | root权限 | 跟root一组的用户权限 | 其他用户权限 | 当前是root用户 | 当前是root组 | 字节数 | 时间 | 文件夹名称 |
| d | rwx | r-x | r-x | root | root | 4096 | Apr 1 2016 | download |



在Linux中：

--------------------------------

/sbin/bin //鼻祖、创始者存放可执行文件目录

/user/bin //各个Linux产品（如Ubuntu、centOS、Redhat）

/local/bin //用户本身存放目录



在VMWare的client机光驱上插入光盘（iso镜像文件）

--------------------------------------------------------------------------------

1.显示VMWare状态栏（右下角）

2.在状态栏光驱右键--->设置--->打开窗口

3.硬件选项卡--->cd/dvd

4.右侧选择连接--->iso镜像文件--->浏览选中iso文件

5.右键光驱--->连接

6.cd/dev

ls -al |grep cd //查看所有cd连接（它们都指向sr0，sr0是一个块文件）

7.创建 /mnt/cdrom //以root身份创建，用来容纳挂载文件

8.sodo mount /dev/cdrom /mnt/cdrom //挂载cdrom光驱到/mnt/cdrom下

9.sudo umount /dev/cdrom //卸载设备

10.sudo umount /mnt/cdrom //卸载挂载目录也可以

命令：

-----------------------------------------------

ln //创建连接 link -s:符号连接 默认是硬连接

ln target link\_m=name //硬连接，相当于实时镜像，删除文件时不会删除连接文件

//硬连接不允许对目录创建

ln -s target link\_name //符号连接，相当于快捷方式（指向某文件的地址）

//可以对文件和目录创建

cp xxx xxx.cak //文件xxx备份

cd -p xxx //进入xxx所在的物理地址

pwd -p xxx //进入xxx所在的物理地址

------------------------------------------------------------

运行脚本文件a.sh方法

不能直接输入a.sh，会被默认为命令执行，提醒找不到此命令

有两种解决办法-------->1.在a.sh前面带上绝对路径 ./a.sh

-------->2.给a.sh文件所在的目录配环境变量 export PATH=${PATH}:.

--------------------------------------------------------------

文件夹权限rwx //r：读取文件夹内部的文件列表

//w：添加/删除文件

//x：是否能够进入

------------------------------------------------------------

user 只有一个primary group，多个secondary group

输出所有脚本参数，一行输出 输出1~9

------------------------------------------------------------

[run.sh]

#！/bin/bash

if [ $# -lt 1 ] ;then

echo no args! ;

else

echo $@ ;

fi

for( i=1;i<10;i++ ) ;

do

echo $1 ;

done

输出所有脚本参数，每个一行 输出1~9

------------------------------------------------------------

[test.sh]

#！/bin/bash

count=$# ;

if [ $count -lt 1 ] ;then

echo no args! ;

else

for( i=1;i<=$count;i++ ) ;

do

echo $1 ;

shift ;

done

fi

在命令行终端测试 -e -f -d

-----------------------------------------------------------

$>[ -e /xx/xx ] //-d 是否文件夹 -f 是否文件 -e 是否存在

$>echo $? // $? 是查看上次执行命令是否成功，0成功，1失败

NetCat

-----------------------------------------------------------

瑞士军刀

模拟TCP、UDP网络通信

实现端口扫描

语法：

服务端 nc -l port //指定监听的端口号

客户端 nc ip port //指定服务器的ip和监听端口号

例：nc -l 1234 & //启动服务器，监听端口1234，&表示后台运行

jobs //查看后台运行的作业

kill %n //杀死指定编号的作业

netstat -lnuppt //查看端口号和任务等情况

netstat -ano //查看详细情况