

1 Linux基础知识

Linux的优点:

1. 免费
2. 很多软件原生是在Linux下运行的，庞大的社区支持，生态环境好
3. 开源，可被定制，开放，多用户的网络操作系统
4. 相对安全稳定

Linux系统的组成:

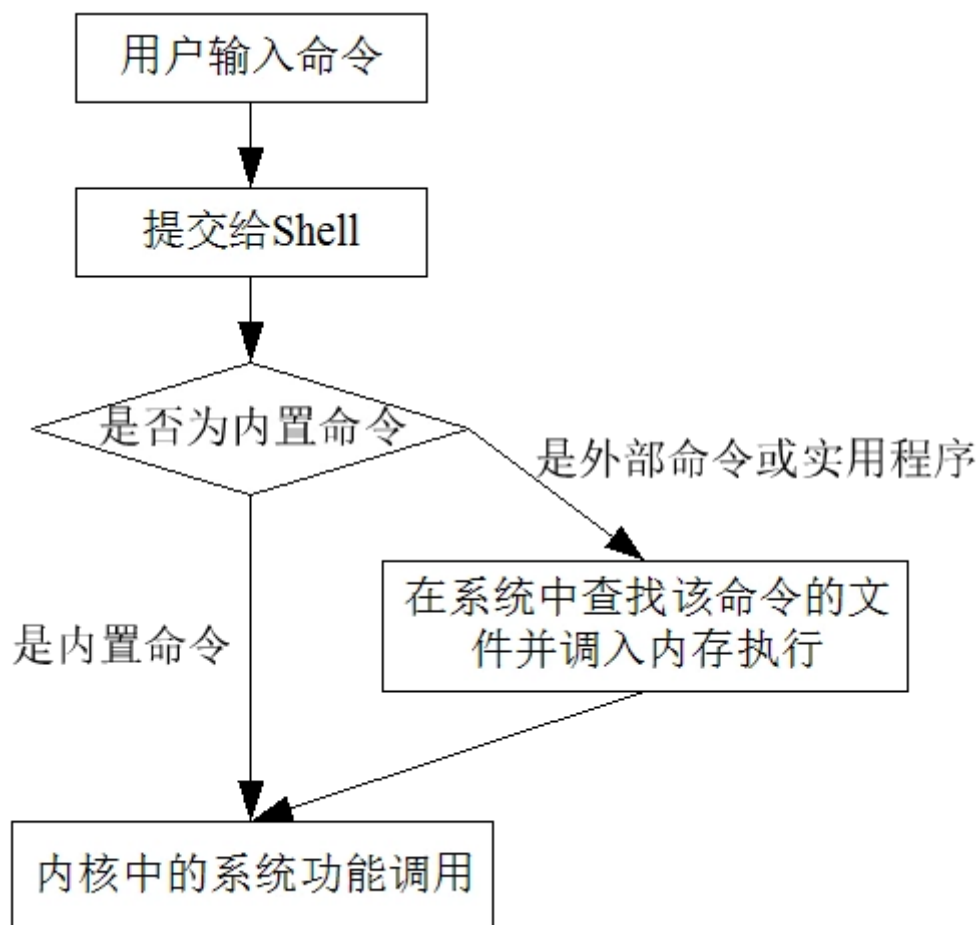
1. linux内核 (linus 团队管理)
2. shell: 用户与内核交互的接口
3. 文件系统: ext3、ext4等。windows 有 fat32、ntfs
4. 第三方应用软件

1.1 Shell的基本知识

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(命令解释器)

Shell可以执行:

- 内部命令
- 应用程序
- shell脚本



使用 `type` 命令可以区分内部命令和外部命令

Shell可以实现的功能:

- 命令行解释(这是用得最多的)
- 命令的多种执行顺序
- 通配符 (wild-card characters)
- 命令补全、别名机制、命令历史
- I/O重定向 (Input/output redirection)
- 管道 (pipes)
- 命令替换 (或\$())
- Shell编程语言 (Shell Script)

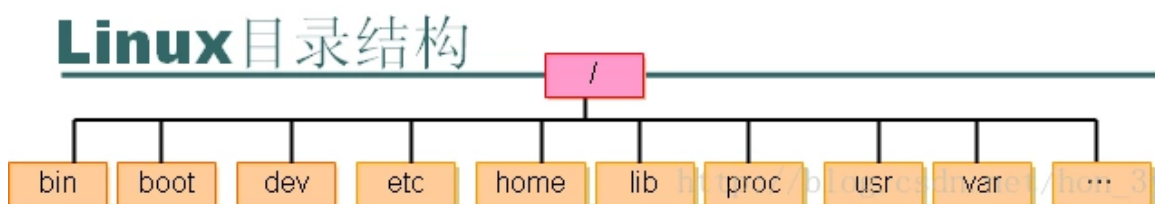
Shell的主要版本有:

Bash (Bourne Again Shell)	bash 是大多数Linux系统的默认Shell。 bash 与 bsh 完全向后兼容, 并且在 bsh 的基础上增加和增强了很多特性。 bash 也包含了很多 C Shell 和 Korn Shell 中的优点。 bash 有很灵活和强大的编程接口, 同时又有很友好的用户界面
Ksh (Korn Shell)	Korn Shell (ksh) 由Dave Korn所写。它是UNIX系统上的标准Shell。 在Linux环境下有一个专门为Linux系统编写的Korn Shell的扩展版本, 即Public Domain Korn Shell (pdksh)。
tcsh (csh 的扩展)	tcsh 是 C Shell 的扩展。 tcsh 与 csh 完全向后兼容, 但它包含了更多的使用户感觉方便的新特性, 其最大的提高是在命令行编辑和历史浏览方面

使用 ps 命令观察正在执行的shell.

1.2 Linux目录结构

Linux基本的目录结构的:



Linux 文件系统是一个目录树的结构, 文件系统结构从一个根目录开始, 根目录下可以有任意多个文件和子目录, 子目录中又可以有任意多个文件和子目录。

- bin 存放二进制可执行文件(ls,cat,mkdir等)
- boot 存放用于系统引导时使用的各种文件
- dev 用于存放设备文件
- etc 存放系统配置文件
- home 存放所有用户文件的根目录
- lib 存放跟文件系统中的程序运行所需要的共享库及内核模块
- mnt 系统管理员安装临时文件系统的安装点
- opt 额外安装的可选应用程序包所放置的位置
- proc 虚拟文件系统, 存放当前内存的映射
- root 超级用户目录

- `sbin` 存放二进制可执行文件，只有root才能访问
- `tmp` 用于存放各种临时文件
- `usr` 用于存放系统应用程序，比较重要的目录`/usr/local` 本地管理员软件安装目录
- `var` 用于存放运行时需要改变数据的文件

1.3 命令基本格式

`cmd [options] [arguments]`，`options`称为选项，`arguments`称为参数

选项和参数都作为Shell命令执行时的输入，它们之间用空格分隔开。

Linux是区分大小写的。

一般来说，后面跟的选项如果单字符选项前使用一个减号`-`。单词选项前使用两个减号`--`

- 例子：`ls -a`和`ls --all`，`a` 单个字符使用一个`-`，一个单词`all` 使用两个`--`

在Linux中，可执行的文件也进行了分类：

- **内置命令**：出于效率的考虑，将一些常用命令的解释程序构造在Shell内部。
- **外置命令**：存放在`/bin`、`/sbin`目录下的命令
- **实用程序**：存放在`/usr/bin`、`/usr/sbin`、`/usr/share`、`/usr/local/bin`等目录下的实用程序
- **用户程序**：用户程序经过编译生成可执行文件后，可作为Shell命令运行
- **Shell脚本**：由Shell语言编写的批处理文件，可作为Shell命令运行

1.4 通配符

类似于正则表达式中的通配符，Linux的通配符多用在搜索的时候：

- *****：匹配任何字符和任何数目的字符
- **?**：匹配单一数目的任何字符
- **[]**：匹配[]之内的任意一个字符
- **[!]**：匹配除了[!]之外的任意一个字符，`!`表示非的意思

1.5 文件的类型

- 普通文件 `-`
- 目录 `d`
- 符号链接 `l`
 - 硬链接：与普通文件没什么不同，`inode` 都指向同一个文件在硬盘中的区块
 - 软链接：保存了其代表的文件的绝对路径，是另外一种文件，在硬盘上有独立的区块，访问时替换自身路径(简单地理解为 Windows 中常见的快捷方式)。
- 字符设备文件 `c`
- 块设备文件 `b`
- 套接字 `s`
- 命名管道 `p`

1.6 用户主目录

Linux是多用户的网络系统！所以可以在Linux下创建多个用户，每个用户都会有自己专属的空间。

所以，在创建用户时，系统管理员会给每个用户建立一个主目录，通常在 `/home/` 目录下

比如：用户osmond的主目录为：`/home/osmond`

用户对自己主目录的文件拥有所有权，可以在自己的主目录下进行相关操作。

2 常用的命令

2.1 常用的文件、目录操作命令

这是使用得最多的命令，Linux最基础的命令！

- 可用 `pwd` 命令查看用户的当前目录
- 可用 `cd` 命令来切换目录
- `.` 表示当前目录
- `..` 表示当前目录的上一级目录（父目录）
- `-` 表示用 `cd` 命令切换目录前所在的目录
- `~` 表示用户主目录的绝对路径名

绝对路径：以斜线（/）开头，描述到文件位置的完整说明，任何时候你想指定文件名的时候都可以使用

相对路径：不以斜线（/）开头，指定相对于你的当前工作目录而言的位置，可以被用作指定文件名的简捷方式

tips：输入命令的时候要常用tab键来补全

- `ls`：显示文件或目录信息
- `mkdir`：当前目录下创建一个空目录
- `rmdir`：要求目录为空
- `touch`：生成一个空文件或更改文件的时间
- `cp`：复制文件或目录
- `mv`：移动文件或目录、文件或目录改名
- `rm`：删除文件或目录
- `ln`：建立链接文件
- `find`：查找文件
- `file/stat`：查看文件类型或文件属性信息
- `cat`：查看文本文件内容
- `more`：可以分页看
- `less`：不仅可以分页，还可以方便地搜索，回翻等操作
- `tail -10`：查看文件的尾部的10行
- `head -20`：查看文件的头部20行
- `echo`：把内容重定向到指定的文件中，有则打开，无则创建
- 管道命令 `|`：将前面的结果给后面的命令，例如：`ls -la | wc`，将ls的结果加油wc命令来统计字数
- 重定向 `>` 是覆盖模式，`>>` 是追加模式，例如：`echo "Java3y,zhen de hen xihuan ni" > qingshu.txt` 把左边的输出放到右边的文件里去

练习题

- Linux的shell程序默认是 `bash` 程序；
- Linux命令格式包含三个部分，分别是：命令、选项、参数；
- Linux命令选项前为单个减号（-），后面一般为单字符，选项前为双减号（--），后面一般为单词；
- Linux命令中使用的通配符有 `?` `*` `[]`；

- 命令 `ls /usr/bin/w*` 的效果是 列出指定目录下的所有以w开头的文件或目录；
- 命令 `ls /usr/bin/w??` 的效果是 列出指定目录下的以w开头名称长度为3的所有文件或目录；
- 命令 `ls /usr/bin/[xyz]*` 的效果是 列出指定目录下的文件名以x或y或z开头的文件或目录；
- 命令 `ls /usr/bin/[!a-h]*` 的效果是 列出指定目录下的文件名不以a到h区间字母开头的文件或目录；
- 目录操作时，“.”表示 当前目录；
- 目录操作时，“..”表示 上一级目录；
- 目录操作时，“-”表示 上一次工作目录；
- 目录操作时，“~”表示 用户主目录；
- 命令ln可以建立文件链接，这种链接分为：硬链接 和 软链接；
- 命令touch可以改变文件的三种时间，分别是：access time、modify time、change time；

>和>> 有何区别？

答：>是覆盖方式重定向到新的文件；>> 是以补充方式，添加到原文件的末尾。

通过管道和more命令，浏览文件列表方便在哪里？

答：对于文件列表超出一个屏幕显示的情况，这种方式可以分屏浏览，比较方便。

2.2 文件打包和压缩命令

命令	功能
xz	使用LZMA 算法的高性能压缩/解压工具
gzip	流行的 GNU gzip 数据压缩/解压程序
bzip2	免费的，无专利的高性能数据压缩工具
zip/unzip	与WinZIP兼容的压缩/解压工具
rar	与WinRAR兼容的压缩/解压工具
7za	使用LZMA 算法的高性能压缩/解压工具
tar	文件打包、归档工具

文件后缀	说明
.bz2	用 bzip2 压缩的文件
.gz	用 gzip 压缩的文件
.xz	用 xz 压缩的文件
.tar	用 tar 打包的文件，也称 tar 文件
.tbz	tar 打包时用 bzip2 压缩的文件
.tgz	tar 打包时用 gzip 压缩的文件
.zip	用 zip/winzip 压缩的文件
.rar	用 rar 压缩的文件
.7z	用 7za 压缩的文件

压缩的方式常用的有：

- gzip
- bzip2
- tar

常用的压缩的命令就有：

- `gzip filename`
- `bzip2 filename`
- `tar -czvf filename`

常用的解压命令有：

- `gzip -d filename.gz`
- `bzip2 -d filename.bz2`
- `tar -xzvf filename.tar.gz`

2.3 正则表达式+grep

如何快速查看某个文本文件下的某些关键字是否存在？

1 正则表达式

正则表达式的元字符

元字符	含义	类型	举例	说明
^	匹配首字符	BRE	^x	以字符x开始的字符串
\$	匹配尾字符	BRE	x\$	以x字符结尾的字符串
.	匹配任意一个字符	BRE	l..e	love, life, live ...
?	匹配任意一个可选字符	ERE	xy?	x, xy
*	匹配零次或多次重复	BRE	xy*	x, xy, xyy, xyyy ...
+	匹配一次或多次重复	ERE	xy+	xy, xyy, xyyy ...
[...]	匹配任意一个字符	BRE	[xyz]	x, y, z
()	对正则表达式分组	ERE	(xy)+	xy, xyxy, xyxyxy, ...

元字符	含义	类型	举例	说明
\{n\}	匹配n次	BRE	go\{2\}gle	google
\{n,\}	匹配最少n次	BRE	go\{2,\}gle	google, gooogole, goooogle ...
\{n,m\}	匹配n到m次	BRE	go\{2,4\}gle	google, gooogole, goooogle
{n}	匹配n次	ERE	go{2}gle	google
{n,}	匹配最少n次	ERE	go{2,}gle	google, gooogole, goooogle ...
{n,m}	匹配n到m次	ERE	go{2,4}gle	google, gooogole, goooogle
	以或逻辑连接多个匹配	ERE	good bon	匹配 good 或 bon
\	转义字符	BRE	*	*

2 grep命令

grep是一个强大的文本搜索工具。grep使用正则表达式搜索文本，并把匹配的行打印出来。

格式： `grep [options] PATTERN [FILE...]`

- PATTERN 是查找条件：可以是普通字符串、可以是正则表达式，通常用单引号将RE括起来。
- FILE 是要查找的文件，可以用空格间隔的多个文件，也可使用Shell的通配符在多个文件中查找PATTERN，省略时表示在标准输入中查找。
- grep命令不会对输入文件进行任何修改或影响，可以使用输出重定向将结果存为文件

例子：

- 在文件 myfile 中查找包含字符串 mystr的行
 - `grep -n mystr myfile`
- 显示 myfile 中第一个字符为字母的所有行
 - `grep '^[a-zA-Z]' myfile`

- 在文件 myfile 中查找首字符不是 # 的行（即过滤掉注释行）
 - `grep -v '^#' myfile`
- 列出/etc目录（包括子目录）下所有文件内容中包含字符串“root”的文件名
 - `grep -lr root /etc/*`

2.4 提高效率的方法

- 自动补全命令：TAB键
- 历史命令：上下箭头键
- 别名alias

2.5 Shell变量和Shell环境

Shell 变量大致可以分为三类：

- **内部变量**：由系统提供，用户只能使用不能修改。
 - ?
 - GROUPS
- **环境变量**：这些变量决定了用户工作的环境，它们不需要用户去定义，可以直接在 shell 中使用，其中某些变量用户可以修改。
- **用户变量**：由用户建立和修改，在 shell 脚本编写中会经常用到。
 - 变量赋值（定义变量）
 - `varName=Value`
 - `export varName=Value`
 - 引用变量 `$varName`

Shell变量的作用域：

- **局部变量**的作用范围仅仅限制在其命令行所在的Shell或Shell脚本文件中；
- **全局变量**的作用范围则包括本Shell进程及其所有子进程。
- **局部变量与全局变量互换**：可以使用 `export` 内置命令将局部变量设置为全局变量。可以使用 `export` 内置命令将全局变量设置为局部变量。

export命令：

- 显示当前Shell可见的全局变量
 - `export [-p]`
- 定义变量值的同时声明为全局变量。
 - `export <变量名1=值1> [<变量名2=值2> ...]`
- 声明已经赋值的某个（些）局部变量为全局变量。
 - `export <变量名1> [<变量名2> ...]`
- 声明已经赋值的某个（些）全局变量为局部变量。
 - `export -n <变量名1> [<变量名2> ...]`

Shell环境变量：

- 环境变量定义 Shell 的运行环境，保证 Shell 命令的正确执行。
- Shell用环境变量来确定查找路径、注册目录、终端类型、终端名称、用户名等。
- **所有环境变量都是全局变量（即可以传递给 Shell 的子进程），并可以由用户重新设置。**

变量名	含义
HOME	用户主目录
LOGNAME	登录名
USER	用户名，与登录名相同
PWD	当前目录/工作目录名
MAIL	用户的邮箱路径名
HOSTNAME	计算机的主机名
INPUTRC	默认的键盘映像
SHELL	用户所使用的 <i>shell</i> 的路径名
LANG	默认语言
HISTSIZE	<i>history</i> 所能记住的命令的最多个数
PATH	<i>shell</i> 查找用户输入命令的路径 (目录列表)
PS1、PS2	<i>shell</i> 一级、二级命令提示符

Shell变量：查询、显示和取消：

- 显示当前已经定义的所有变量
 - 所有环境变量： `env`
 - 所有变量和函数（包括环境变量）： `set`
- 显示某（些）个变量的值
 - `echo $NAME1 [$NAME2]`
- 取消变量的声明或赋值
 - `unset <NAME>`

Shell环境变量的值是否可以修改？为什么？

答：环境变量的值一般情况下，可以修改。但一定要慎重修改，因为一旦修改错误，对*shell*正常运行造成严重影响，甚至导致*shell*无法运行。

3 VI编辑器

vi 可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。

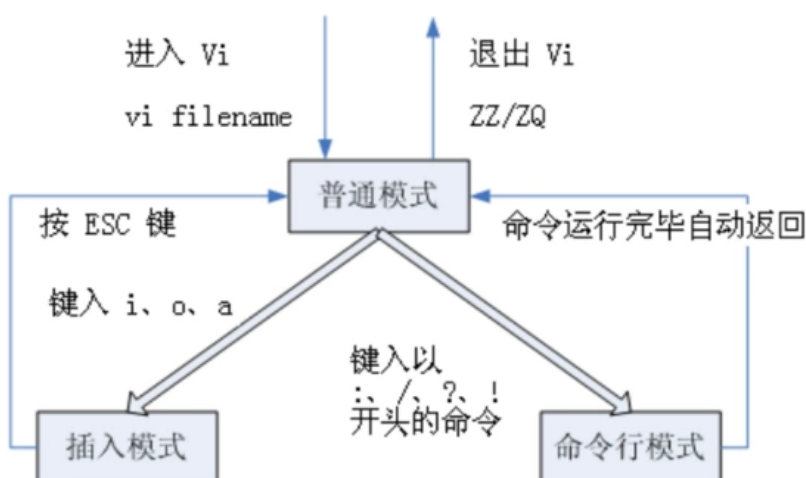
- *vi*可以看做成我们Windows下的记事本
- *vim* 即 *Vi* IMproved，*vi* 克隆版本之一

命令	说明
vi	直接进入
vi filename	打开或新建文件filename，并将光标置于第一行首
vi +n filename	打开文件filename，并将光标置于第n行首
vi + filename	打开文件filename，并将光标置于最后一行首
vi +/pattern filename	打开文件filename，并将光标置于第一个与pattern匹配的串处
vi -r filename	打开上次用vi编辑时发生系统崩溃，恢复filename

Vi有三种模式：

Vi 的3种运行模式

- 普通
(normal)模式
- 插入(insert)模式
- 命令行
(Cmdline)模式



3.1 普通模式

- G 用于直接跳转到文件尾
- ZZ 用于存盘退出Vi
- ZQ 用于不存盘退出Vi
- /和? 用于查找字符串
- n 继续查找下一个
- yy 复制一行
- p 粘贴在下一行，P粘贴在前一行
- dd 删除一行文本
- x 删除光标所在的字符
- u 取消上一次编辑操作 (undo)

3.2 插入模式

在 Normal 模式下输入插入命令 `i`、`a`、`o` 进入 insert 模式。用户输入的任何字符都被 vim 当做文件内容保存起来，并将其显示在屏幕上。

- 在文本输入过程中，若想回到 Normal 模式下，按 `Esc` 键即可。

3.3 命令行模式

Normal 模式下，用户按冒号 `:` 即可进入 Command 模式，此时 vim 会在显示窗口的最后一行 (屏幕的最后一行) 显示一个 “:” 作为 Command 模式的提示符，等待输入命令。

- `:w` 保存当前编辑文件，但并不退出
- `:w newfile` 存为另外一个名为 “newfile” 的文件
- `:wq` 用于存盘退出 Vi
- `:q!` 用于不存盘退出 Vi
- `:q` 用于直接退出 Vi (未做修改)

设置 Vi 环境:

- `:set autoindent` 缩进,常用于程序的编写
- `:set noautoindent` 取消缩进
- `:set number` 在编辑文件时显示行号
- `:set nonumber` 不显示行号
- `:set tabstop=value` 设置显示制表符的空格字符个数
- `:set` 显示设置的所有选项
- `:set all` 显示所有可以设置的选项