

Unix 系统 薛卫

开卷 理论, 上机

ctrl + c 强制关当前命令

ctrl + d 关终端

Unix & Linux 大学教程 > 清华 (要买) 下周逢单周实验 8 号机房

Unix 是一类系统

Linux

版本号

1. 2. 0

→ 偶 (产品化)

1. 3. 0

→ 奇 (测试版)

符号链接

Unix 文件系统

普通文件, 目录文件, 特殊文件

一个根 / 任何对象都是文件

设备文件, 命名管道

proc 文件

伪文件

\$ls -la

发行版: Ubuntu, Fedora, openSUSE

list

Debian, Mandriva, Mint, CentOS

特殊/设备文件 不包含任何数据

FreeBSD, Redhat

(上机用)

→ Apache

/proc 存在于内存中而不是硬盘上

/proc process 进程

→ Nginx

俄罗斯人

cat /proc/cpuinfo

Shell

是一种脚本语言, 独立于操作系统的程序  
解释性执行

符号链接 / 硬链接 / 软链接

BASH (GNU) (默认用 BASH)

适合文件整理

查看登陆信息 dmesg

在命令行要打开图形化界面 startx

命令行下如何关机:

halt 立即关机

poweroff

shutdown -h 10 (10 分钟后关机)

init 是所有进程的祖先是

Init + 运行级别

Init 0 关机

Init 5 (是 xwindows)

Init 6 重启

root

dev 设备

etc 配置

inode	
metadata	user data

无数据

硬链接 In hello.txt 别名.txt

软 In -s hello.txt 别

FHS

/ 根

bin 基本程序

home

/opt 第三方应用程序(可装软件)

/usr/src 放内核源代码

. 当前目录

.. 父目录

pwd cd mkdir rmdir ls

当前目录  
的完全  
路径

删目录

print working directory  
看hello.c

cat /home/al/hello.c

显示内容

\$ cd 回主目录

\$ cd .. 回当前的父目录

ls -a 显示所有文件(包括隐藏文件)

ls -R 显示所有子目录的文件

Vi 三种工作模式

命令

输入

末行(相当于文件菜单)

/etc/passwd 保存用户信息文件(不能乱改)

(并不是密码)

/etc/shadow

重定向 >

0 输入 1 输出 2 错误

ls 错误 1 > output 2 > &1

都输出到 output

/dev/null 垃圾桶 "黑洞"

管道线

一个程序的标准输出可以自动发往

另一个程序的标准输入

← 管道操作

cat file1 file2 | sort

date > currentdate.txt

cat /etc/passwd | tee userids | wc -l

过滤器

重定向创建文件

cat > data 重定向到 data, 输多行文字

cat >> data 追加

复制文件 cat < data > newdata

组合文件 cat file1 file2 file3 > file4

tac 最后一行到第一行

rev 每行字符反转

head [-n lines] 显示前 n 行

tail [-n lines]

colrm 删除数据列

colrm [开始列数编号 < 结束列编号]

cat data1, data2 | wc -l

cat tac rev tac.. | rev

odmesg | tail -25

比较有序文件

comm 12

comm 23

comm 123 (不显示)

diff 版本管理

diff -c 甲 乙

diff -u 甲 乙

→ 统一输出格式

diff -y 甲 乙

→ 并排格式输出

diff 甲 乙 -q

(查看甲乙是否不同)

Win 换行 \r\n

Mac \r

Unix \n

★ 换行符的转变

dos2unix

mac2unix

unix2dos

Who | cut -c 1-8

↑  
谁登录

↑  
抽取指定的列

↑  
字符区间

cut -f list [-d deli] [-s] [file...]

↑  
分隔符

↑  
抑制

显示/etc/passwd的用户和shell

cat /etc/passwd cut -d ':' -f 1,7

若不想输出分隔符 -s

paste 组合数据列

patch [options] [originalfile] [patchfile]

修改文件

内核升级

Linux 源代码安装

patch 可以

eg. 3.0 ~ 升级到 3.4

diff, patch 避免不必要的复制打补丁

正则表达式

grep = global search regular expression and print out the line

• 匹配符号 \* 0或多个是前缀

\\$ \. 让之符号回到原来的含义

[ ] 匹配指定范围

^ 开始 \$ 结束

0~9 [0-9] [0-2]

[^] 匹配一个不在指定范围的字符

\(love\)

标记匹配字符

所有的正数

^[0-9]+\$

整

^[0-9]+\$

~~test~~ ^ t[ae]st\>\$ [ag]oo

~~[0-9]~~ [0-9]+

^ [a-zA-Z]

-u

\. \$ 1998\>.

sort -t 分隔字符

^ [3/5] 1998

-o 重定向

^ [148]

sort -n -k 2 -t = hello.txt

~~oot~~ oot grep -e oot

gq 2.53+g

od

tr

t[ae]st

[ag]oo

[0-9]

^ [a]

o \f z \}

tr -s ['\n'] 多个回车压缩成一个

删除重复的小写字母 tr -s "[a-z]"

删除M, 代之M换行

sed '/^ha. #day#/c \replace line' test.txt

sed '/aaa/c r'''

sed '[0-9]\f3\}

sed 's[h]\f3\} \aaa'

shell编程

Bourne shell, Cshell

不兼容

#!/bin/sh

echo "hello"

exit 0

# for file in \*

>

>

> done

Shell 编程

#a #a[]

echo '我的当前目录是 pwd'

← 索引号  
← 命令

我的当前目录是 /home/curie

echo "my home is \$HOME"

内部变量

## 传给 shell 程序参数的个数

#? 上一条命令的返回值

## 本程序 PID

书 <linux 程序设计> shell 编程

var = \$(expr 1 + 1)

test 1 -lt 4

echo #? → less than

[ 1 -lt 4 ]

1 比 4 小吗?

[ 1 -lt 4 ]

要有空格

./hello.sh

\$ sh myshell.sh

if fi 调皮

case esac

case in

) do

esac

read 变量

y | yes | y\*)

for i in 'seq 1 9'

1 到 9 的序列

expr 1 \\* 10

计算

~~echo~~ '

while [ ]

until [ ]

do

done 1 加到 100

i=1

sum=0

while [ ]