

# 红帽系统管理二

学员练习册

**Internal use only - Not For Sale**

RH134-RHEL8.0-zh-CN-1-20190531  
MAN-RH134SKC.PDF

海量视频题库 myitpub.com www.52myit.com QQ:5565462

海量视频题库 myitpub.com QQ:5565462  
www.52myit.com

加入探索者、建设者和个人的队伍，一起为老问题大胆提供新解决方案。开源技术的创新离不开幕后人员的贡献。



# Red Hat

## Training and Certification

### 学员练习册

红帽企业 Linux 8.0 RH134

红帽系统管理二

版 1



Red Hat

myitpub.com  
www.52myit.com  
QQ:5565462

海量视频题库 myitpub.com QQ:5565462  
www.52myit.com

# 红帽系统管理二

Red Hat

myitpub.com  
www.52myit.com  
qq:5565462

**红帽企业 Linux 8.0 RH134  
红帽系统管理二  
版 1 20190531  
出版日期 20190531**

作者: Fiona Allen, Adrian Andrade, Herve Quatremain, Victor Costea,  
Snehangshu Karmakar, Marc Kesler, Saumik Paul  
编者: Philip Sweany, Ralph Rodriguez, David Sacco, Seth Kenlon, Heather  
Charles

Copyright © 2019 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are  
Copyright © 2019 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but  
not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of  
Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat,  
Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details  
contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please e-mail  
[training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, Hibernate, Fedora, the Infinity Logo, and RHCE are  
trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or  
other countries.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/  
service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack  
Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the  
OpenStack community.

All other trademarks are the property of their respective owners.

供稿者: Achyut Madhusudan、Rob Locke、Rudolf Kastl、Prashant Rastogi、Heider  
Souza、Michael Phillips、Dallas Spohn

海量视频题库 myitpub.com  
www.52myit.com QQ:5565462

<b>文档规范</b>	<b>ix</b>
<b>简介</b>	<b>xi</b>
红帽系统管理二 .....	xi
课堂环境介绍 .....	xii
国际化 .....	xiv
<b>1. 提高命令行生产率</b>	<b>1</b>
编写简单的 Bash 脚本 .....	2
指导练习: 编写简单的 Bash 脚本 .....	5
使用循环更高效地运行命令 .....	8
指导练习: 使用循环更高效地运行命令 .....	14
使用正则表达式匹配命令输出中的文本 .....	16
指导练习: 使用正则表达式匹配命令输出中的文本 .....	24
开放研究实验: 提高命令行生产率 .....	27
总结 .....	33
<b>2. 计划将来的任务</b>	<b>35</b>
计划延迟的用户作业 .....	36
指导练习: 计划延迟的用户作业 .....	38
计划周期性用户作业 .....	41
指导练习: 计划周期性用户作业 .....	44
计划周期性系统作业 .....	47
指导练习: 计划周期性系统作业 .....	50
管理临时文件 .....	54
指导练习: 管理临时文件 .....	57
小测验: 计划将来的任务 .....	60
总结 .....	64
<b>3. 调优系统性能</b>	<b>65</b>
调整调优配置文件 .....	66
指导练习: 调整调优配置文件 .....	71
影响进程调度 .....	73
指导练习: 影响进程调度 .....	76
开放研究实验: 调优系统性能 .....	79
总结 .....	83
<b>4. 使用 ACL 控制对文件的访问</b>	<b>85</b>
解释文件 ACL .....	86
小测验: 解释文件 ACL .....	92
使用 ACL 保护文件安全 .....	94
指导练习: 使用 ACL 保护文件安全 .....	98
开放研究实验: 使用 ACL 控制对文件的访问 .....	103
总结 .....	111
<b>5. 管理 SELinux 安全性</b>	<b>113</b>
更改 SELinux 到强制模式 (及引导式练习) .....	114
指导练习: 更改 SELinux 强制模式 .....	118
控制 SELinux 文件上下文 .....	121
指导练习: 控制 SELinux 文件上下文 .....	125
使用布尔值调整 SELinux 策略 .....	128
指导练习: 使用布尔值调整 SELinux 策略 .....	130
调查和解决 SELinux 问题 .....	133
指导练习: 调查和解决 SELinux 问题 .....	137
开放研究实验: 管理 SELinux 安全性 .....	141
总结 .....	147
<b>6. 管理基本存储</b>	<b>149</b>

添加分区、文件系统和持久挂载 .....	150
指导练习: 添加分区、文件系统和持久挂载 .....	159
管理交换空间 .....	163
指导练习: 管理交换空间 .....	167
开放研究实验: 管理基本存储 .....	171
总结 .....	179
<b>7. 管理逻辑卷</b>	<b>181</b>
创建逻辑卷 .....	182
指导练习: 创建逻辑卷 .....	189
扩展逻辑卷 .....	194
指导练习: 扩展逻辑卷 .....	199
开放研究实验: 管理逻辑卷 .....	203
总结 .....	209
<b>8. 实施高级存储功能</b>	<b>211</b>
使用 Stratis 管理分层存储 .....	212
指导练习: 使用 Stratis 管理分层存储 .....	217
使用 VDO 压缩存储和删除重复数据 .....	222
指导练习: 使用 VDO 压缩存储和删除重复数据 .....	225
开放研究实验: 实施高级存储功能 .....	229
总结 .....	238
<b>9. 访问网络附加存储</b>	<b>239</b>
通过 NFS 挂载网络附加存储 .....	240
指导练习: 通过 NFS 管理网络附加存储 .....	244
自动挂载网络附加存储 .....	248
指导练习: 自动挂载网络附加存储 .....	251
开放研究实验: 访问网络附加存储 .....	257
总结 .....	264
<b>10. 控制启动过程</b>	<b>265</b>
选择启动目标 .....	266
指导练习: 选择启动目标 .....	271
重置 root 密码 .....	274
指导练习: 重置 root 密码 .....	277
修复在启动时出现的文件系统问题 .....	279
指导练习: 修复在启动时出现的文件系统问题 .....	280
开放研究实验: 控制启动过程 .....	283
总结 .....	288
<b>11. 管理网络安全</b>	<b>289</b>
管理服务器防火墙 .....	290
指导练习: 管理服务器防火墙 .....	297
控制 SELinux 端口标记 .....	301
指导练习: 控制 SELinux 端口标记 .....	304
开放研究实验: 管理网络安全 .....	308
总结 .....	316
<b>12. 安装红帽企业 Linux</b>	<b>317</b>
安装红帽企业 Linux .....	318
指导练习: 安装红帽企业 Linux .....	322
使用 Kickstart 自动安装 .....	325
指导练习: 使用 Kickstart 自动安装 .....	333
安装和配置虚拟机 .....	336
小测验: 安装和配置虚拟机 .....	341
开放研究实验: 安装红帽企业 Linux .....	343
总结 .....	349

## 13. 总复习

**351**

总复习 .....	352
开放研究实验: 修复启动问题和维护服务器 .....	355
开放研究实验: 配置和管理文件系统与存储 .....	362
开放研究实验: 配置和管理服务器安全性 .....	369

海量视频题库 myitpub.com QQ:5565462  
www.52myit.com

# 文档规范



## 参考文献

“引用”部分介绍了查找与主题相关的外部文档的位置。



## 注意

“注意事项”包括针对执行手头任务的提示、快捷方式或者替代方法。忽略注意事项不会有不良后果，但您可能错失某个可简化操作的技巧。



## 重要

“重要信息”框详细介绍容易忽略的内容：仅适用于当前会话的配置更改，或在应用更新前需要重新启动的服务。忽略标有“重要信息”的框不会导致数据丢失，但可能引起不便和困扰。



## 警告

“警告”不可忽略。忽略警告将很可能导致数据丢失。



# 简介

## 红帽系统管理二

本课程专为已完成红帽系统管理一 (RH124) 的学员设计。红帽系统管理二 (RH134) 侧重讲解成为 Linux 全职管理员以及通过“红帽认证系统管理员”考试验证必备技能所需的关键任务。本课程将会深入介绍企业 Linux 管理技能，包括文件系统和分区、逻辑卷、SELinux、防火墙和故障排除。

### 课程目标

- 进一步扩展在 红帽系统管理一 (RH124) 课程中学到的技能。
- 培养 RHCSA 认证 红帽企业 Linux 系统管理员所需的技能。

### 培训对象

- 本课程专为已完成 红帽系统管理一 (RH124) 的学员设计。由于课程的特定教学内容编排，学员不适合将 RH134 作为此课程的入门教材使用。我们鼓励尚未学习前一个红帽课程的学员，学习系统管理一 (RH124)（针对 Linux 新手）或 RHCSA 速成课程 (RH200)（针对具有企业 Linux 管理经验的学员）。

### 先决条件

- 完成 红帽系统管理一 (RH124) 课程或具有相关同等知识。

# 课堂环境介绍

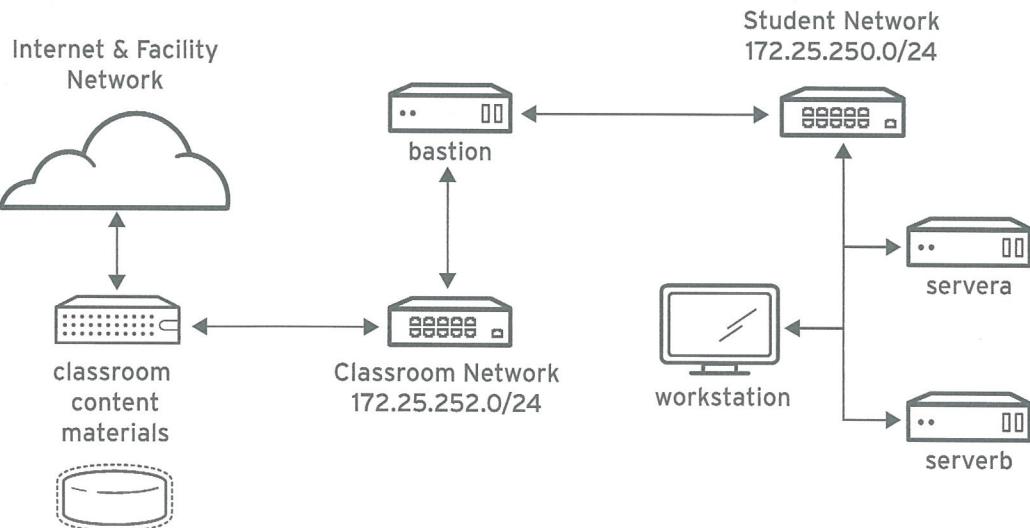


图 0.1: 课堂环境

在本课程中，用于实践学习活动的主要计算机系统是 **workstation**。另有两台计算机供学员用于这些活动：**servera** 和 **serverb**。这三个系统都在 `lab.example.com` DNS 域内。

所有学员计算机系统都有一个标准用户帐户 **student**，其密码为 **student**。所有学员系统的 **root** 密码都是 **redhat**。

## 教室计算机

计算机名称	IP 地址	角色
<code>bastion.lab.example.com</code>	172.25.250.254	用于将学员专用网络连接到课堂服务器的网关系统（必须始终处于运行状态）
<code>workstation.lab.example.com</code>	172.25.250.9	用于进行系统管理的图形工作站
<code>servera.lab.example.com</code>	172.25.250.10	第一台服务器
<code>serverb.lab.example.com</code>	172.25.250.11	第二台服务器

**bastion** 的主要功能是充当连接学员计算机的网络和课堂网络之间的路由器。如果 **bastion** 容机，则其他学员计算机将只能访问各个学员网络上的系统。

课堂中有几个系统提供支持服务。`content.example.com` 和 `materials.example.com` 这两台服务器提供实践活动中使用的软件和实验材料。有关如何使用这些服务器的信息将在这些活动的说明中提供。这些信息由 `classroom.example.com` 虚拟机提供。为确保正确使用实验环境，`classroom` 和 `bastion` 应始终处于运行状态。

**注意**

登录 `servera` 或 `serverb` 时，可能会看到有关激活 `cockpit` 的消息。该消息可以忽略。

```
[student@workstation ~]$ ssh student@serverb
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of
known hosts.
Activate the web console with: systemctl enable --now cockpit.socket

[student@serverb ~]$
```

## 控制您的系统

在讲师主导的培训课堂上，为学员分配物理计算机 (`foundationX.ilt.example.com`)，用以访问他们在该主机上运行的虚拟机。学员以 `kiosk` 用户身份并使用密码 `redhat` 自动登录物理计算机。您操作的课堂系统就是在该主机上运行的虚拟机。

在 `foundationX` 上，使用一个名为 `rht-vmctl` 的特殊命令来操作虚拟机。下表中的命令应当以 `kiosk` 用户身份在 `foundationX` 上运行，可以用于这些虚拟机中的任何一台。

### `rht-vmctl` 命令

操作	命令
启动 <code>server</code> 计算机	<code>rht-vmctl start server</code>
查看“物理控制台”，以进行登录并操作 <code>server</code> 计算机	<code>rht-vmview view server</code>
将 <code>server</code> 计算机重置为之前的状态，并重新启动虚拟机	<code>rht-vmctl reset server</code>

开始实验室练习时，如果显示“reset your `servera`”的说明，则表示您应当在 `foundationX` 系统上以 `kiosk` 用户身份运行 `rht-vmctl reset servera` 命令。类似地，如果显示“reset your `workstation`”说明，则表示您应该在 `foundationX` 上以 `kiosk` 用户身份运行 `rht-vmctl reset workstation` 命令。

# 国际化

## 按用户选择语言

您的用户可能希望使用不同于系统范围默认值的语言显示其桌面环境。他们也可能想要将不同的键盘布局或输入法用于自己的帐户。

### 语言设置

在 GNOME 桌面环境中，用户可能会在第一次登录时看到有关设置首选语言和输入法的提示。如果没有，则个人用户调整首选语言和输入法设置的最简单方式为使用 **Region & Language** 应用。

您可以通过两种方式启动此应用。您可以从终端窗口运行命令 **gnome-control-center region**，或者从顶部栏右侧的系统菜单中，选择菜单左下方的设置按钮（其图标为一个十字螺丝刀加扳手）。

在打开的窗口中，选择 **Region & Language**。单击 **Language** 框，然后从出现的列表中选择首选语言。这也会将 **Formats** 设置更新为该语言的默认值。您下一次登录时，这些更改将全部生效。

这些设置会影响 GNOME 桌面环境以及在其中启动的任何应用，如 **gnome-terminal** 等。但是，从远程系统通过 **ssh** 登录访问该帐户时，或者通过从虚拟控制台（如 **tty5**）以基于文本的方式登录访问时，将不应用这些设置。



#### 注意

您可以使 shell 环境使用与图形环境相同的 **LANG** 设置，即使通过基于文本的虚拟控制台或 **ssh** 登录时也如此。其中一种实现方式是在您的 **~/.bashrc** 文件中放入类似如下的代码。此示例代码会将文本登录时所用的语言设置为与当前针对用户 GNOME 桌面环境设置的语言相匹配：

```
i=$(grep 'Language=' /var/lib/AccountsService/users/${USER} \
    | sed 's/Language=//')
if [ "$i" != "" ]; then
    export LANG=$i
fi
```

基于文本的虚拟控制台中可能无法正确显示日语、韩语、中文和其他包含非拉丁字符集的语言。

在命令行上设置 **LANG** 变量，可使个别命令使用其他语言：

```
[user@host ~]$ LANG=fr_FR.utf8 date
jeu. avril 25 17:55:01 CET 2019
```

后续命令将恢复为使用系统的默认语言进行输出。**locale** 命令可用于确定 **LANG** 的当前值，以及其他相关的环境变量。

## 输入法设置

红帽企业 Linux 7 中的 GNOME 3 自动使用 **IBus** 输入法选择系统，用户可以轻松、快速地更改键盘布局和输入法。

**Region & Language** 应用也可用于启用其他输入法。在 **Region & Language** 应用窗口中，**Input Sources** 框显示当前可用的输入法。默认情况下，**English (US)** 可能是唯一可用的输入法。突出显示 **English (US)**，然后单击**键盘**图标，查看当前的键盘布局。

要添加其他输入法，请单击 **Input Sources** 窗口左下方的 **+** 按钮。此时将打开 **Add an Input Source** 窗口。选择语言，然后选择首选的输入法或键盘布局。

配置了多种输入法时，用户可以通过按 **Super+Space**（有时称为 **Windows+Space**）快速切换。状态指示器也将显示在 GNOME 顶栏中，该顶栏有两个功能：其指示哪个输入方法处于活动状态；并且充当一个菜单，用于切换输入方法或者选择更复杂输入方法的高级功能。

一些输入法标有齿轮，这表示这些输入法具有高级配置选项和功能。例如，日语的 **Japanese (Kana Kanji)** 输入法允许用户预先用拉丁字母编辑文本，然后用 **向下箭头** 和 **向上箭头** 键选择要使用的正确字符。

美式英语用户也可利用此功能。例如，**English (United States)** 下的键盘布局是 **English (international AltGr dead keys)**，它将 PC 104/105 键版本键盘上的 **AltGr**（或右侧 **Alt**）视为“第二换挡”修饰符键和死键激活键，用于输入其他字符。另外，也有 Dvorak 和其他布局供用户选用。



### 注意

如果您知道字符的 Unicode 代码点，您可在 GNOME 桌面环境中输入任意 Unicode 字符。只需键入 **Ctrl+Shift+U**，再输入代码点即可。按 **Ctrl+Shift+U** 后将显示带有下划线的 **u**，表示系统正在等待 Unicode 代码点输入。

例如，小写希腊字母 lambda 的代码点为 U+03BB，可以通过按 **Ctrl+Shift+U**，然后按 **03BB**，再按 **Enter** 来输入。

## 系统范围的默认语言设置

系统的默认语言设为美国英语，使用 Unicode 的 UTF-8 编码作为其字符集 (**en\_US.utf8**)，但可以在安装期间或之后更改此设置。

在命令行中，**root** 用户可以通过 **localectl** 命令更改系统范围的区域设置。如果不带参数运行 **localectl** 命令，它会显示当前的系统范围区域设置。

要设置系统范围的默认语言，请运行 **localectl set-locale LANG=locale** 命令，其中 **locale** 是本章“语言代码参考”表中 **LANG** 环境变量的相应值。此更改存储在 **/etc/locale.conf** 中，将在用户下一次登录时生效。

```
[root@host ~]# localectl set-locale LANG=fr_FR.utf8
```

在 GNOME 中，管理员用户通过单击窗口右上角的 **Login Screen** 按钮，即可在 **Region & Language** 中更改此设置。更改图形登录屏幕的 **Language**，也会调整存储在 **/etc/locale.conf** 配置文件中的系统范围默认语言设置。

**重要**

与在图形环境中运行的虚拟控制台中的终端或 **ssh** 会话的伪终端相比，**tty4** 等基于文本的虚拟控制台在可以显示的字体方面有更多限制。例如，基于文本的虚拟控制台中的日语、韩语和中文字符显示可能会不正常。出于这个原因，您应该考虑将英语或使用拉丁字符集的其他语言用作系统范围默认设置。

类似地，基于文本的控制台所支持的输入法受到较多的限制，而且其管理也与图形桌面环境分离。可以通过 **localectl** 为基于文本的虚拟控制台和图形环境配置可用的全局输入设置。有关详细信息，请参见 **localectl(1)** 和 **vconsole.conf(5)** man page。

## 语言包

称为语言包的特殊 RPM 软件包可安装增添特定语言支持的语言数据包。这些语言包使用依赖项来自动安装其他 RPM 软件包，其中包含系统上其他软件包的本地化版本、字典和翻译。

要列出已安装和可以安装的语言包，请使用 **yum list langpacks-\***：

```
[root@host ~]# yum list langpacks-*
Updating Subscription Management repositories.
Updating Subscription Management repositories.
Installed Packages
langpacks-en.noarch      1.0-12.el8          @AppStream
Available Packages
langpacks-af.noarch       1.0-12.el8          rhel-8-for-x86_64-appstream-rpms
langpacks-am.noarch       1.0-12.el8          rhel-8-for-x86_64-appstream-rpms
langpacks-ar.noarch       1.0-12.el8          rhel-8-for-x86_64-appstream-rpms
langpacks-as.noarch       1.0-12.el8          rhel-8-for-x86_64-appstream-rpms
langpacks-ast.noarch      1.0-12.el8          rhel-8-for-x86_64-appstream-rpms
...output omitted...
```

要添加语言支持，请安装相应的语言包软件包。例如，以下命令添加对法语的支持：

```
[root@host ~]# yum install langpacks-fr
```

使用 **yum repoquery --whatsonplements** 来确定哪些 RPM 软件包可以通过语言包来安装：

```
[root@host ~]# yum repoquery --whatsonplements langpacks-fr
Updating Subscription Management repositories.
Updating Subscription Management repositories.
Last metadata expiration check: 0:01:33 ago on Wed 06 Feb 2019 10:47:24 AM CST
glibc-langpack-fr-0:2.28-18.el8.x86_64
gnome-getting-started-docs-fr-0:3.28.2-1.el8.noarch
hunspell-fr-0:6.2-1.el8.noarch
hyphen-fr-0:3.0-1.el8.noarch
libreoffice-langpack-fr-1:6.0.6.1-9.el8.x86_64
man-pages-fr-0:3.70-16.el8.noarch
mythes-fr-0:2.3-10.el8.noarch
```

**重要**

语言包软件包使用 RPM 弱依赖项，这样，只有在安装了需要补充软件包的核心软件包时，才会安装补充软件包。

例如，在安装前面示例所示的 langpacks-fr 时，只有系统上也安装了 mythes 词库，才会安装 mythes-fr 软件包。

如果之后将 mythes 安装到该系统上，则会自动安装 mythes-fr 软件包，因为它与已安装的 langpacks-fr 软件包存在弱依赖关系。

**参考文献**

**locale(7)**、**localectl(1)**、**locale.conf(5)**、**vconsole.conf(5)**、**unicode(7)** 和 **utf-8(7)** man page

有关图形桌面环境 X11 布局的名称和 **localectl** 中对应名称之间的转换，请参见 **/usr/share/X11/xkb/rules/base.lst** 文件。

## 语言代码参考

**注意**

此表可能无法反映系统上可用的所有语言包。使用 **yum info langpacks-SUFFIX** 获取有关任何特定语言包的更多信息。

### 语言代码

语言	语言包后缀	\$LANG 值
英语（美国）	en	en_US.utf8
阿萨姆语	as	as_IN.utf8
孟加拉语	bn	bn_IN.utf8
简体中文	zh_CN	zh_CN.utf8
繁体中文	zh_TW	zh_TW.utf8
法语	fr	fr_FR.utf8
德语	de	de_DE.utf8
古吉拉特语	gu	gu_IN.utf8
印地语	hi	hi_IN.utf8
意大利语	it	it_IT.utf8
日语	ja	ja_JP.utf8
埃纳德语	kn	kn_IN.utf8

语言	语言包后缀	\$LANG 值
韩语	ko	ko_KR.utf8
马拉雅拉姆语	ml	ml_IN.utf8
马拉地语	mr	mr_IN.utf8
奥里亚语	or	or_IN.utf8
葡萄牙语（巴西）	pt_BR	pt_BR.utf8
旁遮普语	pa	pa_IN.utf8
俄语	ru	ru_RU.utf8
西班牙语	es	es_ES.utf8
泰米尔语	ta	ta_IN.utf8
泰卢固语	te	te_IN.utf8

## 章 1

# 提高命令行生产率

### 目标

通过使用 Bash shell 的高级功能、shell 脚本以及红旗企业 Linux 所提供的各种实用程序，可以更高效地运行命令。

### 培训目标

- 通过编写简单的 shell 脚本自动执行命令序列。
- 使用 for 循环和条件，借助脚本中的项目列表或通过命令行高效地运行命令。
- 使用 **grep** 命令和正则表达式，查找与日志文件和命令输出中的模式相匹配的文本。

### 章节

- 编写简单的 Bash 脚本（及引导式练习）
- 使用循环更高效地运行命令（及引导式练习）
- 使用正则表达式匹配命令输出中的文本（及引导式练习）

### 实验

提高命令行生产率

# 编写简单的 BASH 脚本

## 培训目标

学完本节后，您应能够通过编写简单的 shell 脚本来自动执行相应的命令序列。

## 创建和执行 BASH SHELL 脚本

许多简单、常见的系统管理任务都可以使用命令行工具来完成。更复杂的任务则通常需要将多个命令链接到一起并在它们之间互传结果。借助 Bash shell 环境和脚本编写功能，可以将 Linux 命令与 shell 脚本组合在一起，从而轻松地解决重复而困难的实际问题。

Bash shell 脚本最简单的形式是一个可执行文件，其中包含了命令列表，而且可能还包含用于控制整体任务决策的编程逻辑。写得较好的 shell 脚本自身就是一个强大的命令行工具，并可以供其他脚本利用。

精通 shell 脚本编写是在所有操作环境中成功进行系统管理的必备要素。shell 脚本编写的工作知识在企业环境中至关重要，因为脚本可以提高完成日常任务的效率和准确性。

您可以通过在文本编辑器中打开新的空文件来创建 Bash shell 脚本。尽管可以使用任何文本编辑器，但高级编辑器（如 **vim** 或 **emacs**）理解 Bash shell 语法并且可以提供颜色编码的高亮显示。这种高亮显示有助于识别常见错误，例如语法不正确、引号不成对，以及圆括号、大括号和中括号不成对等等。

## 指定命令解释器

脚本的第一行以符号 '#!' 开头，通常称为 **sh-bang** 或 **she-bang**，其源于 **sharp**（升半音符号）和 **bang**（感叹号）这两个字符的名称。这一特定的双字节魔法数字符号表示可解释性脚本；符号后面的话语为处理该脚本的行时所需要的正确命令解释器的完全限定文件名。要了解魔法数字在 Linux 中如何指示文件类型，请参阅 **file(1)** 和 **magic(5)** 的 man page。对于使用 Bash 脚本编写语法的脚本文件，shell 脚本第一行的开头如下：

```
#!/bin/bash
```

## 执行 Bash Shell 脚本

完成的 shell 脚本必须为可执行文件，能作为常规命令运行。使用 **chmod** 命令可添加执行权限，并且可能与 **chown** 命令组合以更改脚本的文件所有权。仅为脚本的目标用户授予执行权限。

如果将脚本放在 shell 的 **PATH** 环境变量中所列的某个目录下，则可以像其他命令那样单独用文件名来调用 shell 脚本。shell 会使用通过该文件名找到的第一个命令；应避免为 shell 脚本的文件名使用现有的命令名。或者，您可以通过在命令行上输入脚本的路径名来调用 shell 脚本。**which** 命令，后面跟可执行脚本的文件名，可以显示所要执行的命令的路径名。

```
[user@host ~]$ which hello  
~/bin/hello  
[user@host ~]$ echo $PATH  
/home/user/.local/bin:/home/user/bin:/usr/share/Modules/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

## 对特殊字符加引号

一些字符和词语对 Bash shell 具有特殊含义。但是，偶尔您会希望使用这些字符的字面值，而非其特殊含义。为此，请使用以下三种工具之一来取消（或转义）特殊含义：反斜杠 (\)、单引号 ('') 或双引号 ("")。

反斜杠转义字符可取消紧跟在该字符后面的单个字符的特殊含义。例如，要使用 **echo** 命令显示字面值字符串 **# not a comment**，Bash 不能将 # 符号解释为具有特殊含义。请在 # 符号的前面加上反斜杠字符。

```
[user@host ~]$ echo # not a comment
[user@host ~]$ echo \# not a comment
# not a comment
```

当需要转义文本字符串中的多个字符时，可以多次使用转义字符或者采用单引号 ('')。单引号保留其括起的所有字符的字面含义。请注意所用的转义字符和单引号：

```
[user@host ~]$ echo # not a comment #
[user@host ~]$ echo \'# not a comment \'#
# not a comment
[user@host ~]$ echo \'# not a comment \'#
# not a comment #
[user@host ~]$ echo '# not a comment #'#
# not a comment #
```

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。请观察以下所用的各种引号的示例。

使用单引号则可以按字面解译所有的文本。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。问号 (?) 也是一个需要防止扩展的元字符。

```
[user@host ~]$ var=$(hostname -s); echo $var
host
[user@host ~]$ echo "***** hostname is ${var} *****"
***** hostname is host *****
[user@host ~]$ echo Your username variable is \$USER.
Your username variable is $USER.
[user@host ~]$ echo "Will variable $var evaluate to $(hostname -s)?"
Will variable host evaluate to host?
[user@host ~]$ echo 'Will variable $var evaluate to $(hostname -s)?'
Will variable $var evaluate to $(hostname -s)?
[user@host ~]$ echo "\"Hello, world\""
"Hello, world"
[user@host ~]$ echo '"Hello, world"'
"Hello, world"
```

## 从 Shell 脚本提供输出

通过将文本作为参数传递给命令，**echo** 命令可显示任意文本。默认情况下，文本将在标准输出 (STDOUT) 上显示，但是也可以使用输出重定向将其定向到标准错误 (STDERR)。在下面的简单 Bash 脚本中，**echo** 命令向 STDOUT 显示消息 “Hello, world”。

```
[user@host ~]$ cat ~/bin/hello  
#!/bin/bash  
  
echo "Hello, world"  
  
[user@host ~]$ hello  
Hello, world
```



### 注意

该用户只需在提示时运行 **hello** 即可，因为 **~/bin** (**/home/user/bin**) 目录就在用户的 **PATH** 变量中，而其中的 **hello** 脚本为可执行文件。只要 **PATH** 中位于 **/home/user/bin** 之前的目录内没有其他名为 **hello** 的可执行文件，shell 就会自动到那里查找脚本。

**echo** 命令在 shell 脚本中被广泛用来显示信息或错误消息。这些消息可能是有用的脚本进度指示符，并且可以定向到标准输出、标准错误或者重定向到日志文件以进行存档。显示错误消息时，一种良好的做法是将它们定向到 **STDERR** 以便更容易区分错误消息与正常状态消息。

```
[user@host ~]$ cat ~/bin/hello  
#!/bin/bash  
  
echo "Hello, world"  
echo "ERROR: Houston, we have a problem." >&2  
  
[user@host ~]$ hello 2> hello.log  
Hello, world  
[user@host ~]$ cat hello.log  
ERROR: Houston, we have a problem.
```

当尝试调试有问题的 shell 脚本时，**echo** 命令可能也很有帮助。向行为不同于预期的脚本部分中添加 **echo** 语句可帮助阐明正在执行的命令以及正在调用的变量的值。



### 参考文献

**bash(1)**、**magic(5)**、**echo(1)** 和 **echo(1p)** man page。

## ► 指导练习

# 编写简单的 BASH 脚本

在本练习中，您将编写一个包含一系列命令的简单 Bash 脚本，并通过命令行运行它。

## 成果

您应能够：

- 编写并执行一个简单的 Bash 脚本。
- 将简单 Bash 脚本的输出重定向到文件。

## 在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 `lab console-write start` 命令。该命令将运行一个起始脚本，它将确定 `servera` 计算机是否可从网络访问。如果不可用，脚本会提醒您。此外，它还会根据需要安装 vim-enhanced 软件包。

```
[student@workstation ~]$ lab console-write start
```

- 1. 从 `workstation`，以 `student` 用户身份打开连接 `servera` 的 SSH 会话。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 2. 创建并执行一个简单的 Bash 脚本。

- 2.1. 使用 `vim` 文本编辑器在您的主目录下创建一个新的文本文件，并将其命名为 `firstscript.sh`

```
[student@servera ~]$ vim firstscript.sh
```

- 2.2. 插入以下文本并保存文件。请注意，井号 (#) 的数量是任意的。

```
#!/bin/bash  
echo "This is my first bash script" > ~/output.txt  
echo "" >> ~/output.txt  
echo ##### >> ~/output.txt
```

- 2.3. 使用 `sh` 命令执行脚本。

```
[student@servera ~]$ sh firstscript.sh
```