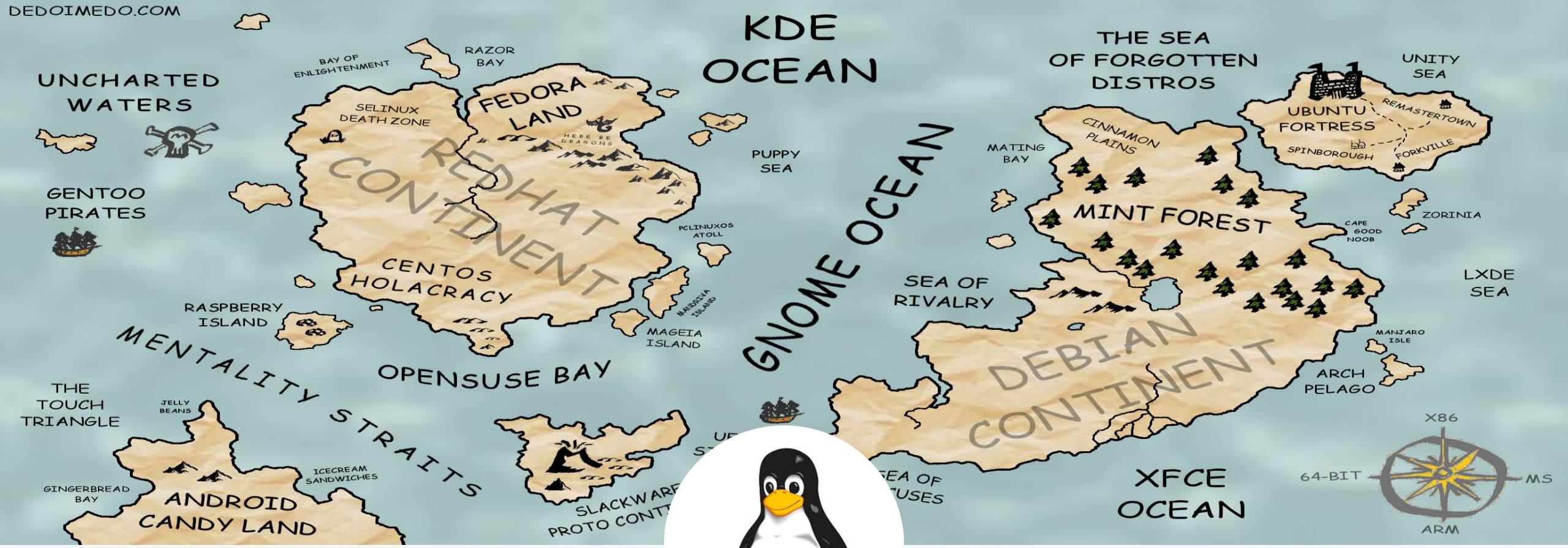


Vim编辑器与Shell命令脚本

任课教师：刘遄 www.LinuxProbe.com

课程概述

01 Vim文本编辑器
VIM Text Editor

02 编写Shell脚本
Writing Shell Scripts

03 流程控制语句
Process Control Statement

04 计划任务服务程序
Scheduled Task Service Procedure

前言

01 介绍如何使用Vim编辑器来编写和修改文档，然后通过逐步配置主机名称、系统网卡以及软件仓库等文件，帮助大家加深Vim编辑器中诸多命令、快捷键与模式的理解。

02 在Shell脚本中以多种方式接收用户输入的信息，能够对输入值进行文件、数字、字符串的判断比较。在熟练使用“与、或、非”三种逻辑操作符的基础上，大家还要充分学习if、for、while、case条件测试语句，并通过10多个实战脚本的实操练习，达到在工作中灵活运用的水准。

03 通过实战的方式演示了使用at命令与crond计划任务服务来分别实现一次性的系统任务设置和长期性的系统任务设置，在分钟、小时、日期、月份、年份的基础上实现工作的自动化，从而让日常的工作更加高效。

Vim文本编辑器

VIM Text Editor

Vim文本编辑器

[Vim的发布]

Vim的发布最早可以追溯到1991年，英文全称为Vi Improved。它也是Vi编辑器的提升版本，其中最大的改进当属添加了代码着色功能，在某些编程场景下还能自动修正错误代码。

[文本编辑器]

在Linux系统中一切都是文件，而配置一个服务就是在修改其配置文件的参数。而且在日常工作中大家也肯定免不了要编写文档，这些工作都是通过文本编辑器来完成的。

[目的]

让读者切实掌握Linux系统的运维方法，而不是仅仅停留在“会用某个操作系统”的层面上，我们这里选择使用Vim文本编辑器，它默认会安装在当前所有的Linux操作系统上，是一款超棒的文本编辑器。

Vim文本编辑器

1

命令模式

控制光标移动，可对文本进行复制、粘贴、删除和查找等工作。

2

输入模式

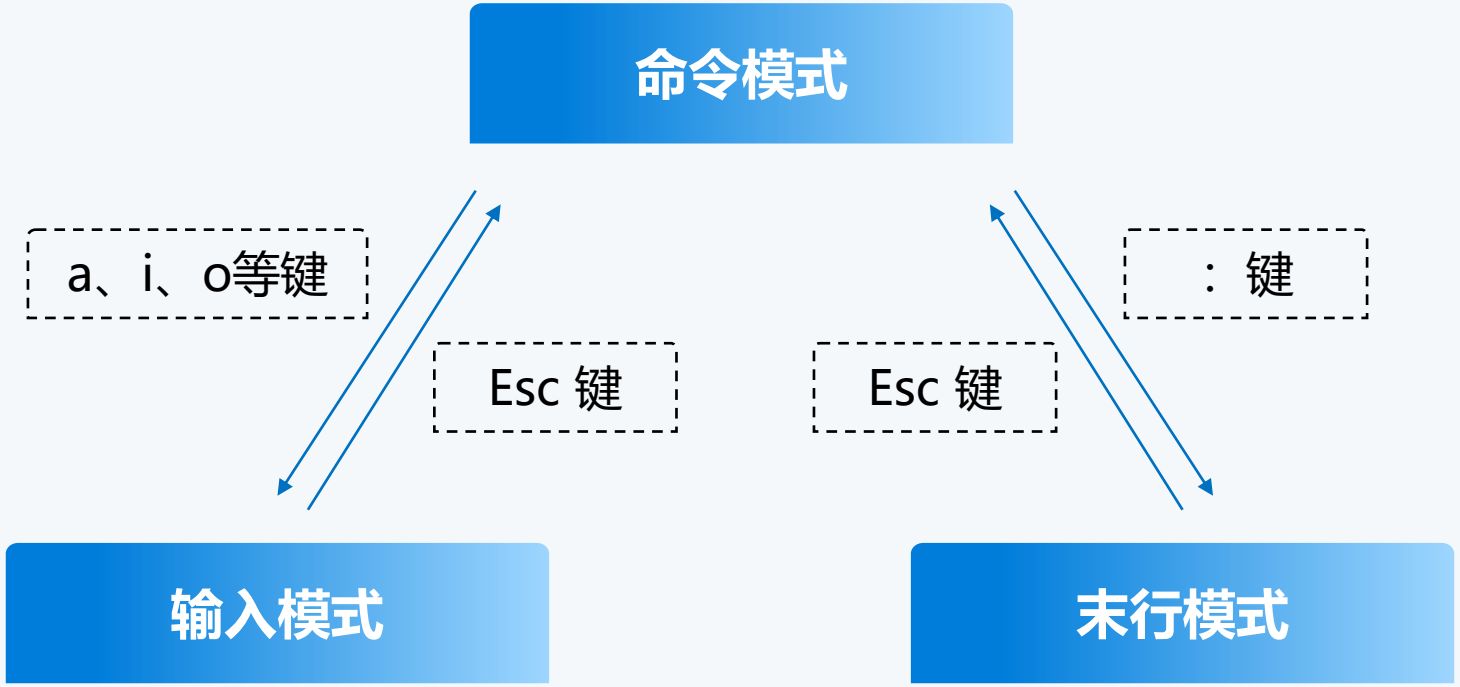
正常的文本录入。

3

末行模式

保存或退出文档，以及设置编辑环境。

Vim文本编辑器



Vim编辑器模式的切换方法

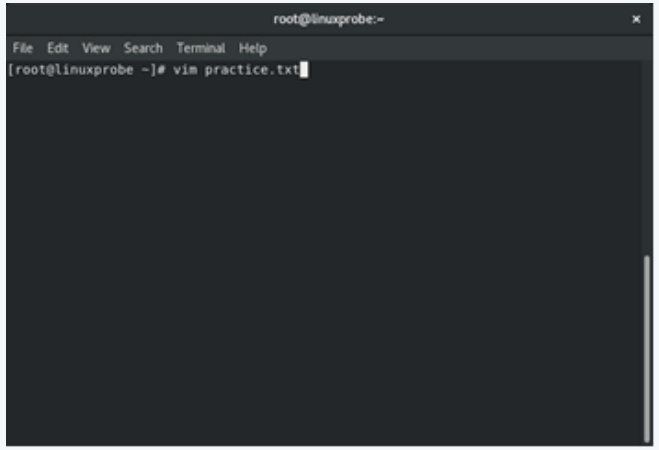
命令模式中最常用的一些命令

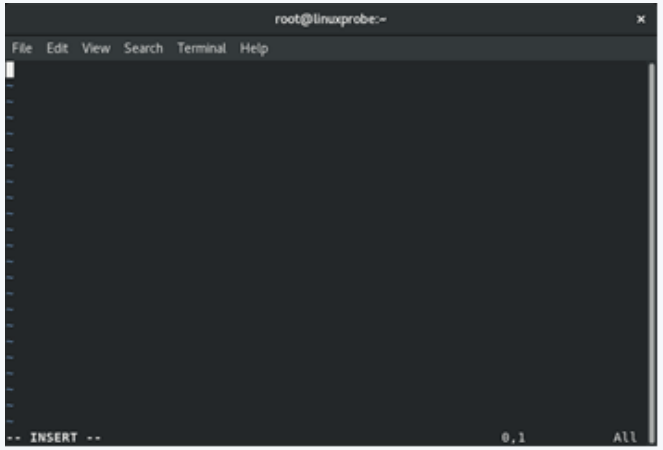
命令	作用
dd	删除（剪切）光标所在整行
5dd	删除（剪切）从光标处开始的5行
yy	复制光标所在整行
5yy	复制从光标处开始的5行
n	显示搜索命令定位到的下一个字符串
N	显示搜索命令定位到的上一个字符串
u	撤销上一步的操作
p	将之前删除（dd）或复制（yy）过的数据粘贴到光标后面

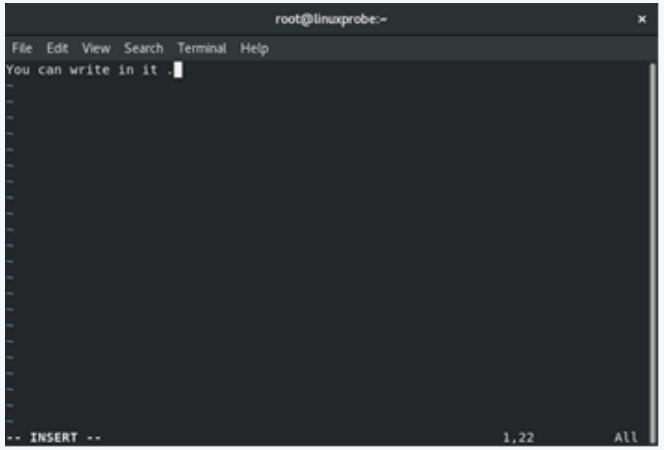
末行模式中常用的一些命令

命令	作用
:w	保存
:q	退出
:q!	强制退出（放弃对文档的修改内容）
:wq!	强制保存退出
:set nu	显示行号
:set nonu	不显示行号
:命令	执行该命令
:整数	跳转到该行
:s/one/two	将当前光标所在行的第一个one替换成two
:s/one/two/g	将当前光标所在行的所有one替换成two
:%s/one/two/g	将全文中的所有one替换成two
?字符串	在文本中从下至上搜索该字符串
/字符串	在文本中从上至下搜索该字符串

编写简单文档

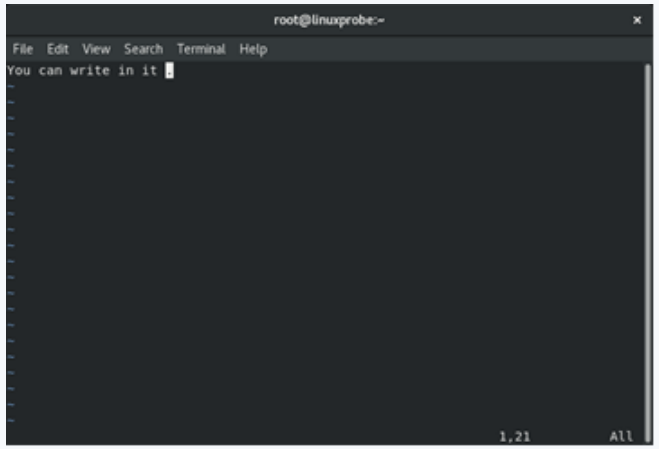


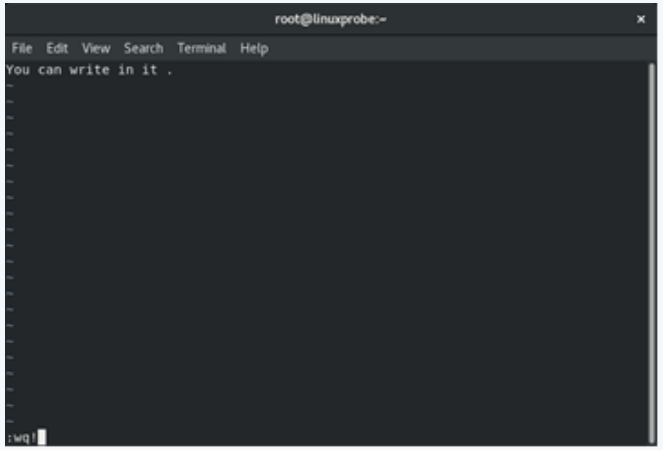


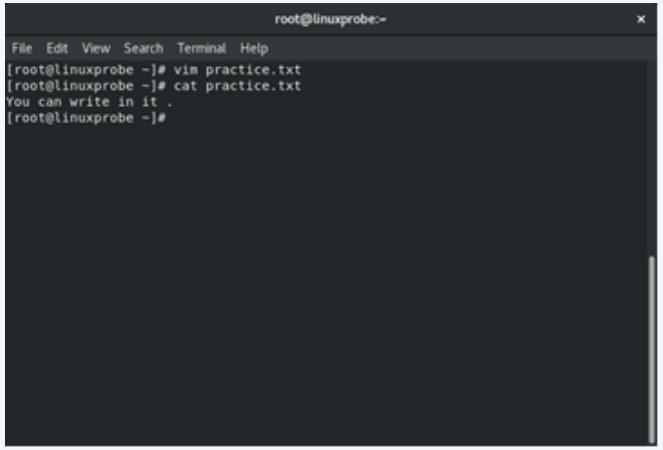


尝试编写文本文档 → 切换至编辑器的输入模式 → 在编辑器中输入文本内容

编写简单文档

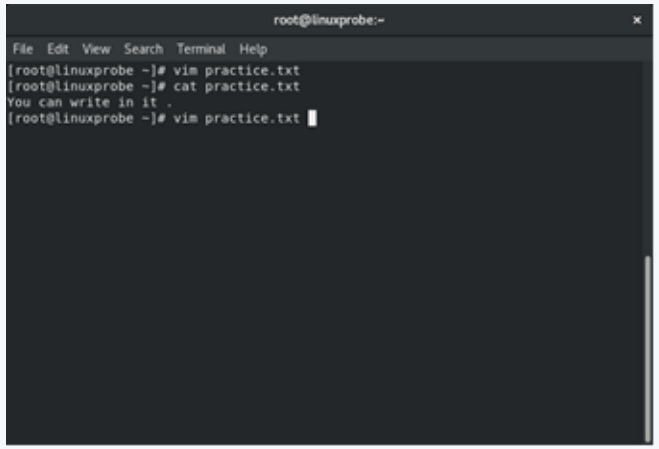


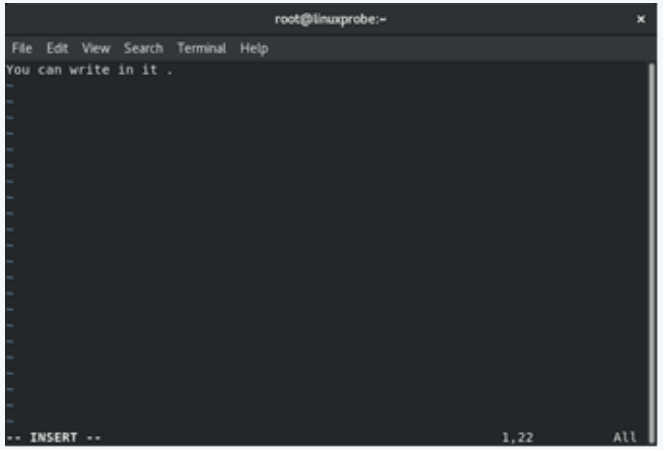


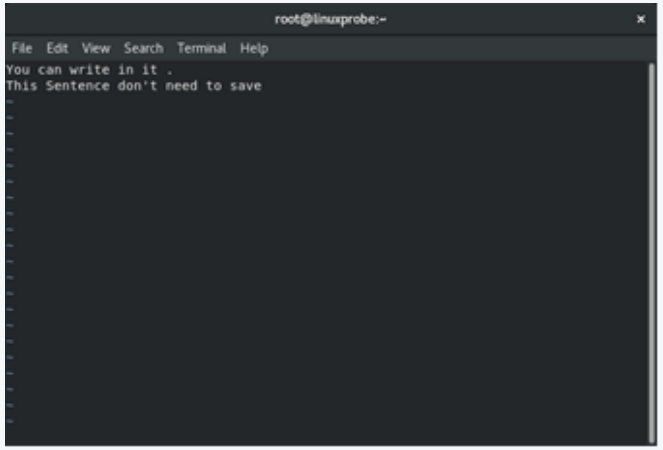


切换至编辑器的命令模式 → 切换至编辑器的末行模式 → 查看文档的内容

编写简单文档





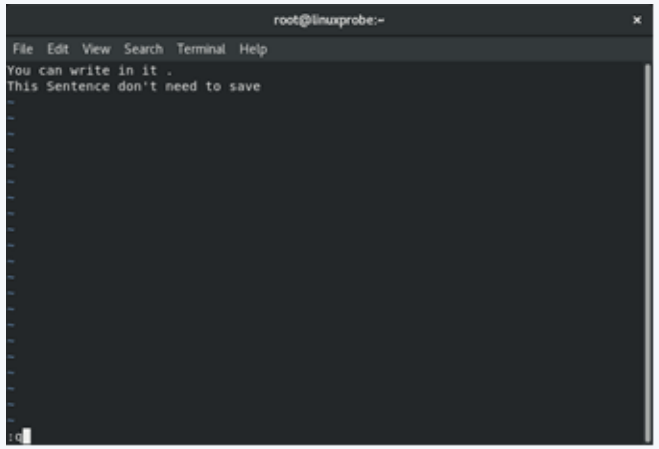


再次通过Vim编辑器编写文档

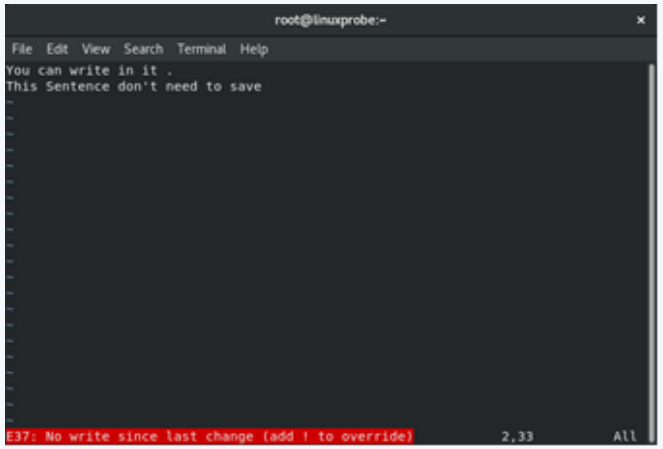
进入Vim编辑器的输入模式

追加写入一行文本内容

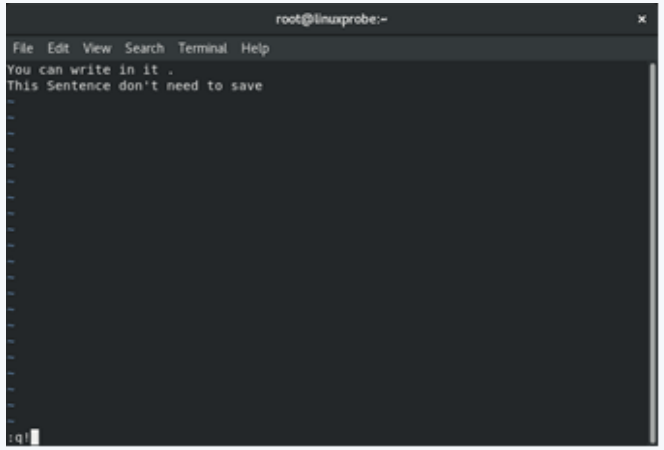
编写简单文档



退出文本编辑器

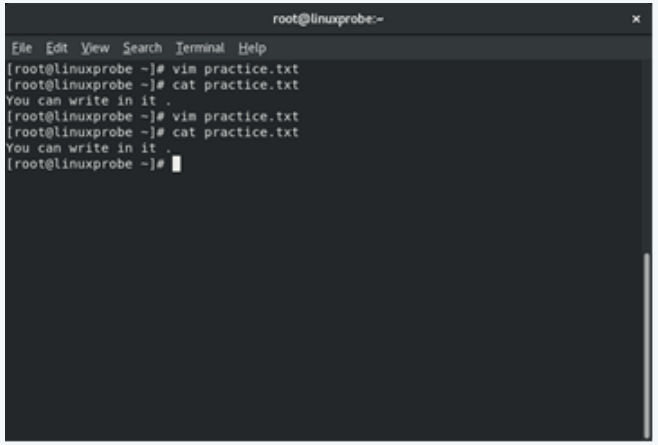


因文件已被修改而拒绝退出操作



强制退出文本编辑器

编写简单文档



查看最终编写成的文本内容

注

下面的实验如果做不成功也很正常，请大家把重心放到Vim编辑器上面，能成功修改配置文件就已经很棒啦！

配置主机名称

1

第1步

使用Vim编辑器修改/etc/hostname主机名称文件。

2

第2步

把原始主机名称删除后追加“linuxprobe.com”。注意，使用Vim编辑器修改主机名称文件后，要在末行模式下执行“:wq!”命令才能保存并退出文档。

3

第3步

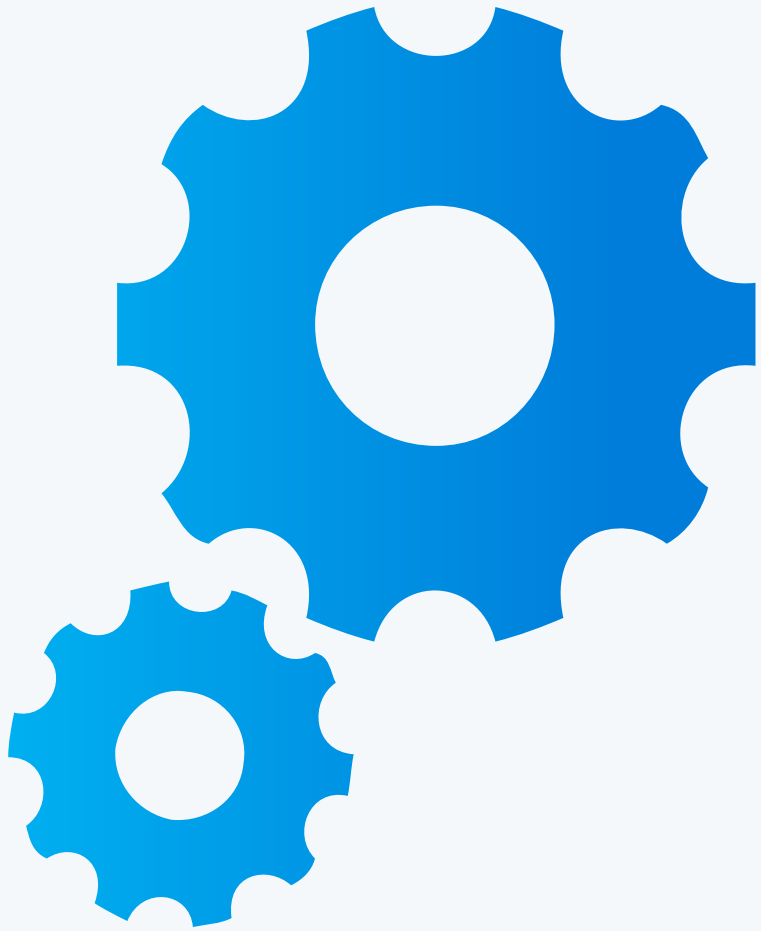
保存并退出文档，然后使用hostname命令检查是否修改成功。

配置网卡信息步骤

01 首先切换到/etc/sysconfig/network-scripts目录中（存放着网卡的配置文件）。

02 使用Vim编辑器修改网卡文件ifcfg-ens160，逐项写入下面的配置参数并保存退出。由于每台设备的硬件及架构是不一样的，因此请读者使用ifconfig命令自行确认各自网卡的默认名称。

设备类型：TYPE=Ethernet
地址分配模式：BOOTPROTO=static
网卡名称：NAME=ens160
是否启动：ONBOOT=yes
IP地址：IPADDR=192.168.10.10
子网掩码：NETMASK=255.255.255.0
网关地址：GATEWAY=192.168.10.1
DNS地址：DNS1=192.168.10.1



03 重启网络服务并测试网络是否连通。

配置软件仓库步骤

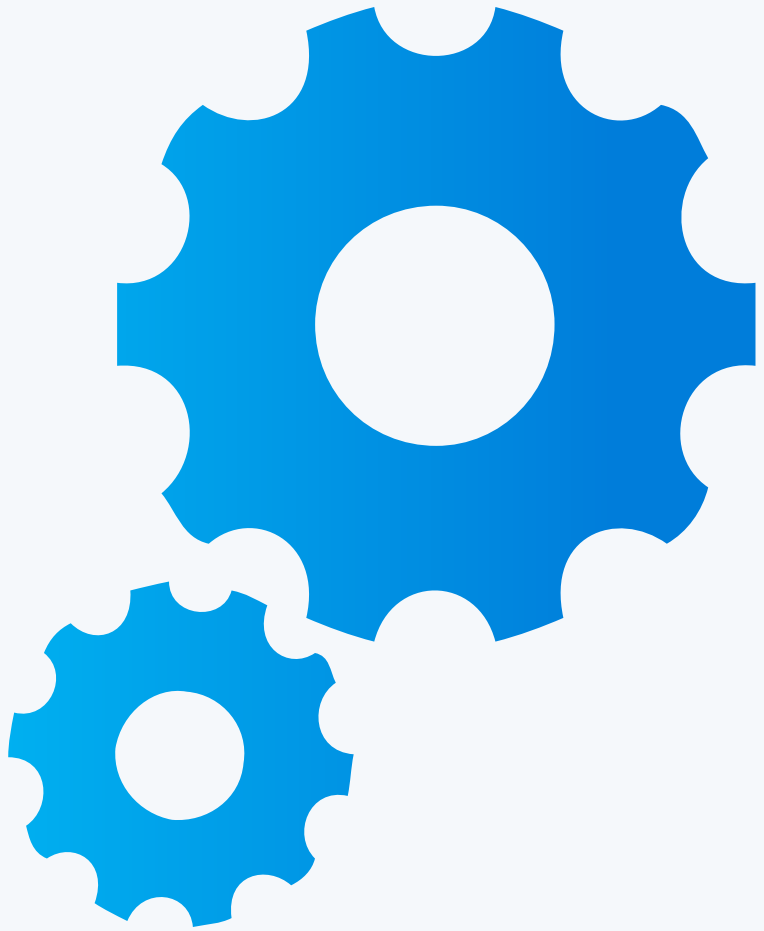
01 进入/etc/yum.repos.d/目录中（因为该目录存放着软件仓库的配置文件）。

02 使用Vim编辑器创建一个名为rhel8.repo的新配置文件（文件名称可随意，但后缀必须为.repo），逐项写入下面的配置参数并保存退出。

仓库名称：具有唯一性的标识名称，不应与其他软件仓库发生冲突。
描述信息（name）：可以是一些介绍性的词，易于识别软件仓库的用处。
仓库位置（baseurl）：软件包的获取方式，可以使用FTP或HTTP下载，也可以是本地的文件（需要在后面添加file参数）。
是否启用（enabled）：设置此源是否可用；1为可用，0为禁用。
是否校验（gpgcheck）：设置此源是否校验文件；1为校验，0为不校验。
公钥位置（gpgkey）：若上面的参数开启了校验功能，则此处为公钥文件位置。若没有开启，则省略不写。

03 按配置参数中所填写的仓库位置挂载光盘，并把光盘挂载信息写入/etc/fstab文件中。

04 使用“dnf install httpd -y”命令检查软件仓库是否已经可用。

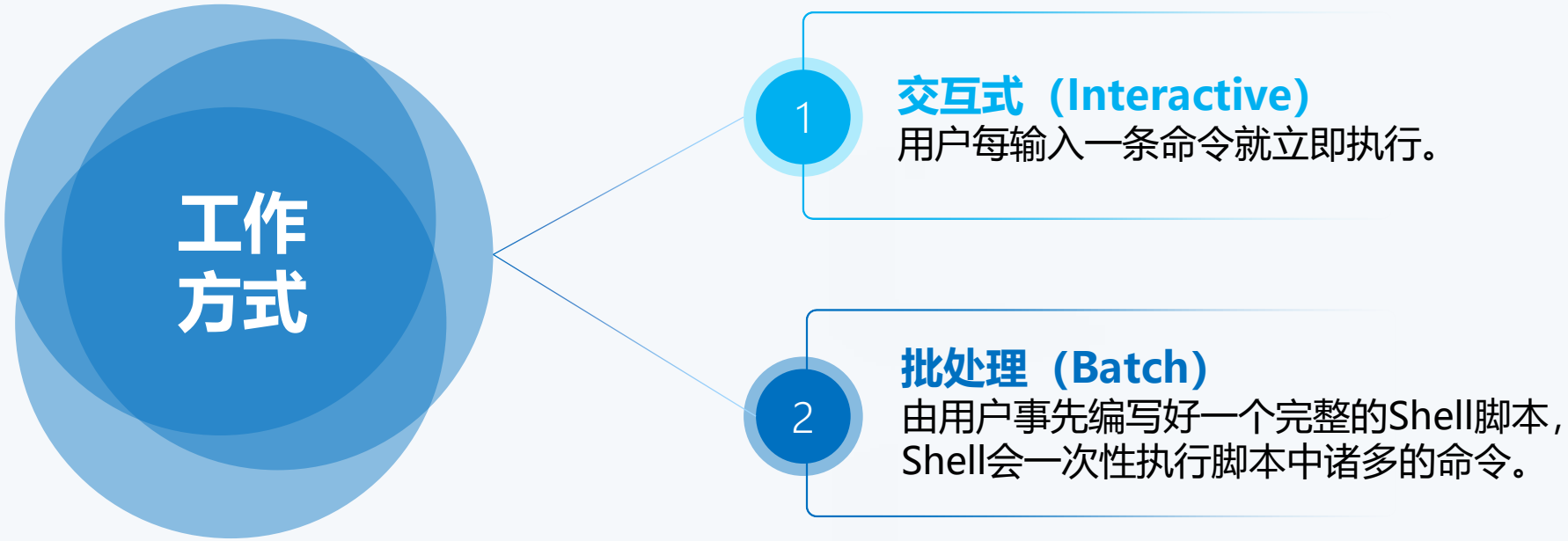


编写Shell脚本

Writing Shell Scripts

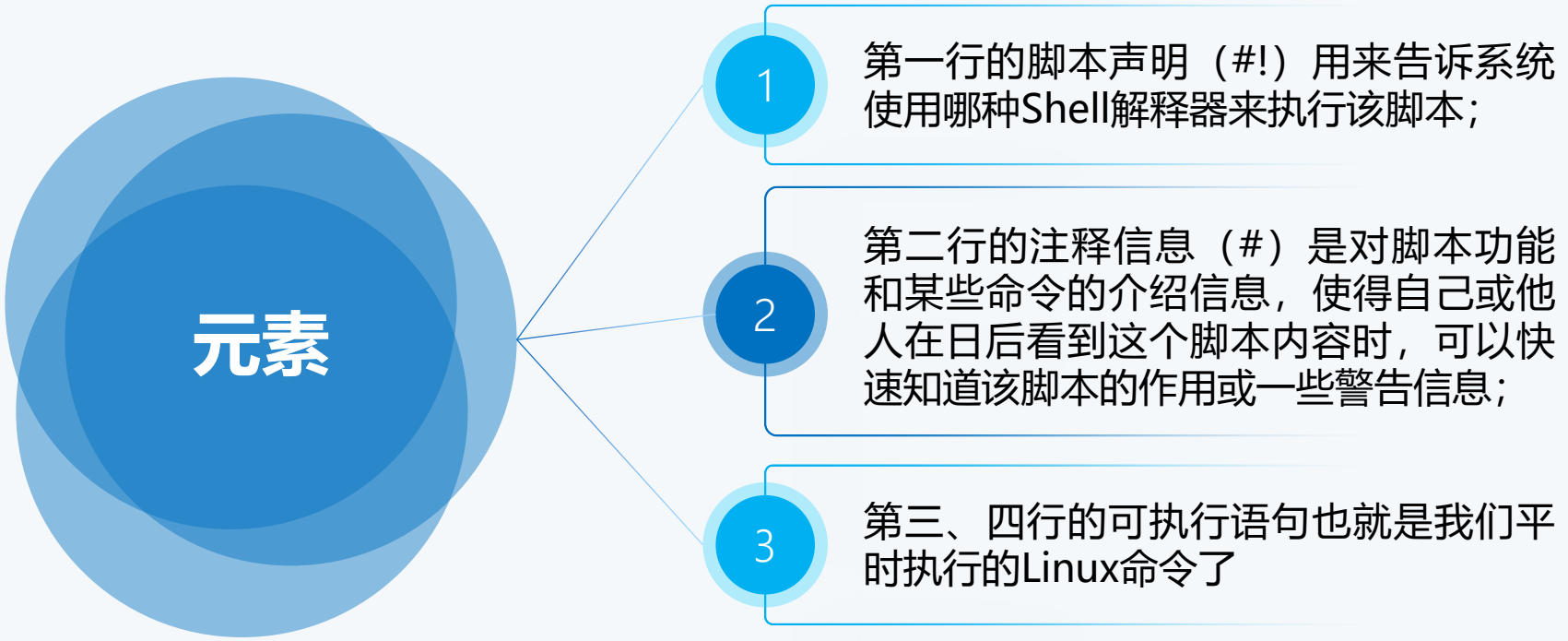
Shell脚本命令

可以将Shell终端解释器当作人与计算机硬件之间的“翻译官”，它作为用户与Linux系统内部的通信媒介，除了能够支持各种变量与参数外，还提供了诸如循环、分支等高级编程语言才有的控制结构特性。

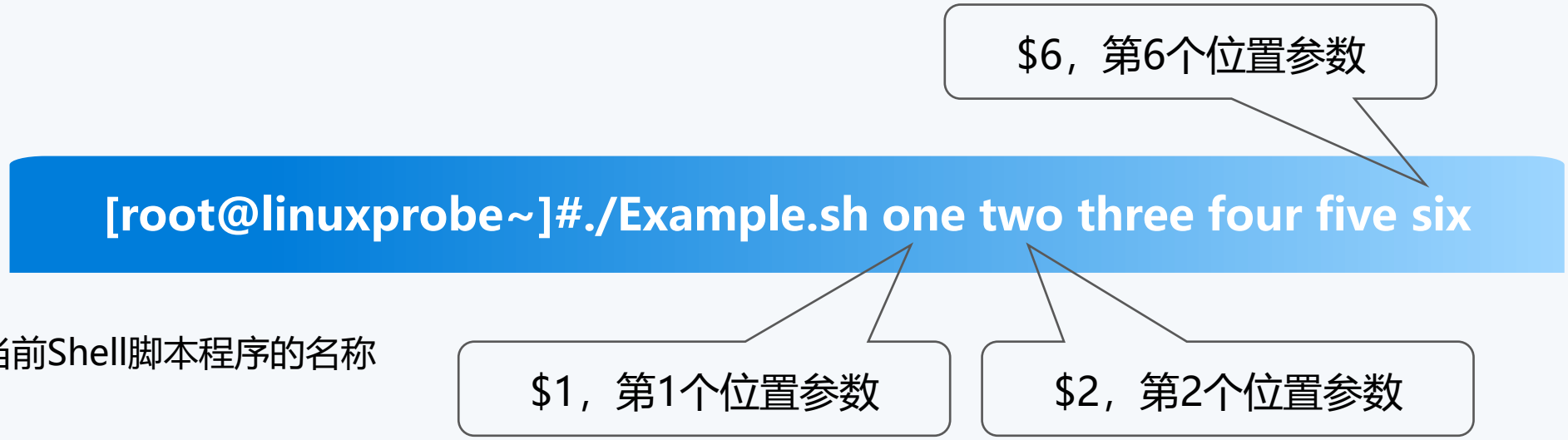


编写简单的脚本

```
[root@linuxprobe~]# vim example.sh
#!/bin/bash
#For Example BY linuxprobe.com
pwd
ls -al
```



接受用户的参数



Shell脚本程序中的参数位置变量

01 $0对应的是当前Shell脚本程序的名称

02 $#对应的是总共有几个参数

03 $*对应的是所有位置的参数值

04 $?对应的是显示上一次命令的执行返回值

05 $1、$2、$3……则分别对应着第N个位置的参数值

判断用户的参数

测试语句格式

hell脚本中的条件测试语法可以判断表达式是否成立，若条件成立则返回数字0，否则便返回非零值。条件表达式两边均应有一个空格。

按照测试对象来划分，条件测试语句

- 1、文件测试语句
- 2、逻辑测试语句
- 3、整数值比较语句
- 4、字符串比较语句

文件测试所用参数

运算符	作用
-d	测试文件是否为目录类型
-e	测试文件是否存在
-f	判断是否为一般文件
-r	测试当前用户是否有权限读取
-w	测试当前用户是否有权限写入
-x	测试当前用户是否有权限执行

逻辑语句

1. &&是逻辑“与”，只有当前面的语句执行成功的时候才会执行后面的语句。
2. ||是逻辑“或”，只有当前面的语句执行失败的时候才会执行后面的语句。
3. !是逻辑“非”，代表对逻辑测试结果取反值；之前若为正确则变成错误，若为错误则变成正确。

可用的整数比较运算符

运算符	作用
-eq	是否等于
-ne	是否不等于
-gt	是否大于
-lt	是否小于
-le	是否等于或小于
-ge	是否大于或等于

常见的字符串比较运算符

运算符	作用
=	比较字符串内容是否相同
!=	比较字符串内容是否不同
-z	判断字符串内容是否为空

流程控制语句

Process Control Statement

流程控制语句

1. 尽管此时可以通过使用Linux命令、管道符、重定向以及条件测试语句来编写最基本的Shell脚本，但是这种脚本并不适用于生产环境。

2. 原因是它不能根据真实的工作需求来调整具体的执行命令，也不能根据某些条件实现自动循环执行。通俗来讲，就是不能根据实际情况做出调整。

3. 通常脚本都是从上到下一股脑儿地执行，效率是很高，但一旦某条命令执行失败了，则后面的功能全都会受到影响。

流程控制语句

if条件测试语句

if条件测试语句可以让脚本根据实际情况自动执行相应的命令。从技术角度来讲，if语句分为单分支结构、双分支结构、多分支结构；其复杂度随着灵活度一起逐级上升。

单分支的if条件语句

```
If   条件测试操作
   then   命令序列
fi
```

```
If   目录不存在
   then   创建该目录
fi
```

双分支的if条件语句

```
If   条件测试操作
   then   命令序列 1
   else   命令序列 2
fi
```

```
If   能够ping通
   then   提示服务器正常工作
   else   报警服务器出现问题
fi
```

多分支的if条件语句

```
If   条件测试操作 1
   then   命令序列 1
Elif   条件测试操作 2
   then   命令序列 2
Else
   命令序列 3
fi
```

```
If   分数为85~100之间
   then   判为优秀
Elif   分数为70~84之间
   then   判为合格
Else
   判为不合格
fi
```

流程控制语句

for条件循环语句

for循环语句允许脚本一次性读取多个信息，然后逐一对信息进行操作处理。当要处理的数据有范围时，使用for循环语句就再适合不过了。

For范围循环语句

```
for   变量名in取值列表
Do
        命令序列
done
```

```
For   用户名in列表文件
Do
        创建用户并设置密码
done
```

流程控制语句

while条件循环语句

while条件循环语句是一种让脚本根据某些条件来重复执行命令的语句，它的循环结构往往在执行前并不确定最终执行的次数，while循环语句通过判断条件测试的真假来决定是否继续执行命令，若条件为真就继续执行，为假就结束循环。

while条件循环语句

```
While　条件测试操作
do
　　命令序列
done
```

```
while　未猜中正确价格
do
　　反复猜测商品价格
done
```

流程控制语句

case条件测试语句

case 条件测试语句和switch语句的功能非常相似！case语句是在多个范围内匹配数据，若匹配成功则执行相关命令并结束整个条件测试；如果数据不在所列出的范围内，则会去执行星号（*）中所定义的默认命令。

case条件测试语句

```
case  变量值in
模式 1）
      命令序列 1
      ；；
模式 2）
      命令序列 2
      ；；
      ……
*）
      默认命令序列
esac
```

```
Case  输入的字符 in
[a-z] | [A-Z] ）
      提示为字母。
      ；；
[0-9] ）
      提示为数字。
      ；；
      ……
*）
      提示为特殊字符
esac
```

计划任务服务程序

Scheduled Task Service Procedure

计划任务服务程序

01

一次性计划任务：今晚23:30重启网站服务。
一次性计划任务只执行一次，一般用于临时的工作需求。
可以用at命令实现这种功能，只需要写成“at时间”的形式就行。
如果想要查看已设置好但还未执行的一次性计划任务，可以使用at -l命令；要想将其删除，可以使用“atrm任务序号”。

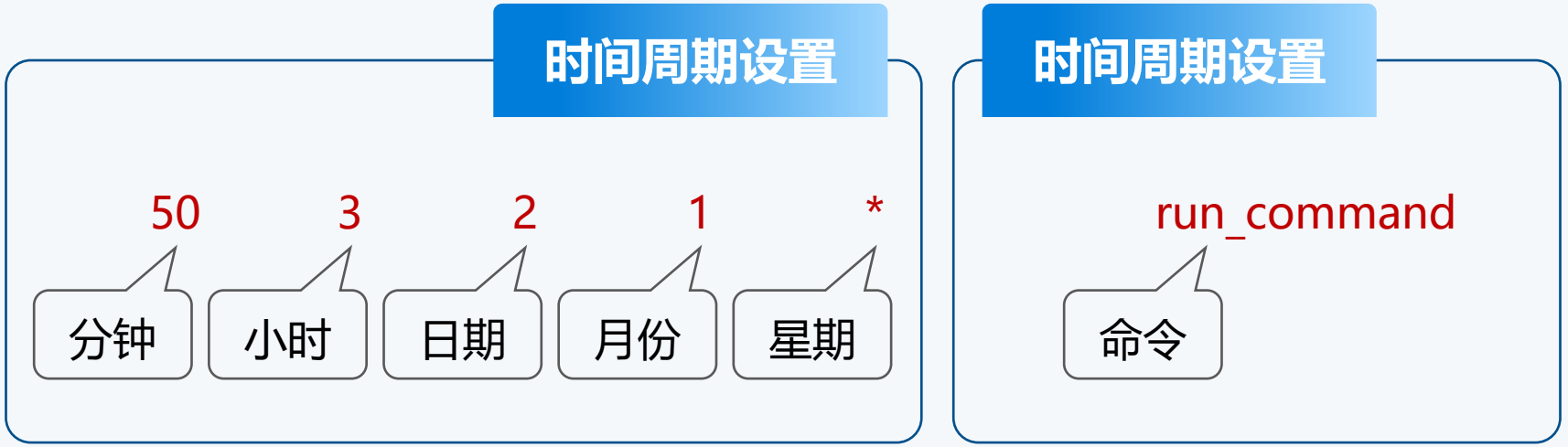
02

长期性计划任务：每周一的凌晨3:25把/home/wwwroot目录打包备份为backup.tar.gz。

参数	作用
-e	编辑计划任务
-u	指定用户名称
-l	列出任务列表
-r	删除计划任务

crontab命令中的参数及其作用

使用crond设置任务的参数格式



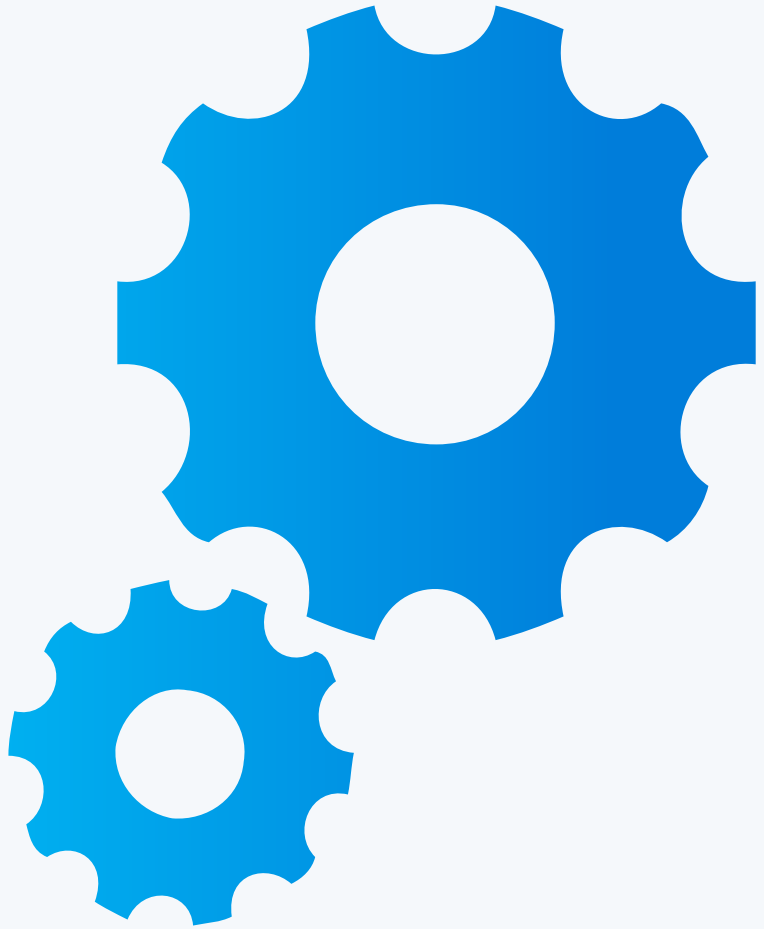
使用crond设置任务的参数字段说明

字段	说明
分钟	取值为0～59的整数
小时	取值为0～23的任意整数
日期	取值为1～31的任意整数
月份	取值为1～12的任意整数
星期	取值为0～7的任意整数，其中0与7均为星期日
命令	要执行的命令或程序脚本

使用计划服务的注意事项

01 在crond服务的配置参数中，一般会像Shell脚本那样以#号开头写上注释信息，这样在日后回顾这段命令代码时可以快速了解其功能、需求以及编写人员等重要信息。

02 计划任务中的“分”字段必须有数值，绝对不能为空或是*号，而“日”和“星期”字段不能同时使用，否则就会发生冲突。



复习题

✓ **1．Vim编辑器的3种模式分别是什么?**

答：命令模式、末行模式与输入模式（也叫编辑模式或插入模式）。

✓ **2．怎么从输入模式切换到末行模式?**

答：需要先敲击Esc键退回到命令模式，然后敲击冒号（:）键后进入末行模式。

✓ **3．一个完整的Shell脚本应该包含哪些内容?**

答：应该包括脚本声明、注释信息和可执行语句（即命令）。

✓ **4．分别解释Shell脚本中$0与$3变量的作用。**

答：在Shell脚本中，$0代表脚本文件的名称，$3则代表该脚本在执行时接收的第3个参数。

✓ **5．if条件测试语句有几种结构，最灵活且最复杂的是哪种结构?**

答：if条件测试语句包括单分支、双分支与多分支等3种结构，其中多分支结构是最灵活且最复杂的结构，其结构形式为if…then…elif…then…else…fi。

复习题

✓ **6．for条件循环语句的循环结构是什么样子的?**

答：for条件循环语句的结构为“for变量名in取值列表do命令序列done”，如图4-21所示。

✓ **7．若在while条件循环语句中使用true作为循环条件，那么会发生什么事情?**

答：由于条件测试值永久为true，因此脚本中的循环部分会无限地重复执行下去，直到碰到exit命令才会结束。

✓ **8．如果需要依据用户的输入参数执行不同的操作，最方便的条件测试语句是什么?**

答：case条件语句。

✓ **9．Linux系统的长期计划任务所使用的服务是什么，其参数格式是什么?**

答：长期计划任务需要使用crond服务程序，参数格式是“分、时、日、月、星期 命令”。

祝同学们学习顺利，爱上Linux系统。