

# Unix基础介绍

# 内容索引

---

- ◆ Unix操作系统介绍
- ◆ Unix常用命令
- ◆ Vi编辑器的使用
- ◆ Shell编程

# Unix操作系统介绍-Unix简介

- ◆ UNIX是较早广泛使用的操作系统，它的第一版于1969年在BELL实验室诞生，1975年对外公布，至今应用到几乎所有类型的计算机上。UNIX主要特点如下：
- ◆ 开放式系统(Open Operate System)
- ◆ 多任务、多用户
- ◆ 并行处理能力，稳定性好
- ◆ 安全保护机制，功能强大的shell
- ◆ 强大的网络支持，Internet上各种服务器首选操作系统

# Unix操作系统介绍-相关名词

- ◆ 多用户 ( Multi-User ) 允许多个用户同时使用计算机的能力
- ◆ 多任务 ( Multi-Tasking ) 为每一个用户同时执行多个任务的能力
- ◆ 用户 ( User ) UNIX是多用户的操作系统，她允许多个用户同时登录。每个用户可以拥有自己独特的操作权限，提示符一般为"\$"
- ◆ 用户组 ( Group ) 同组内用户可以共享权限和信息
- ◆ 超级用户 ( Root ) UNIX特权用户，拥有全部权限，提示符一般为"#"

# Unix操作系统介绍-主流Unix操作

---

- ◆ IBM AIX
- ◆ HP-UX
- ◆ Linux(RedHat,RedFlag,SUSE...)
- ◆ SCO UnixWare ( 被LINUX替代 )
- ◆ Sun Solaris

# Unix操作系统介绍-Unix的组成

- ◆ UNIX操作系统由三大部分组成：Kernel（内核）、Shell（外壳）、工具及应用程序
- ◆ UNIX Kernel是操作系统的核心，指挥调度机器运行，直接控制计算机资源，保护用户程序不受硬件事件影响
- ◆ UNIX Shell是Kernel和用户之间的接口，是命令解释器，常见有B Shell、K Shell、C-Shell
- ◆ K Shell是程序员较为常用的一种Shell，本次培训仅以K Shell为例



# Unix操作系统介绍- Unix文件系统术语

- ◆ 文件类型-Unix上所有的目录、设备、文件均是文件系统的组成部分，分属不同的文件类型：普通文件、设备文件、目录和符号链接
- ◆ i节点：它是UNIX内部用于描述文件特性的数据结构.我们通常称i节点为文件索引结点(信息结点).i节点 含有关于文件的大部分的重要信息,包括文件数据块在磁盘上的地址.每一个i节点有它自己的标志号,我们称为文件顺序号.i节点包含的信息 1.文件类型 2.文件属主关系 3.文件的访问权限 4.文件的时间戳.

# Unix操作系统介绍- Unix文件系统术语

- ◆ 文件权限：指明一个文件的控制权限、所有者等
- ◆ 隐藏文件：以'.'开头的文件，属于系统隐藏文件，需要以特殊命令才能看到
- ◆ 文件路径：指明文件所在的目录及文件名称
- ◆ bin目录：可执行文件目录
- ◆ sbin/scripts:shell脚本文件目录
- ◆ lib:库文件目录
- ◆ etc:配置文件目录
- ◆ dev:设备文件目录
- ◆ include:头文件目录
- ◆ src:源代码文件目录



# Unix操作系统介绍- Unix文件类型

- ◆ 目录 : d
- ◆ 符号链接 : l
- ◆ 字符设备文件 : c
- ◆ 块设备文件 : b
- ◆ 命名管道文件 : p
- ◆ 套接字文件:s
- ◆ 普通文件 : - , 也就是不属于以上几种文件类型的文件
- ◆ 普通文件又分文本文件(text)、二进制文件(binary)和数据文件(data)

# Unix操作系统介绍- 文件权限标识

权限标识	文件	目录
r	可以读取文件	可以读取文件名称
w	可以修改文件内容	可以修改目录下的内容（增加、删除文件、修改文件名、建立新目录等）
x	可以执行文件	可以访问该目录
s	特殊权限	只有超级用户才可以执行该文件

# Unix操作系统介绍-文件用户属性

属性类型	用户属性说明	权限说明
owner	文件、目录的所有者	所有者权限，chown,chmod等操作
group	所有者所在的用户组名称	用户组权限
other	其他所有用户	其他用户权限
root	超级用户	超级用户可任意修改文件权限，可做任何操作

# Unix操作系统介绍-文件属性示例1

- ◆ 每种类型的用户都有三种权限r、w、x
- ◆ \$ ls -l myfile
- ◆ -rwxrw-r-- 1 star32 informix 123 Aug 7 11:24 myfile
- ◆ 第一列第一个字符表示文件类型，如果为-则表示是普通文件,为d则表示为目录，其他字符表示为其他类型的文件系统、设备;之后每三个字符分别表示所有者、组用户和其他用户的读、写、执行权限，无权限则表示为'-'。文件的权限可以由所有者或超级用户使用chmod修改；
- ◆ 第二列为目录下的文件/目录个数，文件则为1

# Unix操作系统介绍-文件属性示例2

- ◆ 第三、四列分别表示文件的所有者、所属组，文件的所有者所属组可以由所有者和超级用户使用chown/chgrp修改
- ◆ 第五列为文件大小（字节数），对于目录不必在意
- ◆ 第六七八列为文件的最后修改日期
- ◆ 第九列为文件名
- ◆ 如果文件类型为链接（第一列第一字符为'l'）则第九列之后显示所链接的源文件路径,例如：
- ◆ lrwxrwxrwx 1 star32 informix 10 7月 22

06:40 GB18030 -> /zh\_cn.gb



# Unix操作系统介绍- 文件路径

- ◆ 根目录：/
- ◆ 用户主目录：/home/star31
- ◆ 当前目录：.
- ◆ 上级目录：..
- ◆ 相对路径：相对路径是不以/开头，相对于当前路径、用户主目录的路径，如：  
bin/ESLISTEN,ESLISTEN，./bin/ESLISTEN
- ◆ 绝对路径是以/开头的文件路径：  
/home/star31/bin/ESLISTEN
- ◆ 用cd可以改变当前目录，用pwd可以获得当前路径



# Unix操作系统介绍-环境变量

- ◆ 环境变量是操作系统提供的为执行程序提供所需要配置的一些参数的机制
- ◆ 环境变量通过在shell中或者用户配置文件中设置的方式生效：如HOME=/home/star31;export HOME
- ◆ shell中通过\$HOME获得HOME环境变量的值：/home/star31，而应用程序中则通过getenv(“HOME”)
- ◆ HOME是用户主目录环境变量
- ◆ PATH是用户可执行文件所在路径:PATH=/bin:/usr/bin:\$HOME/bin:\$HOME/scripts:..
- ◆ LIBPATH/LD\_LIBRARY\_PATH等是程序运行所需动态库文件所在路径

# Unix操作系统介绍-用户配置文件

- ◆ 不同的shell有不同的用户配置文件名.
- ◆ 用户配置文件是用户登录后，shell自动运行的脚本文件，主要用途在于设置用户的环境变量和默认操作，包括程序路径、动态库路径、用户名、用户提示符等等
- ◆ 可以将其理解为用户的启动文件

# 内容索引

---

- ◆ Unix操作系统介绍
- ◆ Unix常用命令
- ◆ Vi编辑器的使用
- ◆ Shell编程

# Unix常用命令-综述

- ◆ 在Shell提示符下可以输入Unix命令，命令基本格式如下：
- ◆ Command 参数1 参数2 ... 参数n
- ◆ Unix命令与Dos命令格式相近，但是需要注意Unix命令是区分大小写的
- ◆ 一般Command -?/Command -help会提示该命令的参数提示信息

# Unix常用命令-获得帮助

## ◆ 获得Unix命令帮助信息的两种方式

- man command
- command -?

## ◆ 命令参数帮助信息基本约定

- [options] : 中括号中间的参数可以不使用
- {optiona|optionb}:花括号中间用竖线分割的参数不能同时使用
- -option 表示参数是标志——不是表示可选，后面可能带真正的参数值

## ◆ - ? 和--help一般是通用的获得命令行参数帮助信息的“选项”

# Unix常用命令列表(1)

命令	解释	示例
ls	列表文件	ls -a abc
cd	切换当前工作路径	cd ../cd /
vi	编辑文件（文本文件或数据文件）	vi src/mysource.c
pwd	获得当前路径	Pwd
rm,rmdir	删除文件，删除空目录	rm -f tobedeletefile
mkdir	创建目录	mkdir log
man	查看帮助	man fprintf;man ls
>,>>	重定向输出，追加输出	>newfile;ls -l>listfile



# Unix常用命令列表(2)

命令	解释	示例
ps	显示当前执行的进程状态	ps -ef
find	查找文件	find . -name "*.txt" -print
clear	清除屏幕	
export	设置环境变量	export PATH=\$PATH:\$HOME/sbin
echo	回显	echo "haha"
passwd	修改用户口令，root用户可以修改其他用户口令	

# Unix常用命令列表(3)

命令	解释	示例
grep	查找文件内容	grep word myfiles.* ; ps -ef grep myprocess
date	查看日历，设置时间	date 08071336
more	一次一屏显示文件	more mylongfile.c
cp	拷贝文件	cp mysrc mydst
mv	移动文件	mv mytxt.txt mysource.c
netstat	查看网络状态	netstat -in

# Unix常用命令列表(4)

命令	解释	示例
sar	显示系统活动报告	sar 1 100
iostat	显示当前系统CPU、输入输出使用状态	iostat 1 100
top/topas	操作系统监控命令	其中AIX为topas，HP、Linux可使用top
uname	打印系统信息	uname -a
chown	改变文件属主	chown star31:informix mydir
chmod	改变文件/目录的权限	chmod 750 myfile
su	切换用户	su - newuser

# Unix常用命令列表(5)

命令	解释	示例
telnet	连接并登陆远程主机	
ftp	在本机与远程主机间传输文件	
exit	退出登陆用户，返回登陆界面	
id	查看当前登陆用户的信息	
tty	查看当前用户设备文件编号	
sed	一个面向数据流的编辑工具	
awk	样式扫描与处理工具	

# Unix命令中常用的字符

- ◆ 管道符'|':将前一个命令的输出当作后一个命令的输入：ps -ef|grep init
- ◆ 重定向符'>',将一个命令的输出重定向到一个文件，其中'>>'表示以追加方式重定向到该文件
- ◆ 命令执行符'.'：如ls `cat filelist`
- ◆ 后台执行符'&':命令行结尾加上&则该命令在后台执行，不等用户交互

# Unix常用命令-find

- ◆ find命令会在指定目录及其子目录下查找符合条件的特定文件：
- ◆ 命令格式：
- ◆ find 目录名 条件
- ◆ 目录名：指定在搜索的目录，可以有多个目录，之间用空格分隔
- ◆ 条件：搜索文件的条件，可以包括文件名称、拥有者、最后修改时间等等



# Unix常用命令-find命令参数解释

条件列表说明	
-name name	指定寻找的文件和目录名称，如：-name '*.c'
-print	将符合条件的文件路径打印出来
-size n	寻找占用n个block的文件
-type x	以文件类型做为查找条件，x如下：d--目录、f--文件、b--块、c--字符、p--管道
-links n	寻找链接数等于n的文件
-atime n	寻找n天前被存取的文件
-mtime n	寻找n天前被修改的文件
-exec command { } \;	用寻找到的文件做为执行command的对象，{ } 内存执行command命令时所需的条件

# Unix常用命令-grep命令

- ◆ grep在整个文本文件中寻找特定字符串，并将出现该字符串的整行内容打印：
- ◆ 命令格式：
- ◆ grep 字符串 文件名
- ◆ 举例：
- ◆ grep abcd test 在test文件中查找abcd字符串
- ◆ grep "who am i" test 在test文件中查找"who am i"字符串由于含有空格，必须用""

# Unix常用命令- chmod命令

## ◆ chmod who oper permission file

- Who:u,g,o,a分别表示属主，组，其他和所有
- Oper:+,-,=分别表示增加、取消和设定权限
- permission:r,w,x读写执行权限
- chmod og-rwx o+r g+rx u+w file
- -r-x—xrw-

## ◆ chmod mode file

- mode为权值表示的权限，三位，分别表示属主、组及其他用户
- 每一位表示一种用户的权限
- 4=r,2=w,1=x，按位或的结果或者直接相加，就是一种用户的权限了
- chmod 754 file等同于上面的chmod语句

## ◆ umask mode设置新文件权值掩码

- ❖ umask 022则新创建的目录默认权值为755,注意文件缺省无执行权限，因此新创建的文件权限为644

# Unix常用命令- sed命令

- ◆ sed是一个面向数据流的编辑工具(stream editor)，可以通过命令直接修改文件，和常用的可视编辑器采用了不同的方式来修改文件。比如我们需要修改文件中的hello为note，在可视编辑器我们需要1.点击替换功能；2.输入查找hello，输入替换note；3.点击替换按钮，而是用sed 我们输入命令 `sed -e 's/hello/note/g' <文件名>` 执行就可以了。这种方式带来的优点就是我们可以使用sed脚本，作用于大量文件，并且会避免很多错误，Sed一般的使用方式有两种：
- ◆ `sed -e '命令' 文件名` 在命令行直接执行命令
- ◆ `sed -f 脚本名 文件名` 把sed脚本写在脚本文件中执行

# Unix常用命令- awk命令

- ◆ awk一个优秀的样式扫描与处理工具，awk的功能与sed和grep很相似，但其功能却大大强于sed和grep。awk提供了极其强大的功能：它几乎可以完成grep和sed所能完成的全部工作，同时，它还可以进行样式装入、流控制、数学运算符、进程控制语句甚至于内置的变量和函数。它具备了一个完整的语言所应具有的所有精美特性。所以awk又被称为样式扫描与处理语言。
- ◆ `ps -ef|awk '{print $3, $10}'`



# Unix常用命令-alias

- ◆ alias用于设置别名或者显示当前的所有别名设置
- ◆ alias cds='cd \$HOME/src/'
- ◆ 设置别名后，别名就等于所代表的一串命令



# 内容索引

---

- ◆ Unix操作系统概述
- ◆ Unix常用命令
- ◆ Vi编辑器的使用
- ◆ Shell编程

# vi编辑器使用-进入vi

- ◆ 在系统提示字符(如\$、#)下敲入vi < 档案名称 > , vi 可以自动帮你载入所要编辑的文件或是开启一个新文件 ( 如果该文件不存在或缺少文件名 ) 。进入 vi 后荧幕左方会出现波浪符号 , 凡是列首有该符号就代表此列目前是空的。
- ◆ vi带文件名参数则打开或新建一个文件并打开该文件
- ◆ vi [file1...filen]

# vi编辑器的使用-vi的两种模式

- ◆ vi存在两种模式：指令模式和输入模式。在指令模式下输入的按键将做为指令来处理：如输入a，vi即认为是在当前位置插入字符。而在输入模式下，vi则把输入的按键当作插入的字符来处理。指令模式切换到输入模式只需键入相应的输入命令即可（如a,A），而要从输入模式切换到指令模式，则需在输入模式下键入ESC键，如果不晓得现在是处于什麼模式，可以多按几次 [ESC]，系统如发出哔哔声就表示已处于指令模式下了。

# vi编辑器的使用- vi指令模式进入编辑模式的指令

- ◆ a : 从光标所在位置後面开始新增资料，光标後的资料随新增资料向後移动
- ◆ A : 从光标所在行最末面的地方开始新增资料
- ◆ i : 从光标所在位置前面开始插入资料，光标後的资料随新增资料向後移动
- ◆ I : 从光标所在行的第一个非空白字元前面开始插入资料
- ◆ o : 在光标所在行下新增一行并进入输入模式。
- ◆ O: 在光标所在行上方新增一行并进入输入模式。

# vi编辑器的使用-保存和退出vi

- ◆ 在指令模式下键入:q,:q!,:wq或:x(注意:号) , 就会退出vi。其中:wq和:x是存盘退出, 而:q是直接退出, 如果文件已有新的变化, vi会提示你保存文件而:q命令也会失效, 这时你可以用:w命令保存文件后再用:q退出, 或用:wq或:x命令退出, 如果你不想保存改变后的文件, 你就需要用:q!命令, 这个命令将不保存文件而直接退出vi。
- ◆ 如果只是要保存文件而不退出, 可以:w或者:w newfilename
- ◆ 注意, 指令模式下输入完命令,是需要按回车的(模式切换命令除外)



# vi编辑器的使用-指令模式编辑命令(1)

- ◆ x 删除光标所在位置起的n个字符
- ◆ dd 删除光标所在位置起的n行
- ◆ rC 替换从光标所在位置起的n个字符为新C字符，C表示要替换成的字符
- ◆ R进入替换状态，新增文字会覆盖原先文字，直到按 [ESC] 回到指令模式下为止
- ◆ s删除光标所在字元，并进入输入模式
- ◆ S删除光标所在的行，并进入输入模式
- ◆ cw删除光标所在的词，并进入输入模式



# vi编辑器的使用-指令模式编辑命令(2)

- ◆ 上下左右键和k,j,h,l键作用相同，但jklh键可以加数字，如3j表示光标向下移动3行，5h表示光标向左移动5字符
- ◆ ^表示当前行第一个非空白字符
- ◆ \$表示当前行尾
- ◆ G ( 大写 ) 表示当前文件最后一行
- ◆ w移动到下一个单词的首字母
- ◆ e移动到下一个词的最后一个字母
- ◆ b移动到本词或者前一个词的第一个字母
- ◆ CTRL+B/F向上/向下翻一页
- ◆ :nn 表示移动到nn行 ( nn是数字 )

# vi编辑器的使用-指令模式下的编辑指令 (3)

- ◆ .执行上次的命令一次
- ◆ ,取消上次的命令一次 ( 与u相同 )
- ◆ yy拷贝一行
- ◆ p将之前拷贝或者删除的内容粘贴到光标之后 ( 如果是行则粘贴到下一行 )
- ◆ P类似p但作用是光标之前
- ◆ /string 从当前光标位置起向后查找字符串string并将光标停在其首字母
- ◆ ?string类似/但作用是向前查找
- ◆ n配合/或者?使用作用是重复一次
- ◆ N配合/或者?使用作用是反方向重复一次

# vi编辑器的使用-指令模式下的文件操作指令

- ◆ :r filename 读取文件内容到光标所在位置下一行
- ◆ :e 重新装入本文件的上次保存内容
- ◆ :w [filename] 保存文件[到filename]，如果要替换已经存在的文件则为:w! [filename], ! 表示强制
- ◆ :wq 保存文件并推出，如果当前编辑内容无保存文件名则不能执行（直接输入vi进入编辑模式的情况）
- ◆ :q 推出vi编辑器，如果文件内容已修改未保存则失败
- ◆ :q! 放弃修改内容，退出编辑器
- ◆ :x 等于:wq

# vi编辑器的使用-指令模式下的其他指令

- ◆ :set list显示转义字符
- ◆ :set nolist 不显示转义字符
- ◆ :help显示帮助
- ◆ v进入或退出可视模式
- ◆ :tabstop=4设置TAB键字符宽度

# vi编辑器的使用-编辑模式下的“指令”

- ◆ 编辑模式下无指令，输入的任何字符都当做输入的内容
- ◆ 编辑指令下移动光标可以使用上下左右键
- ◆ 退格键(BackSpace ) 表示删除光标前的字符
- ◆ Delete表示删除光标所在的字符

# 内容索引

---

- ◆ Unix操作系统介绍
- ◆ Unix常用命令
- ◆ Vi编辑器的使用
- ◆ Shell编程



# Shell编程-概述

- ◆ shell程序是一个包含UNIX命令的普通文件，这个文件的许可权限至少应该为可读和可执行，在shell提示符下键入文件名就可执行shell程序
- ◆ shell程序可以通过三种方式接受数据：
  - 环境变量
  - 命令行参数
  - 用户的输入

# Shell编程-概述

- ◆ shell是一个命令解释器，它会解释并执行命令提示符下输入的命令。但是，你可能想要多次执行一组命令，shell提供了一种功能，让你将这组命令存放在一个文件中，然后你可以象unix系统提供的其他程序一样执行这个文件，当你运行这个文件，它会象你在命令行输入这些命令一样地执行这些命令。这个文件我们称为shell脚本。
- ◆ 你可以写出非常复杂的shell脚本，因为shell脚本支持变量、命令行参数、交互式输入、tests（判断）、branches（分支），和loops（循环）等复杂的结构
- ◆ 理论上，shell脚本能够实现任何编程语言能够实现的功能

# Shell编程-变量

- ◆ 环境变量：Shell中可以使用所有的已设置环境变量，也可以设置新的环境变量以供shell本身使用，在shell中使用\$变量名即可获得环境变量的值
- ◆ 本地变量：Shell中可以设定和使用自己的变量，类似于C语言中的局部变量，但是不需要变量的声明
- ◆ 特定变量：shell中有特定含义的变量，代表shell脚本的参数、返回值等变量

# Shell中的特定变量

变量使用	含义
\$#	传递到脚本的参数个数
\$0-\$n	第0到第N个参数，\$0表示脚本本身的路径名
\$*	以一个单字符串显示所有向脚本传递的参数
\$\$	脚本运行的当前进程号
\$_	后台运行的最后一个进程的进程号
@	与\$#相同，但是使用时加引号并在引号中返回每个参数
-	显示shell使用的当前选项，与set命令功能相同
?	显示最后执行命令的退出状态，0表示成功，其他值表示错误

# Shell编程 - 关键符号

- ◆ # 注释符，shell中只支持行注释，且只有#后面的内容被注释
- ◆ \转义符，跟C语言相同
- ◆ “”双引号：将双引号中间的所有字符当做一个字符串
- ◆ ‘’单引号：与双引号基本类似
- ◆ ( )括号：函数
- ◆ {}花括号:块语句标识
- ◆ []中括号:一些语句的特定字符(相当于test语句)；正则表达式的特定字符
- ◆ ``反引号:shell将反引号中间的字符串当做一个命令，并将其输出作为结果赋值给变量

# Shell编程-关键字符

- ◆ &
- ◆ \* 乘号，正则表达式通配符
- ◆ + 加号
- ◆ - 减号
- ◆ ^ 正则表达式开始位置标识
- ◆ \$ -环境变量和特定变量引用符，正则表达式结尾标识
- ◆ | 管道符
- ◆ ? 单字符通配符



# Shell编程-输入

- ◆ 通过环境变量输入 : `find $HOME -name "*.txt"`
- ◆ 通过命令行参数 : `case $1 in ... case 1)...;;easc`
- ◆ 用户交互读入 : `read char;echo $char`

# Shell编程-条件测试

- ◆ 对于文件、字符串和数字，使用test命令
- ◆ 对于数字和字符串使用expr命令
- ◆ test测试语句有两种形式
  - ❖ test condition
  - ❖ [ condition ]
- ◆ 测试时使用的逻辑操作符：-a ( 与 ), -o ( 或 ), ! ( 非 )
- ◆ 文件测试：-d, -f, -L, -r, -s, -w, -x
- ◆ 字符串测试：= ( 等于 ) , != ( 不等于 ) , -z ( 空串 ) , -n ( 非空串 )
- ◆ 对数字测试：-eq, -ne, -gt, -lt, -le, -ge

# Shell编程-流程控制语句

- ◆ if then else fi判断语句
- ◆ while for until loops条件循环语句
- ◆ case分支匹配语句，类似于C的switch语句
- ◆ return 函数返回语句
- ◆ exit 退出并结束脚本

# Shell编程-函数

## ◆ 定义函数格式：

function 函数名() #function可加可不加,函数名在整个脚本中必须唯一

```
{  
    命令1  
    ...  
    return 返回值  
}
```

## ◆ 使用函数：

retval=函数名 参数1 , ...参数n

## ◆ 函数返回值

if [ "\$?" = "0" ];then...fi

# Shell编程-示例(1)

```
#!/bin/sh
```

#shell中第一行一般必须为#!后面指定所使用的shell类型

```
case $# in #条件控制语句
```

```
case 0 ) #case value)表示输入的值为value时，等于C语言switch里面的  
case n
```

```
echo "no argument input"
```

```
:: #条件分支结束符，相当于C的break
```

```
case 1)
```

```
echo "one argument input:$1"
```

```
::
```

```
case *) #*是通配符，因此此语句相当于C的default
```

```
::
```

```
esac #case语句块结束标识，其他如if结束为fi，大家发现什么规律没？
```

© All Rights Reserved, Copyright 2001, Advanced Digital Technology CO.,Ltd

# Shell编程-示例(2)

```
#!/bin/sh
```

```
filelist=`ls $HOME` #通过ls命令的输出获得文件列表
```

```
for file in $filelist #逐一检查每个文件
```

```
do
```

```
    if [-d $file ]
```

```
    then
```

```
        echo "$file is a directory"
```

```
    elif [ -L $file ]; then
```

```
        echo "$file is a symbol link file"
```

```
    elif [ -s $file -a -x $file ];then
```

```
        echo "$file is a no empty excuteable file"
```

```
    fi
```

```
done #有人上当没？For语句结束不需要rof喔,do语句结束也  
不是od
```



# Shell编程-示例(3)

```
#!/bin/sh
if [ $# -lt 1 ] #无参数则认为要启动所有DTA
then
    TARGET="-b ALL"
elif [ $# -gt 1 ] #多于一个参数则认为是要启动某机器的某个DTA
then
    TARGET="-m $2 -b $1"
else #一个参数就是要启动所有机器的某个DTA
    TARGET="-b $1"
fi

esclient <<! #esclient是Starring软件的可执行程序—客户端
mode $TARGET #mode是esclient的交互式命令,表示切换系统状态
quit #quit是大多数交互式软件的通用命令-退出
!
```

#<<!...!会把两个感叹号中间的内容当做交互式输入的内容，也就起到将交互式改为非交互的作用

# 参考书籍

- ◆ Unix使用入门—书店有售
- ◆ Unix基础知识—本公司电子文档
- ◆ vi基本操作方法—本公司电子文档
- ◆ 0632 Linux 与Unix Shell编程指南 —来自网络的电子文档
- ◆ UNIX环境高级编程—Unix C编程的圣经

# Questions? scoadmin

