200行

shell脚本

ls cd pwd 1 0

100

pstree 查看进程树

cat /etc/shells 查看系统拥有的解释器

yum -y install ksh 安装新解释器

bash的优点: 历史记录, 快捷键,tab键, 管道, 重定向

标准的脚本构成

1,声明解释器

#!/bin/bash

2,注释 脚本功能, 变量含义 , 每个步骤要解决的问题,等 ...

3,具体代码

脚本执行方式:

1,赋予x执行权限,然后使用绝对或相对路径运行该文件

2,使用解释器直接执行脚本,即使没有x权限也可以

bash test1.sh //新开启bash进程执行test1.sh,脚本执行完毕后,新开启的bash自动退出

3,使用source命令执行脚本 //不开启新解释器,使用当前解释器执行脚本

source test1.sh //source可以简写 .

#!/bin/bash

mkdir abc

cd abc

sleep 1000

编写一键部署yum脚本

#!/bin/bash

rm -rf /etc/yum.repos.d/\*.repo

echo "[abc]

name=test

baseurl=http://172.25.254.254/content/rhel7.0/x86\_64/dvd/

enabled=1

gpgcheck=0" > /etc/yum.repos.d/abc.repo

编写一键部署vsftpd脚本

#!/bin/bash

yum -y install vsftpd &> /dev/null

systemctl start vsftpd

systemctl enable vsftpd

systemctl stop firewalld //脚本写完关闭防火墙测试服务

常量 固定不变的内容

变量 存储可能会发生变化的内容, 增加脚本灵活

变量类型:

1,自定义变量, 变量名称可以使用字母,数字,下划线,不能以数字开头,不能使用特殊符号

a=10 定义变量

unset a 取消变量

echo ${a}RMB 加{ } 分隔变量名与常量

2,环境变量 USER UID HOME(家目录) SHELL(解释器) PATH(命令程序存放路径) PS1(一级提示符) PS2(二级提示符)

echo $USER 查看当前用户名

echo $UID 查看当前用户id号

3,位置变量 $1 $2 $3 ......

4,预定义变量 $0 $$ $? $# $\*

#!/bin/bash

echo $1 脚本后跟的第1个参数

echo $2 脚本后跟的第2个参数

echo $3 脚本后跟的第3个参数

echo $$ 当前脚本的进程号

echo $# 位置变量的个数

echo $\* 所有位置变量

echo $? 上一条指令的结果, 0是正常, 非0是异常

echo $0 执行脚本的名字

echo $! 最后一条放入后台的程序进程号,可以用sleep 1000 &测试

env 查看所有环境变量

set 查看所有变量

利用位置变量编写创建账户配置密码脚本

[root@desktop0 opt]#cat test6.sh

#!/bin/bash

useradd $1

echo $2 |passwd --stdin $1

[root@desktop0 opt]#bash test6.sh xyz 789

"" 界定范围

'' 界定范围,可以屏蔽特殊符号

touch a b //创建2个文件

touch "a b" //创建1个文件

touch 'x y' //创建1个文件

a=10

echo "$a" //显示变量的值(内容)

echo '$a' //显示$a,$是特殊符号,功能被屏蔽

``(反撇) 获取指令的结果

a=date //定义变量a,内容是date

a=`date` //定义变量a,内容是date命令执行的结果

stty -echo //关闭回显

stty echo //恢复回显

#!/bin/bash

read -p "请输入用户名" u

useradd $u

stty -echo

read -p "请输入密码" p

stty echo

echo $p |passwd --stdin $u

发布全局变量

a=10 局部变量

echo $a //测试

bash //开启新解释器(进入子进程)

echo $a //测试,没有值

exit //退出新解释器

export a=10 //创建新变量,并发布成全局变量

export a //对已有的变量,发布全局

bash

echo $a //测试,有值,因为a已经成为全局变量

exit

export -n a //撤销全局变量,恢复局部变量

bash

echo $a //测试,无值,已经撤销

exit

shell中的运算

1,

expr 1 + 1 或者 expr $a + 1

expr 1 - 1 .....

expr 1 \\* 1

expr 1 / 1

expr 1 % 1 求模(取余数)

\ 转义符号,屏蔽之后一个字符的特殊含义

2,

echo $[1+1] 或者 echo $[a+1]

echo $[1-1] echo $[a-a]

echo $[1\*1]

echo $[1/1]

echo $[1%1]

3,let 可以改变变量本身的值, 不显示结果

let a=a+1 要写成 let a++

let a=a+2 要写成 let a+=2

let a=a-1 要写成 let a--

let a=a\*10 let a\*=10

let a=a/5 let a/=5

...

echo $a 查看结果

4,

echo "1+1" | bc //使用bc进行非交互式计算

echo "1.1+1" | bc //并且可以计算小数

echo "10/3" | bc //不定义小数点后面长度

echo "scale=3;10/3" | bc //定义小数点后面长度是3位