

**RED HAT®  
TRAINING**



# Red Hat

**RH124 红帽系统管理 I**

**RH124-02-从命令行管理文件**



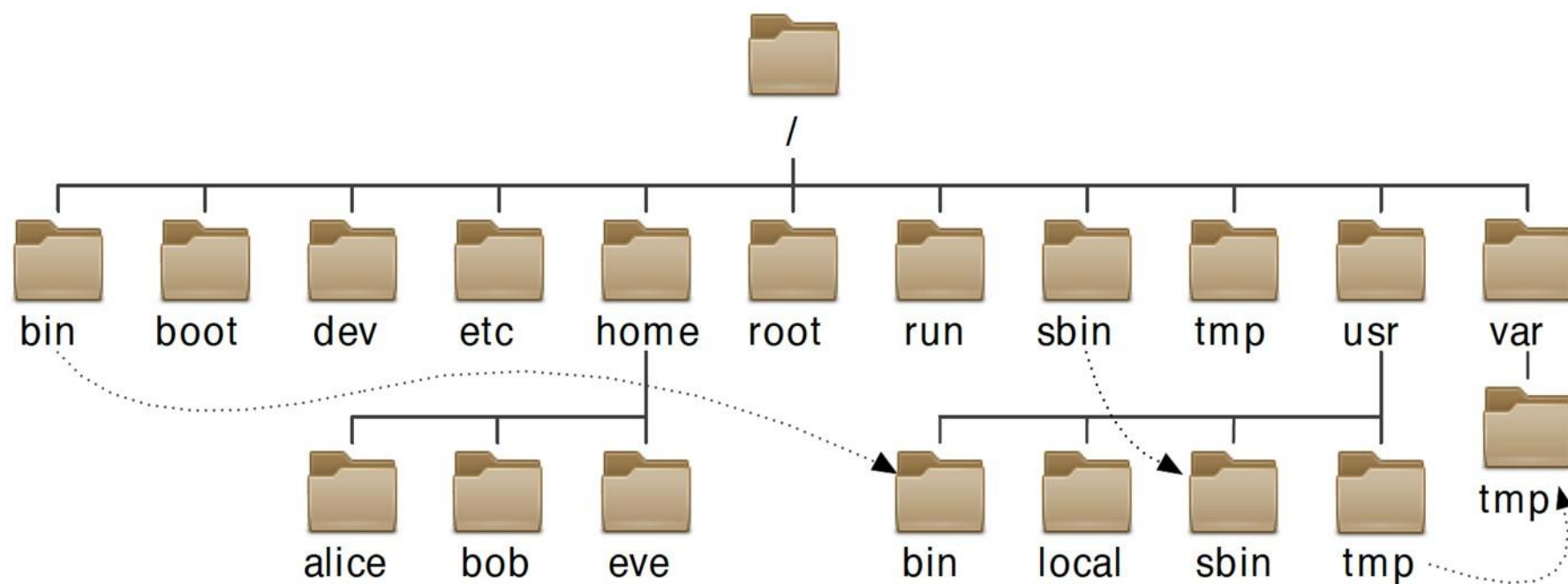
# Red Hat

## 一、Linux文件系统层次结构

# 文件系统层次结构

- ◆ Linux系统中的所有文件存储在文件系统中，他们被组织到一个颠倒的目录树中，称为文件系统层次结构。这棵树是颠倒的，因为树根在该层次结构的顶部，树根的下方延伸出目录和子目录的分支
- ◆ **/**目录是根目录，位于文件系统层次结构顶部。
- ◆ **/**字符还用作文件名中的目录分隔符。

# Linux目录结构



# /下的目录-1

## ◆ /usr

安装的软件、共享库，包括文件静态只读程序数据。重要的子目录：

/usr/bin---用户命令

/usr/sbin---系统管理命令

/usr/local---本地自定义软件

## ◆ /etc

特定于此系统的配置文件

## ◆ /var

特定于此系统的可变数据，在系统启动之间保持永久性。动态变化的文件（如数据库、缓存目录、日志文件、打印机后台处理文档和网站内容）可在/var下找到。

## ◆ /home

普通用户存储其个人数据和配置文件的主目录

## /下的目录-2

### ◆ /run

自上一次系统启动以来启动的进程的运行时数据。这包括进程ID文件和锁定文件。此目录中的内容在重启时重新创建

### ◆ /root

管理超级用户root的主目录

### ◆ /boot

开始启动过程所需文件

### ◆ /tmp

供临时文件使用的全局可写空间。10天内未访问、未更改或未修改的文件将自动从该目录中删除。还有一个临时目录/var/tmp,该目录中的文件如果在30天内未曾访问、更改或修改过，将被自动删除

### ◆ /dev

包含特殊的设备文件，供系统用于访问硬件



# Red Hat

## 二、根据名称查找文件

# 绝对路径和相对路径

## ◆ 绝对路径

绝对路径是完全限定名称，自根目录（**/**）开始，指定到达唯一代表单个文件所遍历的每个子目录。文件系统中每个文件都有一个唯一的绝对路径名。

规则识别：第一个字符是正斜杠（**/**）的路径名是绝对路径

## ◆ 相对路径

与绝对路径一样，相对路径也标识唯一文件，仅指定从工作目录到达该文件所需的路径。

识别规则：第一个字符是正斜杠（**/**）**之外**的其他字符的路径名是相对路径名。



# 路径名与导航路径

## 路径名规则

- ◆ 路径名长度（包含所有/）不可超过4095字节。
- ◆ 路径中通过/字符隔开的每一部分的长度不可超过255字节
- ◆ 区分大小写

## 导航路径

命令：**pwd**

显示当前位置的完整路径名，有助于决定使用相对路径名到达文件所需的相应语法。

命令：**cd**

更改目录位置

# .文件和..文件

**在每一个目录下都有一个.文件 与..文件（可用ls -a查看）**

◆ .文件：是对当前目录的一个硬连接

◆ ..文件：是对上级目录的一个硬连接

**注：**开头为“.”的文件名表示隐藏文件，在使用ls和其他命令的普通视图中不可见。这不是一种安全功能。隐藏文件用于防止必要的用户配置文件让主目录凌乱不堪。

“.”可为当前位置是来源或目标参数的命令指定当前目录，以免键入目录绝对路径名的必要

## 使用cd命令实例

◆ cd ..           #移动到上一个级别

◆ cd -            #将目录更改到用户在进入当前目录之前所在的目录



# Red Hat

## 三、使用命令行工具管理文件

## mkdir（创建目录）

命令：**mkdir**

- ◆ 创建一个或多个目录或子目录
- ◆ 如果文件名已经存在，或者尝试在不存在的父目录中创建目录，将生成错误。
- ◆ -p参数将为请求的目录位置创建缺失的父目录

## cp（复制）

命令：**cp**

复制一个或多个文件，成为新的独立文件

- ◆ 允许将一个现有文件复制为当前或另一个目录中的新文件，或者将多个文件复制到另一个目录中
- ◆ 在任何目标位置上，新文件名都必须唯一，如果新文件名不唯一，那么复制命令将**覆盖**现有文件
- ◆ 复制多个文件时，最后一个参数必须为**目录**
- ◆ 复制带有内容的**非空目录**要求使用**-r**递归选项

# mv（剪切）

命令：**mv**

在同一个目录中重命名文件，或者将文件重新放到新的目录中。

文件内容保持不变

实例：

◆ mv A B #重命名A文件成B

◆ mv A /home/Arthur #文件A移动到目录Arthur的家目录

# rm与rmdir

命令：**rm**

rm默认语法将删除文件，而不是目录。没有命令行取消删除功能，也没有可从中恢复的垃圾箱

参数：

- ◆ -r：删除目录以及其下可能存在的许多子目录和文件
- ◆ -i：交互方式提示每个删除操作
- ◆ -f：强制性进行删除不询问用户

命令：**rmdir**

仅删除**空的目录**。删除的目录无法取消删除



# Red Hat

## 四、使用路径名扩展匹配文件名



## 文件通配：路径名扩展

Bash Shell具有一个路径名匹配功能，以前叫通配。 Bash Shell通配功能通常称为**模式匹配**或**“通配符”**，可以使管理文件变得更加轻松。使用“扩展”的元字符来匹配要寻找的文件名和路径名，可以一次性对集中的一组文件执行命令。

# 模式匹配-1

通配是一种shell命令解析操作，它是一个通配符模式扩展到一组匹配的路径名。

## 常见元字符和模式类：

- ◆ \*：由0个或以上字符组成的任何字符串
- ◆ ?：任何一个字符
- ◆ ~：当前用户的主目录
- ◆ ~username：username用户的主目录
- ◆ [abc...]：括起来的类中的任何一个字符
- ◆ [!abc...]：不在括起的类中的任何一个字符
- ◆ [^abc...]：不在括起的类中的任何一个字符

## 模式匹配-2

- ◆ `[:alpha:]` : 任何字母字符
- ◆ `[:lower:]` : 任何小写字母
- ◆ `[:upper:]` : 任何大写字母
- ◆ `[:alnum:]` : 任何字母字符或数字
- ◆ `[:punct:]` : 除空格和字母数字以外的任何可打印字符
- ◆ `[:digit:]` : 任何数字, 即0-9
- ◆ `[:space:]` : 任何一个空白字符; 可能包含制表符、换行符、或回车符, 以及换页符和空格

# 实例

- ◆ `a*` 以字母a开头的文件或目录
- ◆ `*a` 以字母a结尾的文件或目录
- ◆ `*a*` 文件或目录名中包含字母a的
- ◆ `[!a]*` 不以字母a开头的文件或目录
- ◆ `[^abc]` 不包括a、b或c的文件或目录
- ◆ `???*` 文件名不少于3个字符长度的文件或目录
- ◆ `*[[:digit:]]*` 文件名中包含数字
- ◆ `[[:upper:]]*` 文件名以大写字母开头的

## { }扩展

大括号扩展用于生成任意字符串。大括号包含字符串的逗号分隔列表或顺序表达式。

**实例：**

- ◆ `echo {1,5,8}.log`
- ◆ `echo {1..3}.txt`
- ◆ `echo file{a,b}{1,2}.txt`
- ◆ `echo file{a{1,2},b,c}.txt`

# 命令替换

- ◆ 命令替换允许命令的输出替换命令本身。当命令包含前导美元符号和括号`$(command)`或者反引号``command``时，就会发生命令替换。
- ◆ 反引号形式比较成旧，有两个缺点：
  - 1) 反引号在视觉上容易和单引号混淆
  - 2) 反引号无法嵌套在反引号内。`$(command)`形式可以互相嵌套多个命令扩展

## 实例：

`echo today is `date +%A``

`echo the time is $(date +%M)`

# 防止参数被扩展

在Bash Shell中，许多字符有特殊含义。为了忽略元字符的特殊含义，使用引用和转义来防止它们被shell扩展。反斜杠(\)是bash中的一个转义字符，防止其后的一个字符被特殊解释。如果要保护**较长的字符串**，则使用单引号(')或双引号(")来括起字符串

使用**双引号**可以阻止通配和shell扩展，但**依然允许**命令和变量转换。变量替换在概念上与命令替换一样，但可能使用可选的大括号语法。

**实例：**

◆ mkdir redhat 1

# 单引号 ' '

使用**单引号**则可以按字面解译所有的文本。在屏幕和键盘上观察单引号 ( ' ) 和命令替换反引号 ( ` ) 的区别。

除了阻止通配和shell扩展外，引号也指示shell额外**阻止命令替换和变量替换**。

**实例：**

```
a=100      #创建一个变量a，值为100
```

**比较以下效果：**

```
echo "price is $a"
```

```
echo 'price is $a'
```



## 双引号 ""

使用**双引号**可以阻止通配和shell扩展，但依然**允许命令和变量替换**。变量替换在概念上与命令替换一样，但可能使用可选的**大括号语法**。

实例：

```
host=$(hostname)      #创建变量host  
echo "Will variable $host evaluate to $(hostname)?"
```

## 随堂练习

1、在当前用户家目录下创建12个空文件，命名规则为tv\_seasonX\_episodeY.ogg, X的取值范围1~2，Y的取值范围1~6。

```
[root@xmws ~]# touch tv_season{1,2}_episode{1..6}.ogg
```

2、创建8个空文件，命名规则为mystery\_chapterX.odf,X的取值范围1~8

```
[root@xmws ~]# touch mystery_chapter{1..8}.odf
```

3、在家目录的Video目录中创建2个目录season1和season2

4、移动刚才创建的文件tv\_season1开头的文件到Video/season1目录中。tv\_season2开头的文件到Video/season2目录中

5、使用一条命令在家目录的Documents目录中创建my\_bastseller目录，并在my\_bastseller目录中创建chapters目录

6、使用一条命令在刚才创建的my\_bastseller目录中创建editor，plot\_change，vacation三个子目录

7、切换到chapters目录中，并将家目录中所有mystery\_chapter开头的文件复制到其中

8、将复制过来的mystery\_chapter文件中前两个文件移动到editor目录中，使用相对路径

9、将复制到chapters目录中的7、8号文件复制到vacation中，使用一条命令完成

10、用一条命令切换到season2目录，tv\_season2\_episode1.ogg文件复制到vacation目录中。返回上一级目录，并列出目录中的内容。

11、使用一条命令切换到vacation目录中。列出目录中的内容。返回上一目录，将tv\_season2\_episode2.ogg文件复制到vacation目录中。再返回上一目录，列出目录中所有内容

12、切换到my\_bastseller目录中，复制chapters目录中5、6号文件到plot\_change目录中。列出chapters及其子目录的所有内容。

微思网络----福建IT精英的发源地！

