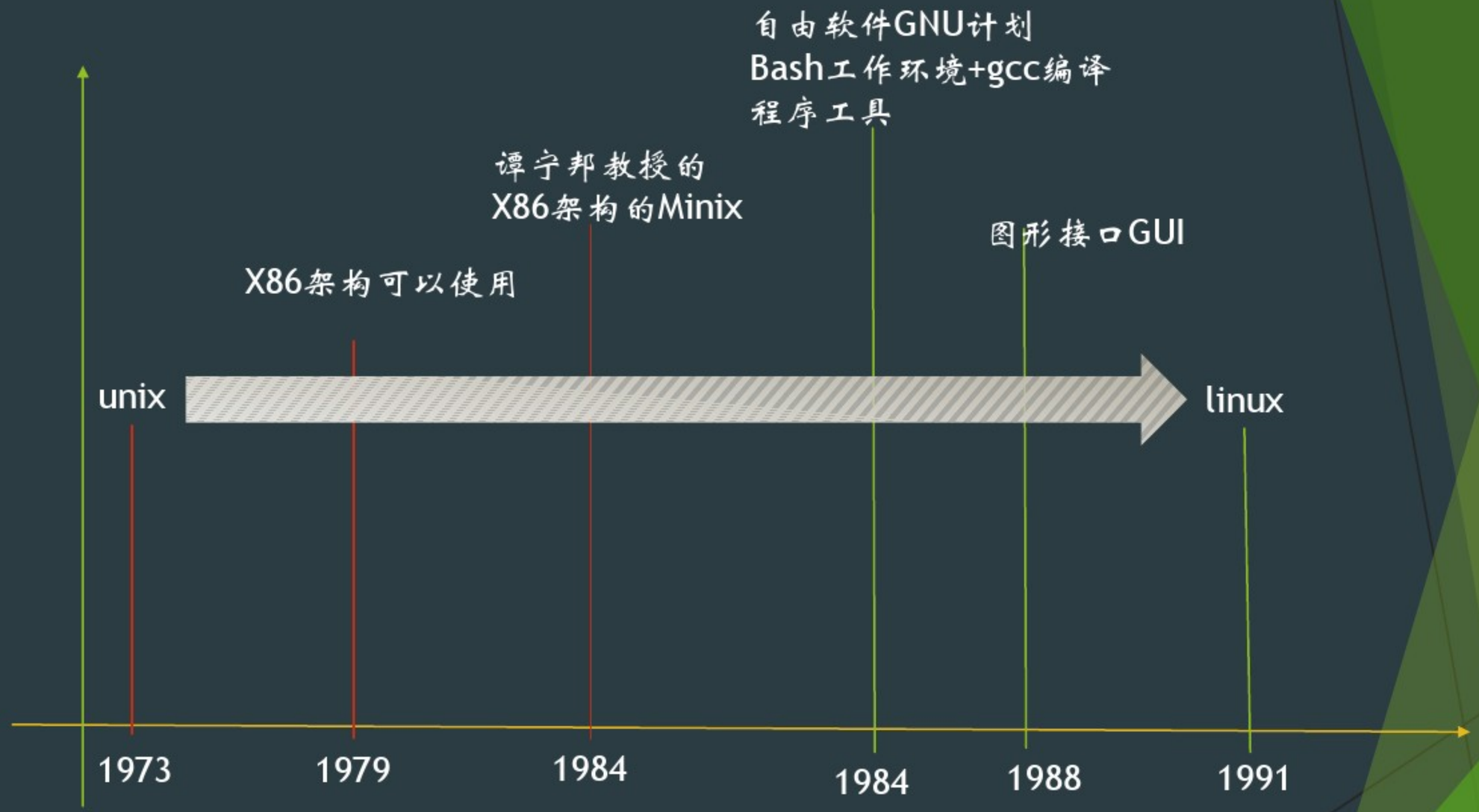
Linux基础知识讲解

作者：李玲枝

2018/1/3



* Linux是linus Torvalds 于1991年在Intel 386上由minix改造并带GUI界面的，使用bash、gcc等自由软件工具开发而来。

Linux系统安装-光盘安装

1.启动pc,开机按下[del]按键
2.进入bios，调整光盘启动为第一启动级别
3.保存退出
4.装入Centos 6.x i386的DVD在光盘中，重启电脑

安装图解：
http://www.runoob.com/linux/linux-install.html

Linux下载地址：
https://www.centos.org/download/

密码破解：
http://www.linuxidc.com/Linux/2014-01/94775.htm

作者：李玲枝

2018/1/3

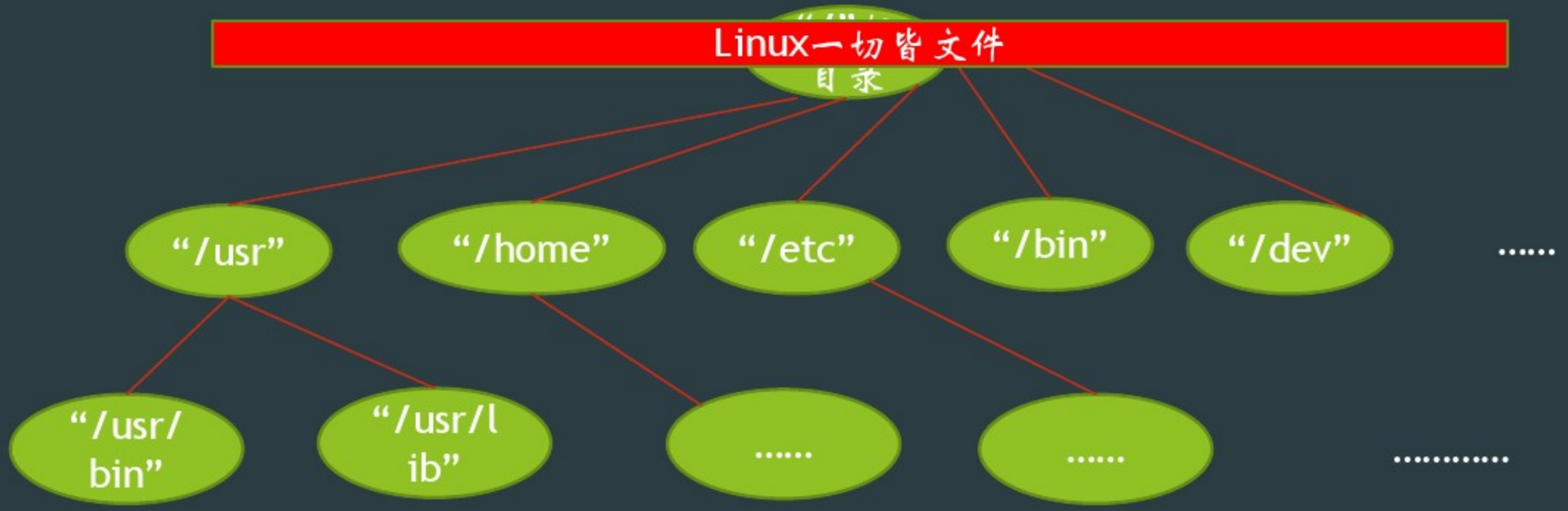
linux基础命令

- 1.文件/目录管理
- 2.用户/组管理
- 3.磁盘管理
- 4.网络配置
- 5.vim、vi编辑
- 6.yum软件安装
- 7.本地yum配置

作者：李玲枝

2018/1/3

Linux基础-文件/目录管理



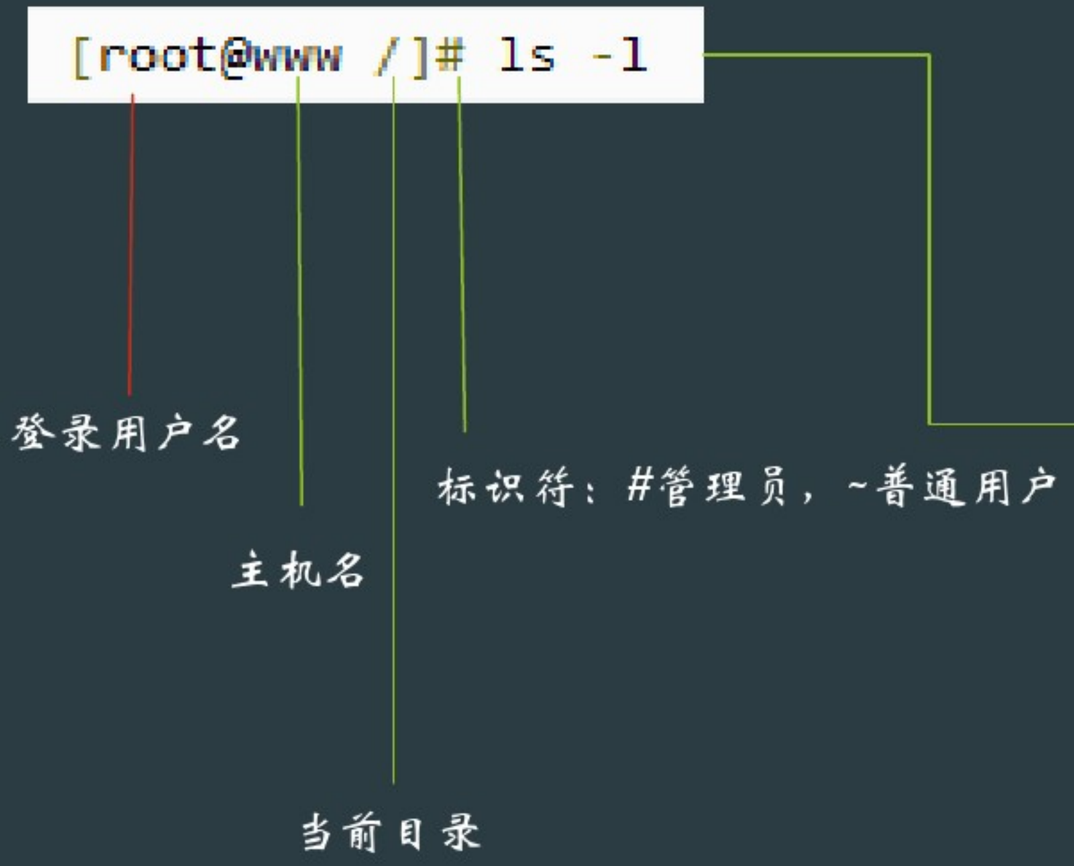
Linux目录管理：
http://www.jb51.net/LINUXjishu/151820.html

Linux中的文件类型
http://blog.csdn.net/zhuo_wang/article/details/8241516

作者：李玲枝

2018/1/3

Linux基础-文件/目录管理



【命令】

- ls: 列出目录
- cd：切换目录
- pwd：显示目前的目录
- mkdir：创建一个新的目录
- rmdir：删除一个空的目录
- cp: 复制文件或目录
- rm: 移除文件或目录
- cat 由第一行开始显示文件内容
- tac 从最后一行开始显示，可以看出 tac 是 cat 的倒著写！
- nl 显示的时候，顺道输出行号！
- more 一页一页的显示文件内容
- less 与 more 类似，但是比 more 更好的是，他可以往前翻页！
- head 只看头几行
- tail 只看尾巴几行
- chmod修改文件权限
- mv：移动文件

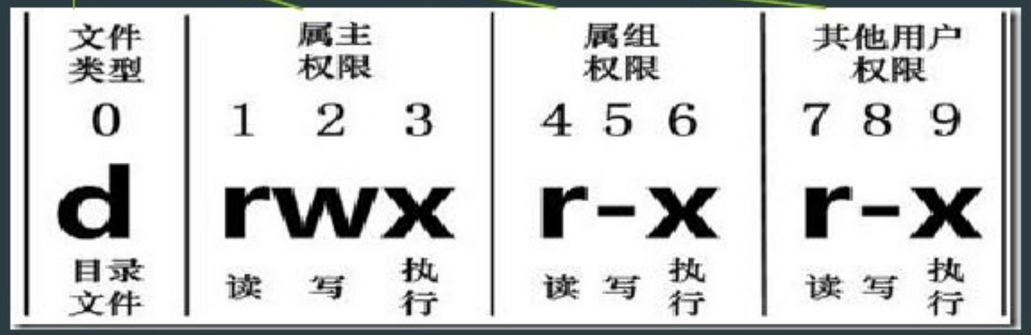
作者：李玲枝

2018/1/3

[root@www /]# ls -l

Linux基础-文件/目录管理

```
[root@www /]# ls -l
total 64
dr-xr-xr-x   2 root root 4096 Dec 14  2012 bin
```



文件-windows中文件
目录：windows中的文件夹

作者：李玲枝

2018/1/3

Linux基础-用户/用户组管理

Linux系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统

创建	修改	删除
useradd	usermod	userdel
	passwd（改密码）	

创建	修改	删除
groupadd	groupmod	groupdel

作者：李玲枝

2018/1/3

Linux基础-磁盘管理

添加磁盘

mount：查询系统中已经挂载的设备
mkdir /mnt/cdrom :建立挂载点
mount -t iso9660 /dev/cdrom /mnt/cdrom:挂载光盘

查看当前磁盘配额

df：列出文件系统的整体磁盘使用量
du：检查磁盘空间使用量

划分区管理磁盘

fdisk /dev/hdc:进入磁盘划分程序
mkfs -t ext3 /dev/hdc6:格式化分区

作者：李玲枝

2018/1/3

Linux基础-网路配置

vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0#网卡设备名称
ONBOOT=yes#启动时是否激活 yes | no
BOOTPROTO=static#协议类型 dhcp bootp none
IPADDR=192.168.1.90#网络IP地址
NETMASK=255.255.255.0#网络子网地址
GATEWAY=192.168.1.1#网关地址
BROADCAST=192.168.1.255#广播地址
HWADDR=00:0C:29:FE:1A:09#网卡MAC地址
TYPE=Ethernet#网卡类型为以太网

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.90
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
BROADCAST=192.168.1.255
HWADDR=00:0C:29:FE:1A:09
TYPE=Ethernet

#service network restart
#ifconfig

作者：李玲枝

2018/1/3

Linux基础-vim、vi编辑

所有的 Unix Like 系统都会内建 vi 文书编辑器，其他的文书编辑器则不一定会存在。
但是目前我们使用比较多的是 vim 编辑器

命令模式（Command mode），输入模式（Insert mode）和底线命令模式

i 切换到输入模式，以输入字符。
x 删除当前光标所在处的字符。
：切换到底线命令模式，以在最底一行输入命令

方向键，在文本中移动光标
HOME/END，移动光标到行首/行尾
Page Up/Page Down，上/下翻页
ESC;退出输入模式，切换到命令模式

q 退出程序
w 保存文件

作者：李玲枝

2018/1/3

Linux基础-yum软件安装

配置163 yum源
网易（163）yum源是国内最好的yum源之一 ，无论是速度还是软件版本，都非常的不错，将yum源设置为163 yum，可以提升软件包安装和更新的速度，同时避免一些常见软件版本无法找到

1.首先备份/etc/yum.repos.d/CentOS-Base.repo
mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup
2.下载对应版本repo文件，放入/etc/yum.repos.d/
3.运行以下命令生成缓存
yum clean all（清除下载的安装包）
yum makecache （生成本地缓存）

repo下载地址：
CentOS5 ： http://mirrors.163.com/.help/CentOS5-Base-163.repo
CentOS6 ： http://mirrors.163.com/.help/CentOS6-Base-163.repo

作者：李玲枝

2018/1/3

Linux基础-本地yum配置

步骤一：搭建一个本地Yum，将centos6光盘手动挂载到/media
命令操作如下所示：
[root@localhost ~]# mount /dev/cdrom /media/
步骤二：将本地设置为客户端，进行Yum验证
Yum客户端需编辑配置文件，命令操作如下所示：
[root@localhost ~]# cd /etc/yum.repos.d/　　//必须在这个路径下
[root@localhost yum.repos.d]# ls　　//此路径下事先有配置文件的模板
rhel-source.repo
[root@localhost yum.repos.d]# cp rhel-source.repo rhel6.repo //配置文件必须以.repo结尾
[root@localhost yum.repos.d]# vim rhel6.repo
[rhel-6]　　//中括号里内容要求唯一，但不要出现特殊字符
name=Red Hat Enterprise Linux 6　　//此为描述信息，可以看情况填写
baseurl=file:///media/　//此项为yum软件仓库位置，指向光盘挂载点
enabled=1　　//此项为是否开启，1为开启0为不开启
gpgcheck=1　//此项为是否检查签名，1为监测0为不检测
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release　//签名认证信息的路径

作者：李玲枝

2018/1/3

谢谢！

作者：李玲枝

2018/1/3

Shell 编程

一个用C语言编写的程序，Shell是一种命令语言，也是一种程序设计语言

一个应用程序，这个应用程序提供了一个界面，用户通过这个界面访问操作系统内核的服务

Shell 编程跟 java、php 编程一样，只要有一个能编写代码的文本编辑器和一个能解释执行的脚本解释器就可以了

Shell 种类众多，常见的有：

Bourne Shell（/usr/bin/sh或/bin/sh）

Bourne Again Shell（/bin/bash）

C Shell（/usr/bin/csh）

K Shell（/usr/bin/ksh）

Shell for Root（/sbin/sh）

.....

Bash是大部分linux默认的shell，且免费方便

作者：李玲枝

2018/1/3

编写一个shell脚本(text.sh)

打开文本编辑器(可以使用 vi/vim 命令来创建文件)，新建一个文件 test.sh，扩展名为 sh（sh代表shell），扩展名并不影响脚本执行，见名知意就好，如果你用 php 写 shell 脚本，扩展名就用 php 好了

vim编辑脚本text.sh

```
#！ /bin/bash #标注使用/bin/bash作为执行命令解释器
echo “hello world！ ” #直接输出hello word！
```

执行脚本 text.sh

```
chmod +x ./text.sh #使脚本具有执行权限
./text.sh #执行脚本
```

直接写 test.sh，linux 系统会去 PATH 里寻找有没有叫 test.sh 的，而只有 /bin, /sbin, /usr/bin，/usr/sbin 等在 PATH 里，你的当前目录通常不在 PATH 里，所以写成 test.sh 是会找不到命令的，要用 ./test.sh 告诉系统说，就在当前目录找。

作者：李玲枝

2018/1/3

Shell变量

变量赋值

```
your_name=“runoob.com” #给变量赋值，变量名和等号之间不能有空格
for fire in ‘ls /etc’ # 将/etc下目录的文件名循环出来
```

使用一个定义过的变量，只要在变量名前面加美元符号即可

```
your_name=“llz”
echo $your_name
echo ${your_name}
```

变量名外面的花括号是可选的，加不加都行，
加花括号是为了帮助解释器识别变量的边界

```
readonly myUrl #只读变量，运行脚本后结果如下：

your_name="tom"
echo $your_name
 your_name="alibaba"
echo $your_name
```

这样写是合法的，但注意，第二次赋值的时候不能写
$your_name="alibaba"，使用变量的时候才加美元符（$）

作者：李玲枝

2018/1/3

删除变量 unset，unset 命令不能删除只读变量

变量类型
1) **局部变量** 局部变量在脚本或命令中定义，仅在当前shell实例中有效，其他shell启动的程序不能访问局部变量。
2) **环境变量** 所有的程序，包括shell启动的程序，都能访问环境变量，有些程序需要环境变量来保证其正常运行。必要的时候shell脚本也可以定义环境变量。
3) **shell变量** shell变量是由shell程序设置的特殊变量。shell变量中有一部分是环境变量，有一部分是局部变量，这些变量保证了shell的正常运行

Shell字符串
字符串可以用单引号，也可以用双引号，也可以不用引号

单引号字符串的限制：
单引号里的任何字符都会原样输出，单引号字符串中的变量是无效的；
单引号字串中不能出现单引号（对单引号使用转义符后也不行）。

pk

双引号的优点：
双引号里可以有变量
双引号里可以出现转义字符

```
string="abcd"
echo ${#string} #输出 4
```

```
string="runoob is a great site" echo ${string:1:4} # 输出 unoo
```

```
string="runoob is a great company" echo `expr index "$string" is` # 输出 8
```

作者：李玲枝

2018/1/3

echo ${var%%/*}

Shell 数组

bash支持一维数组（不支持多维数组），并且没有限定数组的大小。
类似与C语言，数组元素的下标由0开始编号。获取数组中的元素要利用下标，下标可以是整数或算术表达式，其值应大于或等于0。

数组名=(值1 值2 ... 值n) 定义数组

```
array_name=(value0 value1 value2 value3)
```

```
array_name=(
value0 value1
value2
value3)
```

```
array_name[0]=value0
array_name[1]=value1
array_name[n]=valuen
```

${数组名[下标]} 读取数组

# 取得数组元素的个数 length=${#array_name[@]} # 或者 length=${#array_name[*]} # 取得数组单个元素的长度
lengthn=${#array_name[n]}

echo ${var%/*}：#、## 表示从左边开始删除。一个 # 表示从左边删除到第一个指定的字符；两个 # 表示从左边删除到最后一个指定的字符。
%、%% 表示从右边开始删除。一个 % 表示从右边删除到第一个指定的字符；两个 % 表示从左边删除到最后一个指定的字符。删除包括了指定的字符本身

从左边第几个字符开始，及字符的个数 echo ${var:0:5}

作者：李玲枝

2018/1/3

Shell 传递参数

我们可以在执行 Shell 脚本时，向脚本传递参数，脚本内获取参数的格式为：**$n**。**n** 代表一个数字，1 为执行脚本的第一个参数，2 为执行脚本的第二个参数，以此类推……

实例

以下实例我们向脚本传递三个参数，并分别输出，其中 **$0** 为执行的文件名：

```
#!/bin/bash
# author:菜鸟教程
# url:www.runoob.com echo "Shell 传递参数实例！ ";
echo "执行的文件名：$0";
echo "第一个参数为：$1";
echo "第二个参数为：$2";
echo "第三个参数为：$3";
```

为脚本设置可执行权限，并执行脚本，输出结果如下所示：

```
$ chmod +x test.sh
$ ./test.sh 1 2 3
Shell 传递参数实例！
执行的文件名：./test.sh
第一个参数为：1
第二个参数为：2
第三个参数为：3
```

作者：李玲枝　　2018/1/3

Shell 运算符

Shell 和其他编程语言一样，支持多种运算符，包括：

- 算数运算符
- 关系运算符
- 布尔运算符
- 字符串运算符
- 文件测试运算符

运算符；
http://www.runoob.com/linux/linux-shell-basic-operators.html

Echo 命令
http://www.runoob.com/linux/linux-shell-echo.html

作者：李玲枝

2018/1/3

Shell 函数

linux shell 可以用户定义函数，然后在shell脚本中可以随便调用。

shell中函数的定义格式如下：

```
[ function ] funname [()]

{

    action;

    [return int;]

}
```

Shell 文件包含

和其他语言一样，Shell 也可以包含外部脚本。这样可以很方便的封装一些公用的代码作为一个独立的文件。

Shell 文件包含的语法格式如下：

```
. filename   # 注意点号(.)和文件名中间有一空格

或

source filename
```

作者：李玲枝

创建两个 shell 脚本文件。

test1.sh 代码如下：

```
#!/bin/bash
# author:菜鸟教程
# url:www.runoob.com
url="http://www.runoob.com"
```

test2.sh 代码如下：

```
#!/bin/bash
# author:菜鸟教程
# url:www.runoob.com
#使用 . 号来引用test1.sh 文件
. ./test1.sh
# 或者使用以下包含文件代码
# source ./test1.sh

echo "菜鸟教程官网地址：$url"
```

2018/1/3