Xxxxxx

Linux目录树型结构：

高级命令：

awk：文本处理命令

grep： 文本查找

sed： 文本替换

Linux 应用的情景：

1. 搭建服务器的（虚拟机的安装、IP配置DNS访问网络、防火墙的配置、tomcat服务器的配置、JDK配置、MYSQL数据库服务器、远程连接SSH、SFTP文件传输）
2. 常用的shell操作命令：ls （list 列出当前所有的目录和文件） cp (copy 复制) mv （move 剪切） tar（打包解压命令 ） rm （remove 移除） chmod （改变文件的权限 777 r4 w2 x1） mkdir （make direction 新建文件夹） rmdir 、 pwd （打印当前的路径）、ip addr （查看IP地址）、dh –f (查看存储使用率)、cd （change direactory 相对路径、绝对路由 / .. 回到上一级目录 . 表示当前目录） 、vi （文件编辑 i在当前位置之前插入 o 表示在当前位置下一行开始插入 :wq 保存退出 :q! 不存盘退出）、SSH （ssh root@IP 远程登陆） 、cat （查看文本信息）、 >>定向到文本到尾部追加 、>覆盖 、SCP （远程copy）、tar 打包、解包
3. 搭建mysql数据库服务器(远程连接)
4. Shell脚本开发:IFS 分届符 Shell操作mysql Shell读取文件 Shell数组遍历 Shell 管道符 | Shell 权限修改 chmod 755 SCP 远程拷贝 Shell参数设置 （练手：语法、变量定义、 表达式、数组、控制选择循环、函数、参数化）

相对路径： . 当前 .. 上一级 ../.. 上一级的上一级

绝对路径： / 从根开始起的完整的路径

常用命令：

ls: 列出目录

cd：切换目录

pwd：显示目前的目录

mkdir：创建一个新的目录

rmdir：删除一个空的目录

cp: 复制文件或目录

rm: 移除文件或目录

mv: 移动文件与目录，或修改文件与目录的名称

Chomd

u+x user添加可执行的权限

g-w group与该用户在用一个组的组员不能写

o+w other其他用户添加可写的权限

按权值划分：r-4 w-2 x-1

drwxrwxrwx :

第一组rwx对文件拥有者的操作权限

第二组rwx表示对与拥有者同属于一个组对组员的操作权限

第三组rwx 表示其他人的操作权限

时间命令：date

在shell脚本中经常会需要获取当前日期的地方，linux的系统时间在shell里是可以直接调用系统变量： 获取今天时期---`date +%Y%m%d` 或 `date +%F` 或 $(date +%y%m%d)

eg: date +%Y%m%d

时间域如下：

% H 小时（00..23）

% I 小时（01..12）

% k 小时（0..23）

% l 小时（1..12）

% M 分（00..59）

% p 显示出AM或PM

% r 时间（hh：mm：ss AM或PM），12小时

% s 从1970年1月1日00：00：00到目前经历的秒数

% S 秒（00..59）

% T 时间（24小时制）（hh:mm:ss）

% X 显示时间的格式（％H:％M:％S）

% Z 时区 日期域

% a 星期几的简称（ Sun..Sat）

% A 星期几的全称（ Sunday..Saturday）

% b 月的简称（Jan..Dec）

% B 月的全称（January..December）

% c 日期和时间（ Mon Nov 8 14：12：46 CST 1999）

% d 一个月的第几天（01..31）

% D 日期（mm／dd／yy）

% h 和%b选项相同

% j 一年的第几天（001..366）

% m 月（01..12）

% w 一个星期的第几天（0代表星期天）

% W 一年的第几个星期（00..53，星期一为第一天）

% x 显示日期的格式（mm/dd/yy）

% y 年的最后两个数字（ 1999则是99）

% Y 年（例如：1970，1996等）

定时调度命令：crontab

3．命令参数：

-u user：用来设定某个用户的crontab服务，例如，“-u ixdba”表示设定ixdba用户的crontab服务，此参数一般有root用户来运行。

file：file是命令文件的名字,表示将file做为crontab的任务列表文件并载入crontab。如果在命令行中没有指定这个文件，crontab命令将接受标准输入（键盘）上键入的命令，并将它们载入crontab。

-e：编辑某个用户的crontab文件内容。如果不指定用户，则表示编辑当前用户的crontab文件。

-l：显示某个用户的crontab文件内容，如果不指定用户，则表示显示当前用户的crontab文件内容。

-r：从/var/spool/cron目录中删除某个用户的crontab文件，如果不指定用户，则默认删除当前用户的crontab文件。

-i：在删除用户的crontab文件时给确认提示。

minute hour day month week command

# For details see man 4 crontabs

# Example of job definition:

.---------------------------------- minute (0 - 59) 表示分钟

| .------------------------------- hour (0 - 23) 表示小时

| | .---------------------------- day of month (1 - 31) 表示日期

| | | .------------------------- month (1 - 12) OR jan,feb,mar,apr ... 表示月份

| | | | .---------------------- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat 表示星期（0 或 7 表示星期天）

| | | | | .------------------- username 以哪个用户来执行

| | | | | | .------ command 要执行的命令，可以是系统命令，也可以是自己编写的脚本文件

| | | | | | |

\* \* \* \* \* user-name command to be executed

格式示例：

格式 说明

\*/1 \* \* \* \* service httpd restart 每一分钟 重启httpd服务

0 \*/1 \* \* \* service httpd restart 每一小时 重启httpd服务

30 21 \* \* \* service httpd restart 每天 21：30 分 重启httpd服务

26 4 1,5,23,28 \* \* service httpd restart 每月的1号，5号 23 号 28 号 的4点26分，重启httpd服务

26 4 1-21 \* \* service httpd restart 每月的1号到21号 的4点26分，重启httpd服务

\*/2 \* \* \* \* service httpd restart 每隔两分钟 执行，偶数分钟 重启httpd服务

1-59/2 \* \* \* \* service httpd restart 每隔两分钟 执行，奇数 重启httpd服务

0 23-7/1 \* \* \* service httpd restart 每天的晚上11点到早上7点 每隔一个小时 重启httpd服务

0,30 18-23 \* \* \* service httpd restart 每天18点到23点 每隔30分钟 重启httpd服务

0-59/30 18-23 \* \* \* service httpd restart 每天18点到23点 每隔30分钟 重启httpd服务

59 1 1-7 4 \* test 'date +\%w' -eq 0 && /root/a.sh 四月的第一个星期日 01:59 分运行脚本 /root/a.sh ，命令中的 test是判断，%w是数字的星期几

\*表示任何时候都匹配

"a,b,c" 表示a 或者 b 或者c 执行命令

"a-b" 表示a到b 之间 执行命令

"\*/a" 表示每 a分钟(小时等) 执行一次

crontab 不能编辑系统级的 任务

示例代码程序：

exportFromMysql.sh

#!/bin/bash

HOSTNAME="192.168.1.41"

PORT="3306"

USERNAME="root"

PASSWORD="123456"

DBNAME="whdgy"

select\_sql="select \* from teacher"

result=`mysql -h${HOSTNAME} -P${PORT} -u${USERNAME} -p${PASSWORD} ${DBNAME} -Bse "${select\_sql}"`

dump\_data=./data.user.txt

>$dump\_data

echo -e "$result"> $dump\_data

示例程序：

#一行一行到读取文件中的数据，返回在分割，最后连接数据库插入到表中

#!/bin/bash

HOSTNAME="192.168.1.41"

PORT="3306"

USERNAME="root"

PASSWORD="123456"

DBNAME="whdgy"

OLD\_IFS="$IFS"

IFS=","

cat $1 | while read line

do

echo "hello $line"

arr=($line)

for s in ${arr[@]}

do

echo "$s"

done

select\_sql="insert into teacher values ('${arr[0]}','${arr[1]}')"

result=`mysql -h${HOSTNAME} -P${PORT} -u${USERNAME} -p${PASSWORD} ${DBNAME} -Bse "${select\_sql}"`

done

IFS="$OLD\_IFS"

AWK 学习：

1、掌握：

$0 打印所有信息

$1~$n 打印第n列的信息

FS 每一列的分隔符

RS 每一行的分割符

OFS 输出的时候列的分隔符

ORS 输出的时候行的分割符

用法：awk –F 指定分隔符 ‘BEGIN {第一行要显示的内容信息}{对每一行信息经常处理的逻辑}END{最后一行要显示的信息}’ 要处理的文本数据源

2、了解基本的正则表达式的语法：+ 表示匹配一个或者多个

3、字符串的匹配、替换、分割、查找

替换

|  |  |
| --- | --- |
| 1  2  3 | awk 'BEGIN{info="this is a test2010test!";gsub(/[0-9]+/,"!",info);print info}' this is a test!test!  在 info 中查找满足正则表达式， /[0-9]+/ 用”!”替换，并且替换后的值，赋值给 info 未给 info 值，默认是$0 |

查找

|  |  |
| --- | --- |
| 1  2 | awk 'BEGIN{info="this is a test2010test!";print index(info,"test")?"ok":"no found";}'  ok #未找到，返回 0 |

匹配查找

|  |  |
| --- | --- |
| 1  2 | awk 'BEGIN{info="this is a test2010test!";print match(info,/[0-9]+/)?"ok":"no found";}'  ok #如果查找到数字则匹配成功返回 ok，否则失败，返回未找到 |

截取

|  |  |
| --- | --- |
| 1  2 | awk 'BEGIN{info="this is a test2010test!";print substr(info,4,10);}'  s is a tes #从第 4 个 字符开始，截取 10 个长度字符串 |

分割

|  |  |
| --- | --- |
| 1  2  3  4 | awk 'BEGIN{info="this is a test";split(info,tA,"");print length(tA);for(k in tA){print k,tA[k];}}' 4  4 test 1 this 2 is 3 a  #分割 info,动态创建数组 tA,awk for …in 循环，是一个无序的循环。 并不是从数组下标  1…n 开始 |