Unix开发环境

目录

[第一章 Solaris Computing Environment](#_Toc10083_WPSOffice_Level1) [2](#_Toc10083_WPSOffice_Level1)

[1.认识Linux](#_Toc13754_WPSOffice_Level2) [2](#_Toc13754_WPSOffice_Level2)

[1.1 Linux的兴起与发展](#_Toc13754_WPSOffice_Level3) [2](#_Toc13754_WPSOffice_Level3)

[1.2.Linux系统结构(洋葱图)](#_Toc19631_WPSOffice_Level3) [3](#_Toc19631_WPSOffice_Level3)

[1.3 Kernel](#_Toc19010_WPSOffice_Level3) [3](#_Toc19010_WPSOffice_Level3)

[1.4 Shell](#_Toc2280_WPSOffice_Level3) [3](#_Toc2280_WPSOffice_Level3)

[1.5 The File System](#_Toc27334_WPSOffice_Level3) [4](#_Toc27334_WPSOffice_Level3)

[1.6 Pathnames](#_Toc24557_WPSOffice_Level3) [5](#_Toc24557_WPSOffice_Level3)

[第二章 Basic Linux Commands](#_Toc13754_WPSOffice_Level1) [6](#_Toc13754_WPSOffice_Level1)

[1.Linux基础命令](#_Toc27334_WPSOffice_Level2) [6](#_Toc27334_WPSOffice_Level2)

[1.1命令行语法](#_Toc5025_WPSOffice_Level3) [6](#_Toc5025_WPSOffice_Level3)

[1.2 基本命令](#_Toc21690_WPSOffice_Level3) [6](#_Toc21690_WPSOffice_Level3)

[第三章 Metacharacters](#_Toc19631_WPSOffice_Level1) [11](#_Toc19631_WPSOffice_Level1)

[1.元字符](#_Toc24557_WPSOffice_Level2) [11](#_Toc24557_WPSOffice_Level2)

[1.1元字符定义：](#_Toc31684_WPSOffice_Level3) [11](#_Toc31684_WPSOffice_Level3)

[1.2 常用元字符](#_Toc2769_WPSOffice_Level3) [11](#_Toc2769_WPSOffice_Level3)

[2.Text Viewer](#_Toc5025_WPSOffice_Level2) [12](#_Toc5025_WPSOffice_Level2)

[第四章 Full-screen Text Editor](#_Toc19010_WPSOffice_Level1) [12](#_Toc19010_WPSOffice_Level1)

[1.满屏文本编辑器](#_Toc2769_WPSOffice_Level2) [12](#_Toc2769_WPSOffice_Level2)

[1.1 vi介绍](#_Toc23959_WPSOffice_Level3) [12](#_Toc23959_WPSOffice_Level3)

[1.2打开vi 12](#_Toc23959_WPSOffice_Level3)

[1.3保存文件：](#_Toc14077_WPSOffice_Level3) [13](#_Toc14077_WPSOffice_Level3)

[1.4退出vi：](#_Toc19769_WPSOffice_Level3) [13](#_Toc19769_WPSOffice_Level3)

[1.5删除文本：](#_Toc22360_WPSOffice_Level3) [14](#_Toc22360_WPSOffice_Level3)

[1.6修改文本：](#_Toc21743_WPSOffice_Level3) [14](#_Toc21743_WPSOffice_Level3)

[1.7复制和粘贴文本：](#_Toc20867_WPSOffice_Level3) [14](#_Toc20867_WPSOffice_Level3)

[1.8高级功能：](#_Toc29021_WPSOffice_Level3) [14](#_Toc29021_WPSOffice_Level3)

[第五章 Setup Initialization Files](#_Toc2280_WPSOffice_Level1) [15](#_Toc2280_WPSOffice_Level1)

[1.配置文件](#_Toc23959_WPSOffice_Level2) [15](#_Toc23959_WPSOffice_Level2)

[1.1定义](#_Toc7166_WPSOffice_Level3) [15](#_Toc7166_WPSOffice_Level3)

[1.2种类](#_Toc23745_WPSOffice_Level3) [15](#_Toc23745_WPSOffice_Level3)

[1.3内容](#_Toc31477_WPSOffice_Level3) [16](#_Toc31477_WPSOffice_Level3)

[1.4相关指令](#_Toc22413_WPSOffice_Level3) [16](#_Toc22413_WPSOffice_Level3)

**第一章 Solaris Computing Environment**

## 1.认识Linux

## 1.1 Linux的兴起与发展

提起linux的缘起，不能不涉及UNIX。UNIX系统早期之所以能够取得巨大成功并迅速得到普及，主要在于其三个重要特点：简洁性、开放性与可移植性。向大学和研究机构公开了源代码，激发了软件开发人员的兴趣，从而使诸多大学以UNIX作为操作系统课程的研究对象，使UNIX成为大学操作系统课程的代名词，同时也培养了许多潜在的系统用户。

而后期的商业化运作方式，使得UNIX系统及其源代码成为专属产品，从而限制了对UNIX的研究、开发和使用。另外，为了考虑特定的及其结构，商业化的UNIX也变得越来越复杂，基本上失去了可移植性的特点。所以催生了Linux。

1984年，Richard Stallman发起了异常自由软件共享活动，创建了一个自由软件基金会，支持开发与共享自由软件，其中的GUN项目旨在开发一个完全免费的类似UNIX的系统，到了1990年，GUN几乎都有了自己的相应软件，但是缺乏一个内核。1991年在芬兰赫尔辛基大学读书的Linus Torvalds创建了一个类似于UNIX操作系统的内核，推出了Linux0.01版。

目前，存在许多不同版本的Linux产品，其中比较著名的就是Red Hat,Fedora,Deebian,Ubuntu等。Ubuntu Linux系统是众多Linux发行品牌之一。在Linux世界中，Ubuntu只是一后来者，短短几年就发展成为了一个流行的、成熟的以及桌面环境丰富的Linux系统，受到了从Linux初学者到资深专家的大批Linux用户的追捧。

## 1.2.Linux系统结构(洋葱图)

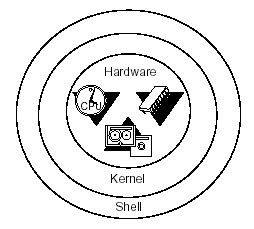


图1.1

Linux系统结构包含：1.Kernel：内核：驱动硬件工作的第一个软件

2.Shell：提供给用户一种交互方式。基于命令式的交互

3.File System:文件系统

4.Application:应用程序

## 1.3 Kernel

内核是操作系统的核心，具有很多最基本功能：

1.它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。

2.控制系统程序，系统工具和系统硬件的交互

3.管理交换空间，精灵程序，文件系统和其他功能。

## 1.4 Shell

shell是系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行，是一个命令解释器。另外，shell编程语言具有普通编程语言的很多特点，用这种编程语言编写的shell程序与其他应用程序具有同样的效果。目前主要有下列版本的shell。

1．Bourne Shell:是贝尔实验室开发的。由Stephen Bourne为AT&T Unix而写,是最早最有名的Shell。

2．BASH：是GNU的Bourne Again Shell，是GNU操作系统上默认的shell,大部分linux的发行套件使用的都是这种shell。

3．Korn Shell：是对Bourne SHell的发展，在大部分内容上与Bourne Shell兼容。

4．C Shell：是SUN公司Shell的BSD版本。

## 1.5 The File System

文件系统是由目录，子目录，以及文件为了特定的目的而组合在一起的一个体系结构。

目录：其他文件和目录的所在位置，能让你通过目录层次结构创建文件和目录。

子目录：在另一个目录下的任何目录，例如：在/目录下有/usr, /etc, and /kernel。

下图为solaris的文件系统：

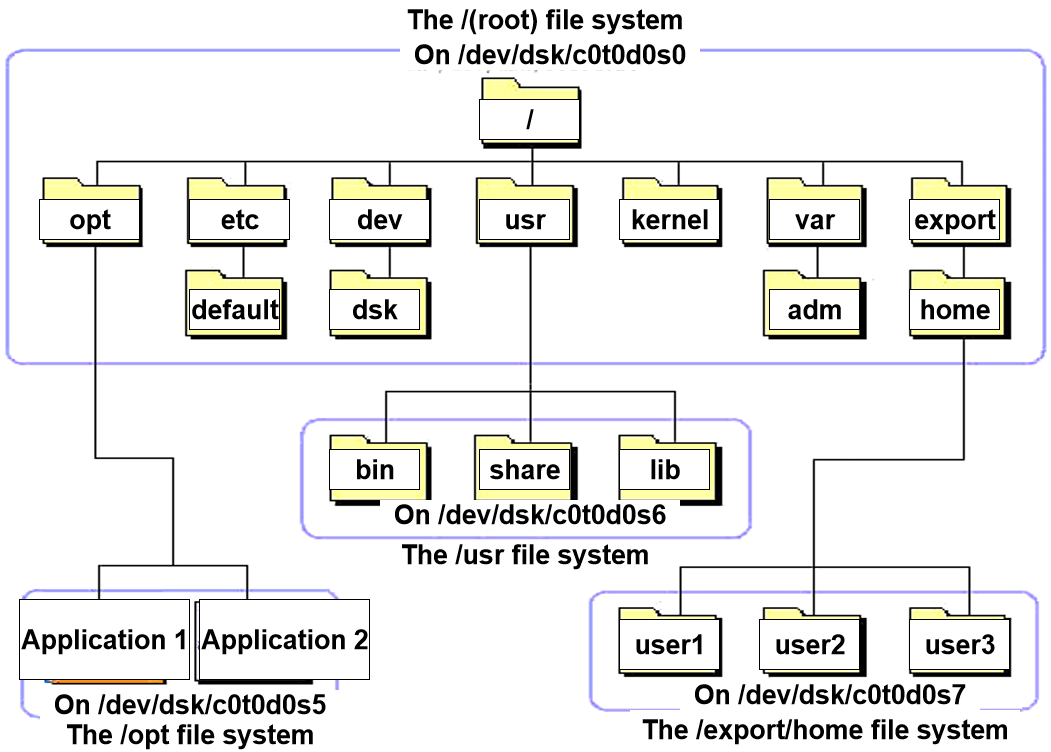


图1.2

Linux使用标准的目录结构，在安装的时候，安装程序就已经为用户创建了文件系统和完整而固定的目录组成形式，并指定了每个目录的作用和其中的文件类型。

/: 根路径

/opt:存放自己安装的第三方的软件,补丁

/etc:系统管理和配置文件,随着系统启动就会读取的文件，如passwd等cd

/dev:设备文件，该目录存放的是linux的外部设备，如硬盘，分区，键盘，鼠标，usb等。

/usr:一般存放系统自带的,但是供所有用户使用的。最庞大的目录，要用到的应用程序和文件几乎都在这个目录、

/usr/include: linux下开发和编译引用程序所需要的头文件。

/usr/local: 本地增加的命令和库。

/usr/sbin: 超级用户的一些管理程序

/usr/src：源代码

/usr/bin: 众多的而应用程序，二进制可执行程序

/usr/share:共享文件

/usr/lib:常用的动态链接库和软件包的配置文件。

/boot:存放内核程序(unix叫kernel),系统启动文件，包括Grub,lilo启动器程序。

/var: 存放管理员特有的，如系统日志

/home:用来存放用户的主目录(用户家目录)。在Linux中，每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的。

/bin: 存放系统中最常用的可执行文件 例如ls

/lib：存放共享的库文件，包含许多被/bin和/sbin中程序使用的库文件

/lost+found:这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些零散文件

/media:ubuntu系统自动挂载的光驱、usb设备，存放临时读入的文件

/mnt：作为被挂载的的文件系统得挂载点。

/proc:这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息，这里存放所有标志位文件的进程。

/root:该目录为系统管理员，也称作超级权限者的用户主目录。

/sbin:s就是Super User的意思，这里存放的是系统管理员使用的系统管理程序，如系统管理，目录查询等关键命令文件。

/srv:存放系统所提供的服务数据

/sys:系统设备和文件层次结构，并向用户程序提供详细的内核数据信息。

/tmp:这个目录用来存放一些临时的文件，所有用户对此目录都有读写权限。

## 1.6 Pathnames

路径名通过指定特定的文件或目录的位置唯一地标识该文件或目录。路径中每个目录名之间的/(斜杠)表示不同的目录，第一个位置的/表示根目录。路径分为以下两种：

1. 绝对路径：以/开头的路径,用来指明文件或目录与根目录之间的关系。
2. 相对路径：不以/开头的路径,指定一个文件和当前工作目录之间的关系。

**第二章 Basic Linux Commands**

## 1.Linux基础命令

**1.1命令行语法**

Command [options] [arguments]

注意：

1.命令一定使用空格进行分割

2.命令一定严格区分大小写。

3.单行命令一般不超过256个字符

**1.2 基本命令**

ctrl+alt+t 进入Ubuntu终端

exit 来退出终端

ctrl+shift+t是在terminal中打开多个标签

多个标签之间的切换ctrl + pageUp  ctrl + pageDown.

关闭触摸板：

sudo modprobe -r psmouse

打开触摸板：

sudo modprobe psmouse

ctrl -   字体缩小  
 ctrl +(有的电脑是“ ctrl shift  +  ”  三个键同时按下)字体放大  
 ctrl 0 恢复正常模式

ctrl+ d：关闭一个terminal快捷键

1.clear:清屏命令

2.pwd:显示当前工作路径的绝对路径名

3.修改密码：

命令：passwd

作用：修改当前或指定用户密码

例如：

(1)修改当前用户密码：

passwd

Changing password for briup.

(current) UNIX password:

Enter new UNIX password:

Retype new UNIX password:

(2)修改指定用户密码： passwd username

超级管理员 : root

新建的linux 超级管理员用户是没有密码的，需要先设置密码。

sudo passwd root

[sudo] password for briup: briup

输入新的 UNIX 密码： root

重新输入新的 UNIX 密码： root

passwd：已成功更新密码

注释:红色字体是自己输入的，输入密码的时候是不会显示的。

4.切换用户：

su - username: 切换到某个用户之下

su username: 还在当前目录之下（以当前系统身份切换）

切换到管理员: su root

注：刚创建的操作系统root用户是没有密码的，没有密码的root用户不能使用

设置密码：sudo passwd 如图2-1：

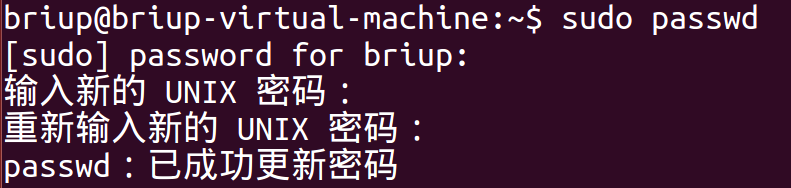


图2-1

5.cd:切换目录 cd pathname

. :当前路径

..:父路径(上级目录)

~：家目录

6.ls [options] pathname:列出文件和目录的列表

ls : 列出当前工作目录下的所有文件和目录列表（不包含隐藏文件）

ls -a: 显示所有文件包含隐藏文件。(.)

ls -R: 显示递归的列表

ls -t: 按照最后一次修改时间顺序显示文件和目录

ls -l: 显示文件或目录的长列表信息

显示格式：-rw-rw-r-- 1 briup briup 109 Jun 22 17:08 note.txt

(1).文件类型：

-：普通文件

d：目录文件

l：链接文件

s：套接字文件

p：管道文件

b：块设备文件

c：字符设备文件

1. .rw-rw-r--:代表文件权限

u g o

拥有者-生成文件时登录的人,权限最高, u 表示

同组人-系统管理员分配的同组的一个或几个人, g 表示

其它人-除拥有者,同组人外的人, o 表示

所有人-包括拥有者、同组人及其它人, a 表示

r：读权限 w：写权限 x：执行权限 -：没有权限

(3).链接号

(4).拥有者

(5).拥有组

(6).文件大小

(7).最后一次修改时间

(8).文件名

ls -F: 显示文件类型

目录: /

普通文件:

链接文件:@

可执行文件:\*

ls -d: 显示指定目录

ls -ld /opt: 显示/opt目录自身的长列表信息

ls -l /opt: 显示/opt目录下的所有文件和目录的长列表信息

7.chmod : 修改文件权限

(1). chmod mode filename :权限模式

如图2-2：

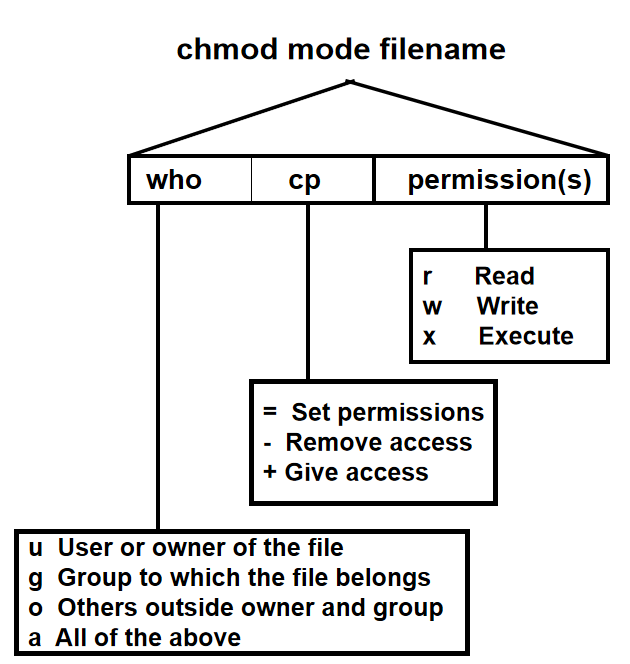


图2-2

例如：chmod o+w HelloWorld.java

(2).八进制数的形式标识:权限值模式

如图2-3：

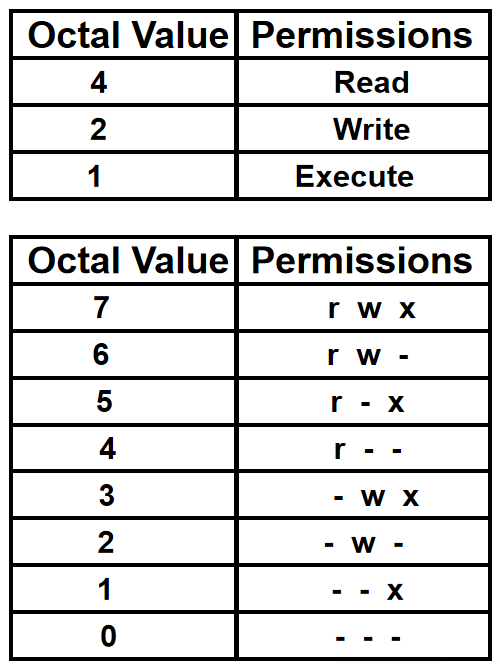


图2-3

例如：chmod 664 HelloWorld.java

注意:

对于文件 r表示可以查看文件内容 w表示可以向文件中写内容 x表示可以执行文件

对于目录 r表示可以查看目录中的内容 w表示可以在目录中创建内容 x表示可以进入目录内部

8.touch: 创建新的空的文件或者更新文件的最后一次访问时间

touch filename(s)

9.mkdir:创建目录

mkdir [options] pathname

-p : 加入父目录不存在，创建父目录

例如：mkdir dir1/dir2

mkdir dir3 dir4 dir5

mkdir -p dir6/dir7/dir8

10.cp : 拷贝文件

(1)源是文件

-i：提示是否覆盖

cp [-i] source file destination file:目标文件中的内容被源文件替换

cp [-i] source file destination directory:将源文件复制到目标目录中，源文件依旧存在

cp [-i] source file 不存在目标:不存在文件以不存在目标名字创建出来,内容是源文件的内容

(2)源是目录 必须加-r 代表进入目录

cp -r[i] source\_directory(s) destination\_directory:将源目录拷贝一份放入目标目录中

cp -r[i] source directory(s) 不存在:将源目录中所有的内容放入以不存在名字命名的目录中,而不是直接把源目录放进去。

11.mv 移动或者重命名文件或目录

(1)源是文件：

mv source file destination file: 将源文件内容复制到目标文件中，然后删除源文件

mv source file destination directory:将源文件复制到目标目录中，然后删除源文件

mv source file 不存在：覆盖目标文件内容，并且删除源文件或者可以理解为重命名(不存在名字重新命名源文件名字)

(2)源是目录：

mv source directory destination file:无法覆盖

mv source directory destination directory：将源目录复制到目标目录中，并删除源目录

mv source directory 不存在：重命名目录

12. rm : 删除文件或者非空目录

rm -r : 删除非空目录

rmdir : 删除空目录

rm -i ：删除的同时提示用户，让用户选择是否删除

13.man ：

man命令用来提供在线帮助，使用权限是所有用户。在Linux系统中存储着一部联机使用的手册，以供用户在终端上查找。使用man命令可以调阅其中的帮助信息，非常方便和实用。

常用指令：

spacebar(空格键) : 一次滚动一个屏幕

enter(回车) : 一次滚动一行(一行一行往下翻)

b:向前移动一屏幕

f:向后移动一屏幕

q:退出man命令

/string:查找匹配字符串

n:查找匹配字符串的下一次出现

14.网络工具

1. telnet

语法：telnet ip

作用：远程访问主机

2.退出： exit/logout

logout,用来退出当前登录的shell

exit：可以退出所有，一般用exit就可以

3. ping

语法：ping ip

作用：和windows操作一样，用来测试是否可以和某台主机连接

ctrl+c 退出

4. ifconfig

语法： ifconfig /ifconfig -a

作用：查看ip,和windows下的ipconfig类似

**第三章 Metacharacters**

**1.元字符**

**1.1元字符定义：**

指对当前shell具有特殊含义的字符

**1.2 常用元字符**

1.\* : 代表0-n个任意字符 例如：ls j\* 显示所有以j开头的文件或目录名字

2.？：代表任意一个字符 例如：ls j? 以j开头 后面有一位的文件或目录

3.; : 多个命令的分隔符 例如：cd;ls 先切换再显示列表；

date;cal;pwd 查看当前日期、查看当前日历和显示当前工作路径的绝对路径名

4.[] : 从[]指定的集合或者范围中选择一个字符

例如：[a-c]:aAbBc

[abc]:abc

ls[a-c]\*:查找以包含aAbBc字母开头的文件或目录

5.> : 重定向命令的输出到文件而不是屏幕（覆盖源文件）

例如：cal 10 2004 > a.txt 将2004年10月份的日历输出到a.txt文件中

6. >> : 将命令的输出追加到一个已经存在的文件

例如：cal 10 2004 >> a.txt 将2004年10月份的日历追加到a.txt文件中

7.| : 将前一个命令的输出作为下一个命令的输入

例如：ls -l /etc | more 使用more命令查看/etc目录下的内容

**2.Text Viewer**

1.cat

语法：cat filename(s)

作用：(1)将文本文件的内容一次性显示在屏幕上

(2)常用来将两个或者更多的文件整合成一个文件

例如cat a.txt: 将a.txt文件中的内容显示到屏幕上

cat a.txt b.txt > large.txt 将a.txt以及b.txt中的内容输出到large.txt

清空文件：

/dev/null 这个文件是linux中的黑洞,无论向这个文件中输入什么文件还是空

比如：cal > /dev/null 依旧是空

awk:处理文字程序，以行为单位来处理文字

-F指定分隔符

注意：-F和分隔符之间没有空格

'{print }':打印语句的固定格式

$1 "\t" $6

$1:分割后的第1个部分

$6:分割后的第6个部分

\t:制表符

\n:换行符

\:续行符

2.more

语法：more filenames

作用：一次一屏幕的显示文本文件内容

指令：

space bar : 一次滚动一屏幕

return : 一次向下一行。

b : 一次回退一屏幕

f : 一次向前一屏幕

h/? : 显示帮助信息文档

q : 退出，回到shell终端

/String : 向前查找字符串

n : 查找下一个字符串

head: -c 4,显示文件头的前4个字节，

-n 5， 显示前5行

tail: -c 4,显示文件头的后4个字节，

-n 5， 显示后5行

-F ：实时监控文件变更

**第四章 Full-screen Text Editor**

**1.满屏文本编辑器**

**1.1 vi介绍**

作用：满屏文本编辑器。

vi编辑器的三种模式：

1.底行模式

2.命令行模式

3.输入模式

三者之间的相互转换如图4-1：

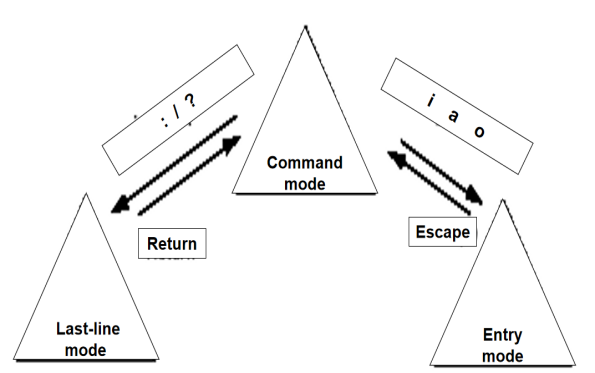


图4-1

模式转换：

(1)命令行模式-----》输入模式：

插入：

a： 在光标所在位置之后插入

A： 在光标所在行的行尾插入

i： 在光标所在位置插入

I： 在光标所在行的行首插入

o： 在光标所在行的下一行插入

O： 在光标所在行之上插入

(2)输入模式------》命令行模式

Esc

1. 命令行模式-----》底行模式

ESC+:or / or ?

**1.2 打开vi：**

vi ： 打开一个新的文件，退出或者保存时命名

vi filename : 打开或者创建一个新的文件

**1.3 保存文件：**

底行模式下：

:w newFile : 存成新文件

:w : 存盘

**1.4 退出vi：**

底行模式下：

:wq : 保存退出

:q! : 强行退出不存盘

**1.5删除文本：**

x: 删除一个字符

s: 删除当前字符，并进入输入模式

dw： 删除当前词

ndw: 删除n个词

dd: 删除当前行

ndd: 删除之后的n行

:5,10d: 删除5-10行之间的内容

**1.6修改文本：**

r ： 替换一个字符，

cw ：替换一个单词， 进入输入模式

cc ：替换一行， 进入输入模式

C ：替换从光标至行尾。 进入输入模式

**1.7复制和粘贴文本：**

yw : 拷贝词

yy ： 拷贝行

p : 当前行下粘贴

:1,2co3: 拷贝行1，行2在行3之后

:4,5m6: 移动行4，行5在行6之后。

**1.8高级功能：**

~ ： 改变大小写

J ： 把当前行和下一行连起来

u ： 废除刚才的编辑操作

:set nu : 显示行标

:21 光标停在指定行

21G 光标停在指定行

/str 从当前行往下查找

?str 从当前行往上查找

n 继续查找

:r file 在光标所在位置插入另一个文件

:1,$s/old/new/g 替换文本(%s/old/new/g)

1. **Setup Initialization Files**

**1.配置文件**

**1.1定义**

系统配置文件用来记录一些系统运行时需要的相关配置信息，环境变量信息等。

**1.2种类**

1.系统级别：/etc/profile

一般存放在/etc目录下

例如:

（1）/etc/enviroment 是系统的环境变量。

（2）/etc/profile： 是所有用户的环境变量。当用户第一次登录时,该文件被执行.并从/etc/profile.d目录的配置文件中搜集shell的设置。

（3）/etc/bashrc: 为每一个运行bash shell的用户执行此文件.当bash shell被打开时,该文件被读取。

2.用户级别:~/.bashrc

一般存在用户家目录之下

例如：

（1）~/.bash\_profile: 每个用户都可使用该文件输入专用于自己使用的shell信息,当用户登录时,该文件仅仅执行一次!默认情况下,他设置一些环境变量,执行用户的.bashrc文件。

（2）~/.bashrc: 该文件包含专用于你的bash shell的bash信息,当登录时以及每次打开新的shell时,该该文件被读取。

（3） ~/.bash\_logout:当每次退出系统(退出bash shell)时,执行该文件. 另外,/etc/profile中设定的变量(全局)的可以作用于任何用户,而~/.bashrc等中设定的变量(局部)只能继承 /etc/profile中的变量,他们是”父子”关系。

（4）~/.bash\_profile 是交互式、login 方式进入 bash 运行的~/.bashrc 是交互式 non-login 方式进入 bash 运行的通常二者设置大致相同，所以通常前者会调用后者。

**1.3内容**

1.设置默认的创建权限

2.设置环境变量

3.设置默认终端提示符

4.设置默认命令行命令的路径

5.设置客户自定义命令

**1.4相关指令**

1.设置环境变量

(1)在Shell中定义临时变量：

varName=value

1. 定义用户级别的环境变量

在用户级别的配置文件中:常用的~/.bashrc

export varName=value

让文件执行：source ~/.bashrc

(3)定义系统级别的环境变量

在系统级别的配置文件中：常用的/etc/profile

export varName=value

2.读取变量的值：

echo $varName

3.path变量：

path环境变量用来指明操作系统下常用命令的搜索路径

4.which

语法: which commandName

作用：在系统环境变量中查找指定命令的路径名

which是根据使用者所配置的 PATH 变量内的目录去搜寻可运行文档的！所以，不同的 PATH 配置内容所找到的命令可能不一样！

5.whereis

语法：whereis commandName

作用：whereis命令只能用于程序名的搜索，而且只搜索二进制文件（参数-b）、man说明文件（参数-m）和源代码文件（参数-s）。如果省略参数，则返回所有信息。 whereis查找的速度非常快，这是因为linux系统会将 系统内的所有文件都记录在一个数据库文件中，当使用whereis时，会从数据库中查找数据，而不是通过遍历硬盘来查找，效率自然会很高。但是该数据库文件并不是实时更新，默认情况下时一星期更新一次，因此，我们在用whereis时，有时会找到已经被删除的数据，或者刚刚建立文件，却无法查找到，原因就是因为数据库文件没有被更新。

6.history

显示命令行历史记录

history 10 显示最近的10条记录

!n 执行第n条历史命令

!! 执行刚刚执行过的历史命令

7.alias

语法：alias 别名=命令

作用：用来给一个长的命令，起一个简单的别名

例如：

给history起个别名h

alias h=history

查看所有的别名

alias

取消绑定

unalias h

8.umask

跟系统创建的目录和文件的默认权限相关。

这个值决定了当前用户所创建文件和目录的默认权限

目录权限： 777 - umask

文件权限： 777 - umask - 所有人的执行权限

9.各个不同后缀的含义如下：

.tar打包格式：tar程序打包的数据，并没有压缩过

.gz压缩格式：gzip程序压缩的文件

.tar.gz打包压缩：tar程序打包的文件，其中并且经过 gzip 的压缩

.tar

压缩：tar cvf FileName.tar FileName

解压：tar xvf FileName.tar

.gz

解压：tar zxvf FileName.tar.gz -C ./d

压缩：tar zcvf FileName.tar.gz DirName

-C把文件解压到 特定目录中

# ,配置虚拟机的共享文件夹

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 地图, 文字

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 截图, 地图

描述已自动生成

共享文件夹完成，window下有gx文件夹，如果哪个 文件想要放到linux中，就把这个文件拷贝到gx文件夹中

测试共享文件夹:

window中:放了a.txt到gx文件夹中

图片包含 截图, 监控, 游戏机, 屏幕

描述已自动生成

开启linux操作系统:

打开命令行 : Ctrl+Alt+T

briup@briup-virtual-machine:~$ cd /

briup@briup-virtual-machine:/$ cd mnt/

briup@briup-virtual-machine:/mnt$ ls

hgfs

briup@briup-virtual-machine:/mnt$ cd hgfs

briup@briup-virtual-machine:/mnt/hgfs$ ls

gx

briup@briup-virtual-machine:/mnt/hgfs$ cd gx

briup@briup-virtual-machine:/mnt/hgfs/gx$ ls

a.txt

把jdk安装包，整到 /home/briup下

在/home/briup下构建文件夹soft

使用命令 tar -xvf ./jdk-8u172-linux-x64.gz -C soft

然后改名字为jdk

图片包含 截图, 游戏机

描述已自动生成

配置环境变量:

系统级别

/etc/profile

用户级别:

/home/用户名/.bashrc

新开控制台:

vi ./.bashrc

在最后输入:

export JAVA\_HOME=/home/briup/soft/jdk

export CLASSPATH=.

export PATH=$PATH:$JAVA\_HOME/bin

如下图

图片包含 游戏机, 截图

描述已自动生成

保存退出:

先按ESC 再Shift+: 完事再输入!wq 回车。

退出vi以后，要输入 source .bashrc

让配置信息立刻生效

# 配置hadoop:

1,把共享文件夹中的 hadoop.gz复制到家目录中

图片包含 游戏机

描述已自动生成

2,解压hadoop.gz

切换到~家目录下(hadoop.gz在这个目录下)

tar -xvf ./hadoop-3.0.3.tar.gz -C soft

使用命令修改解压后的名字

briup@briup-virtual-machine:~$ ls

aa.txt hadoop-3.0.3.tar.gz 公共的 图片 音乐

a.txt jdk-8u172-linux-x64.gz 模板 文档 桌面

examples.desktop soft 视频 下载

briup@briup-virtual-machine:~$ mv soft/hadoop-3.0.3/ soft/hadoop

现在状态

在soft/hadoop/里面有很多文件

# 解压完成以后，搭建hadoop:

## 1,机器名:改为hadoopPD

先修改hostname的权限:

briup@briup-virtual-machine:~$ ls -al /etc/hostname

-rw-r--r-- 1 root root 22 5月 9 2016 /etc/hostname

briup@briup-virtual-machine:~$ sudo chmod 777 /etc/hostname

[sudo] password for briup: briup

briup@briup-virtual-machine:~$ ls -al /etc/hostname

-rwxrwxrwx 1 root root 22 5月 9 2016 /etc/hostname

权限已满

Vi /etc/hostname

图片包含 游戏机, 截图

描述已自动生成

把hostname的内容改为hadoopPD

按ESC 然后 Shift+: 再输入wq!回车。结束。

改好了。

检查:

输入hostname

briup@briup-virtual-machine:~$ hostname

hadoopPD

## 2,修改ip和机器名映射

先给/etc/hosts文件授予权限:

sudo chmod 777 /etc/hosts

briup@briup-virtual-machine:~$ sudo chmod 777 /etc/hosts

briup@briup-virtual-machine:~$ ls -al /etc/hosts

-rwxrwxrwx 1 root root 236 5月 9 2016 /etc/hosts

修改映射关系

vi /etc/hosts

127.0.0.1 localhost

#127.0.1.1 briup-virtual-machine

# The following lines are desirable for IPv6 capable hosts

::1 ip6-localhost ip6-loopback

fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

127.0.0.1 hadoopPD

保存退出:

按ESC 然后 Shift+: 再输入wq!回车。结束。

改好了。

检查:

briup@briup-virtual-machine:~$ ping hadoopPD

PING hadoopPD (127.0.0.1) 56(84) bytes of data.

64 bytes from localhost (127.0.0.1): icmp\_seq=1 ttl=64 time=0.010 ms

64 bytes from localhost (127.0.0.1): icmp\_seq=2 ttl=64 time=0.021 ms

64 bytes from localhost (127.0.0.1): icmp\_seq=3 ttl=64 time=0.205 ms

64 bytes from localhost (127.0.0.1): icmp\_seq=4 ttl=64 time=0.044 ms

^C就可以一直按Ctrl+C 停掉 ping

配置环境变量:

修改家目录下的 .bashrc文件

下面内容追加

excute HADOOP\_HOME=/home/briup/soft/hadoop

excute HADOOP\_USER\_NAME=hadoopPD

在path后面追加

excute PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sin

Eg:

briup@briup-virtual-machine:~$ vi ~/.bashrc

最后编写上:

export JAVA\_HOME=/home/briup/soft/jdk

export CLASSPATH=.

export HADOOP\_HOME=/home/briup/soft/hadoop

export HADOOP\_USER\_NAME=hadoopPD

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

保存退出:

按ESC 然后 Shift+: 再输入wq!回车。结束。改好了。

让配置文件立刻生效:

Source .bashrc

测试:

直接输入hadoop回车，有反应基本就对了。

修改配置文件:

Hadoop-evn.sh:

添加JAVA\_HOME

briup@briup-virtual-machine:~/soft/hadoop/etc/hadoop$ vi hadoop-env.sh

在37行 修改

37 JAVA\_HOME=/home/briup/soft/jdk

在插入模式，按esc 再shitf+: 然后输入set nu 回车。

保存退出

core-site.xml

briup@briup-virtual-machine:~/soft/hadoop/etc/hadoop$ vi core-site.xml

在configuration标签中添加

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoopPD:9000</value>

</property>

保存退出

hdfs-site.xml

vi hdfs-site.xml

briup@briup-virtual-machine:~/soft/hadoop/etc/hadoop$ vi hdfs-site.xml

在configuration标签中添加

<!--hdfs的备份因子，1就是只有一个备份。也就是只有一份-->

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<!-- hdfs集群的名称 -->

<property>

<name>dfs.nameservices</name>

<!-- hdfs集群的名称可以随意取 -->

<value>hadoop-cluster</value>

</property>

<!-- 配置nameNode第二节点的位置-->

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>hadoopPD:50090</value>

</property>

<!-- hdfs集群web访问的地址,

这句不配置访问的路径就是localhost:9870-->

<!--<property>

<name>dfs.http.address</name>

<value>hadoopPD:9870</value>

</property>-->

<!-- hdfs集群上 块的大小 -->

<property>

<name>dfs.blocksize</name>

<value>32m</value>

</property>

<!-- hdfs集群的namenode(主节点) 运行过程中的中间数据存放位置 -->

<property>

<name>dfs.namenode.name.dir</name>

<value>/home/briup/soft/hadoop/da/namenode</value>

</property>

<!-- hdfs集群的datanode(从节点) 数据存放位置 -->

<property>

<name>dfs.datanode.data.dir</name>

<value>/home/briup/soft/hadoop/da/datanode</value>

</property>

mapred-site.xml

vi mapred-site.xml

在configuration标签中添加

<!-- mapReduce使用什么运行 -->

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

yarn-site.xml

vi yarn-site.xml

<!-- yarn集群的 resource manager 地址，主机名 -->

<property>

<name>yarn.resourcemanager.hostname</name>

<!-- 这个hadoopPD是计算机名 cmd中hostname可以查看到 -->

<value>hadoopPD</value>

</property>

<!-- 指定resourceManager的网页地址 -->

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>localhost:8088</value>

</property>

<property>

<name>yarn.nodemanager.local-dirs.hostname</name>

<value>/home/briup/soft/hadoop/da/namenode</value>

</property>

<!-- MapReduce运行的模式是什么，混洗 -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.nodemanager.log-dirs</name>

<value>/home/briup/soft/hadoop/da/yarn/logs</value>

</property>

创建上面用到的文件夹:

cd /home/briup/soft/hadoop

briup@briup-virtual-machine:~/soft/hadoop/etc/hadoop$ cd /home/briup/soft/hadoop

briup@briup-virtual-machine:~/soft/hadoop$ ls

bin include libexec NOTICE.txt sbin

etc lib LICENSE.txt README.txt share

briup@briup-virtual-machine:~/soft/hadoop$ mkdir -p ./da/yarn/logs ./da/namenode ./da/datanode

初始化hdfs文件系统:

hdfs namenode -format

启动namenode

hadoop-daemon.sh start namenode

启动datanode

hadoop-daemon.sh start datanode

输入jps查看

briup@hadoopPD:~$ jps

8481 NameNode

8956 DataNode

5886 ResourceManager

9295 Jps

有namenode 和 datanode 就成功

输入ifconfig查看linux的ip地址

在window中浏览器输入 上面看到的ip:9870然后看到hdfs管理界面即可。