沈超：

微博：weibo.com/lampsc

一 shell简介

1 概念

命令解释器

2 常见shell

bash [**Linux**](http://lib.csdn.net/base/linux)标准shell

附：她保存在/etc/bash里面，是一个二进制可执行文件，所以不能直接：

vim/bin/bash 会是乱码

sh 早期shell，较简单

csh（UNIX默认的Shell！） ksh tcsh unix shell

附：他们的大部分是相同的，但是也会有所区别，比如配置文件的保存位置

vi /etc/shells linux支持的shell

直接输入csh并不报错，因为他已经从bash切换到了csh

（返回exit）

3 shell脚本

附：创建文件夹/文件目录：mkdirsh

Linxu并不区分扩展名，这点与Windows不同!

#！/bin/bash

例1：打印时间

#!/bin/bash

echo"hello world!"

脚本执行方式：

1） 赋予执行权限(惯用法)

如：chmod755 hello.sh

路径执行 /root/shell/hello.sh#绝对路径

./hello.sh #相对路径

2） 不赋予执行权限

bash 脚本名

如：bashecho.sh

4 bash常见功能

1）历史命令

默认保存1000条历史命令：

vi /etc/profile 修改环境变量配置文件，要生效，必须注销

HISTSIZE=1000 修改默认历史命令条数

history 查询系统历史命令

历史命令保存文件

~/.bash\_history

history -w 把内存中命令历史，保存入文件

history -c 清空所有的历史命令

重复历史命令

！n 重复第n条命令

！str 重复最后一个以str开头的命令

上箭头 调用上面的命令

   2）别名

附：Linux默认使用ls之后会以蓝色显示文件夹

但是也可以使用：ls-- color=never 来取消显示颜色

alias 查看系统中生效的别名

附：

alias ls='ls --color=never' 手工设定别名，临时生效：相当与个人操作习惯

aliasls='ls –color=tty' #改回来了

vi ~/.bashrc 写入别名，永久生效#第二个环境变量配置文件

5 输入输出重定向

标准输入 /dev/stdin 0 键盘

标准输出 /dev/stdout 1 显示器

标准错误输出 /dev/stderr 2 显示器

#设备文件名 #文件描述符 #默认设备

1）输出重定向

把应该输出到屏幕的输出，重定向到文件。

> 覆盖

>> 追加

ls > aa 覆盖到aa

ls >> aa 追加到aa

ls gdlslga 2>>aa 错误信息输出到aa 强调：错误输出，不能有空格

ls &>aa 错误和正确都输入到aa

掌握（查看系统版本:uname -r）

ls >> aa 2>>bb 正确信息输入aa，错误信息输入bb

ls >> aa 2>&1 错误和正确都输入到aa，可以追加

2>&1 把标准错误重定向到标准正确输出

6 多命令顺序执行

1）命令1 ； 命令2；命令3 命令123顺序执行。之间没有任何关系如：cd;ls ;date #date 用于打印当前系统时间

2）命令1 && 命令2 命令1正确执行后，命令2才会执行

3）命令1 || 命令2 命令1执行不正确，命令2才会执行，也就是说：如果第一个命令正确执行，那么第二个命令就不会执行了！

类test[**测试**](http://lib.csdn.net/base/softwaretest)：

a)ls && echo "Yes" || echo "No"

b)ls skdjfkdj && echo "Yes" || echo "No"

lsaa && echo "cunzai" || echo "bu cunzai!" 执行ls aa，判断如果正确，输出“存在”。如果不存在，输出“不存在”

7 管道符：两个命令之间有联系，

命令1 | 命令2 命令1的执行结果，作为命令2的执行条件

注意：第一条命令一定得有输出

netstat-tlun | grep 80 查询监听的端口号，并查看80端口是否启动。

附：

netstat-tlun列出监听了多少个端口；

grep提取命令，如 grep80 #提取包含有80的行

80是Apache的端口

ls -l /etc/ | more 分屏显示ls内容

ls-l /etc/ | grep vim

附：more分屏显示文件内容

二 变量：Linux将所有的变量都识别程字符串型#应该是Shell吧？Idon‘t know！

设置变量的目的：像程序脚本中传值。

1分类

本地变量

环境变量

位置参数变量

预定义变量

2本地变量

1）声明 变量名=变量值 注意：=号左右不能有空格

aa=123

2）调用

echo $变量名

3）查看变量

set 查看所有变量，包括环境变量和本地变量

4）删除

unset 变量名

3变量设定规则

1）变量以等号连接值，等号不能有空格

2）变量名由数字和字母和下划线组成，不能以数字开头

3）变量值中有空格，用引号括起来如：aa=”helloworld” #其中不能有符号！！！

4）双引号内，有特殊字符。如$

5）单引号中特殊字符无含义

6）在变量值中，可以使用\转义符

7）变量值可以直接调用系统命令。 `命令` $(命令)

如cc=`ls/dev` # ` 是反引号，Esc键下的那个

或者：cc=$(ls/dev) 作用相同，推荐！

调用：echo$cc

8）变量值可以累加 aa=123 aa="$aa"456 echo $aa --->123456

9）环境变量一定设为大写

4环境变量：

附：pstree：查询进程树

一般不要声明环境变量

1）声明

export 变量名=变量值

export aa #变量已经存在的情况

2）查看

set 查看所有变量

env 或者export 只能查看环境变量，并且export能够看清本质-\_-

declare 声明变量类型的，如果不特别声明，所有变量为字符串型

-i 声明为int

-x 声明为环境变量

3）删除

unset 变量名

4）常见环境变量

echo$PATH

/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

系统查找命令的路径

PATH="$PATH":/root/shell 在系统默认路径后，追加/root/shell目录作为命令查找路径

5）环境变量配置文件

/etc/bashrc 所有用户生效

/etc/profile

~/.bashrc只对指定用户生效

~/.bash\_profile