免费访问交互式UNIX培训网站

PRENTICE HALL PTR interactive

实践大师：UNIX awk和sed编程篇

UNIX awk and sed PROGRAMMER'S INTERACTIVE WORKBOOK

[美] PETER PATSIS 著
吴中华　陈兴志　陈　红　等译
吴中华　审校

- 精通awk和sed编程
- 内容包含grep和正则表达式
- 特别适合UNIX初学者和中级使用者
- 通过实践、练习和项目进行学习

PTR PH
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.com.cn

00004903

实践大师：UNIX awk 和 sed 编程篇

UNIX awk and sed PROGRAMMER'S INTERACTIVE WORKBOOK

[美] PETER PATSIS 著

吴中华 陈兴志 陈 红 等译

吴中华 审校

電子工業出版社

Publishing House of Electronics Industry

Authorized translation from the English language edition published by Prentice-Hall, Inc.

Copyright © 1998

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry, China.
Copyright © 2000

本书中文简体专有翻译出版权由美国 Prentice-Hall, Inc. 授予电子工业出版社。该专有出版权受法律保护。

图书在版编目(CIP)数据

实践大师:UNIX awk 和 sed 编程篇/(美)帕特西斯(Patsis. P. A.)著;吴中华等译. - 北京:电子工业出版社,2000.1
书名原文:UNIX awk and sed PROGRAMMER'S INTERACTIVE WORKBOOK
ISBN 7-5053-5632-1

Ⅰ.实… Ⅱ.①帕… ②吴… Ⅲ.UNIX 操作系统 - 基本知识 Ⅳ.TP316.81

中国版本图书馆 CIP 数据核字(1999)第 74138 号

书　　名:**实践大师:UNIX awk 和 sed 编程篇**
原 书 名:UNIX awk and sed PROGRAMMER'S INTERACTIVE WORKBOOK
著　　者:[美]PETER PATSIS
译　　者:吴中华　陈兴志　陈　红　等
审 校 者:吴中华
策　　划:电子工业出版社外版书编辑部
责任编辑:吴　源
特约编辑:荆显英
排版制作:电子工业出版社计算机排版室监制
印 刷 者:
装 订 者:北京科技印刷厂
出版发行:电子工业出版社　URL:http://www.phei.com.cn
　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销:各地新华书店
开　　本:787×1092　1/16　印张:26.25　字数:636 千字
版　　次:2000 年 1 月第 1 版　2000 年 1 月第 1 次印刷
书　　号:ISBN 7-5053-5632-1
　　TP·2886
印　　数:4000 册　定价:50.00 元
版权贸易合同登记号　图字:01-1999-3361

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。
若书店售缺,请与本社发行部联系调换。电话　68279077

译者的话

您知道正则表达式吗？您用过 grep 吗？

您知道 sed 和 awk 是干什么的吗？

对以上问题的回答中有一个如不知道或不清楚，那我认为您应该买下这本书，除非您对 UNIX 不感兴趣——但我相信您既然已拿起了这本书，您一定对 UNIX 有一点感情——或者您就能写出这样一本书——那我也认为您有必要买，看看本书作者是如何写书，也许您有一天也会把您的经验写成书，造福广大 UNIX 爱好者！

如果您还犹豫不决，那么就看看本书的内容再决定吧。本书介绍了 3 个 UNIX 工具——grep，sed 和 awk。它们几乎和 UNIX 具有相同的悠久历史，并一直被广泛使用。第 1 章和第 2 章先讲述了 3 个工具都要用到的正则表达式和元字符。UNIX 工具功能强大的一个重要原因是有正则表达式和元字符，所以学 UNIX 不可不知它们兄弟俩。第 3 章和第 4 章介绍了一个非常实用的工具 grep。grep 在查找方面具有强大的功能，和 UNIX 的其他工具配合使用，几乎可找到所有您想找的东西。第 5 章到第 8 章详细讲述了 sed(流编辑器)，功能比 grep 强大，使用也比 grep 复杂。sed 除了查找功能外还有替换、插入和添加文本等功能，是一个功能实用、完备的流编辑器。流编辑器的最大特点是在您不能使用可视化编辑器时使用它，例如可以在 shell 脚本程序中使用或者文件太大无法用可视化编辑器装载时使用等。3 个工具中功能最强大的要数 awk，本书剩余一半的章节讲述了 awk。awk 的强大使之变成了一门计算机语言。它包含了表达式，语句，常量，变量，函数和流程控制等基本语言元素。简而言之，它能完成您想完成的文本处理工作。

本书特点是有大量的练习题和自测题。练习题覆盖了各种实际使用中会出现的情况。练习题的答案可帮助您更好地理解本书内容。自测题可让您掌握学习进度。用作者的话说是："我必须强调学习计算机语言和工具的最好方法是练习实践。"每一章最后是思考题，它可让您充分理解并扩充所学内容。与本书对应还有一个 Internet 网站。在那儿您可以和其他学习者交流学习体会和问题，当然那是英文的，我们就不负责翻译了。

参加翻译和审校工作的同志有：吴中华、陈兴志、陈红、佟继周、李天昊和杨震。

翻译书是件很繁琐的事。时间的紧迫和译者水平的限制，使本书难免有不当或错误之处，敬请读者和专家批评指正。

译　者

总编的话

Prentice Hall 出版社的《实践大师》丛书提供的正是您需要的。它帮助您迅速提高 UNIX 水平，操作如飞。

相信通过阅读本丛书，您会找到一条学习的捷径。每一本丛书的每一章开头都有一个简单明了的学习目标。每一章的核心是一系列的实践，而每一个实践用练习的形式教您特定的技能。请在计算机上完成这些练习题，并回答指定的问题。问题的答案将引导您更进一步地探讨。每一个实践的末尾还有多项选择的自测题，可进一步巩固刚学过的知识。每章的最后还准备了思考题，这些思考题综合了本章中出现的各种技能，是对您学习的挑战。

我们的目标是使学习过程紧凑有效，使您更优秀。

本丛书并不是独立的，因为每一本书都有各自的“伴侣网站”。在网站上有关于本丛书中讨论的概念的更详细信息；还有额外的自测题，可加深对学习材料的理解；可能最重要的是在网站上您可以和其他的丛书用户交流各种技能和学习体会。

本丛书的所有伴侣网站都位于：

http://www.phptr.com/phptrinteractive/

Mark L. Taub
Editor-in-Chief
Prentice Hall PTR Interactive

试读结束：需要全本请在线购买： www.ertongbook.com

引　言

本书讲述了三个 UNIX 工具：grep、sed 和 awk。用这三个工具可写出各种简洁的应用程序。这些工具几乎和 UNIX 操作系统有相同的悠久历史，直至今日仍被广泛用来解决各种任务。grep、awk 和 sed 在处理文件时非常有用；以命令行方式查找文件内容；和其他 UNIX 工具一起以命令行方式完成共同的任务；或者写一些简短的脚本(script)程序解决一个应用题。这仅仅是这三个工具能解决的一小部分任务。本书的目标是介绍这三个工具并让您成为使用 grep、sed 和 awk 的专家，方便快速地解决需要完成的应用。另外，在阅读本书的过程中，当您在有一个应用题需要完成时，希望您能充分理解这三个工具并考虑使用它们，这样既解决了您的应用题，又完成了一个问题的解答方案。为了达到这个目标，需要学习每一个工具的原理。每一个工具的概念和原理通过特定的语法、行为、规则和每个概念及原理的细微差别来描述。练习和练习讨论将提高和增强您对原理和概念的理解。

三个工具有它们共有的特性。它们都能利用标准输入、标准输出或用户指定的文件完成工作。它们还可以和 UNIX 环境之间通过管道来实现交互运作。它们都可以使用正则表达式。

本书是如何组织的

本书首先讲述正则表达式，因为三个工具都需要用到它们。如果您不熟悉正则表达式，有关章节将介绍它们，学习它们要用到的各种元字符。虽然这部分涉及的范围较宽，但不打算向读者提供关于这方面的高级知识。但是学习它们对初学者或者是中等水平的用户来说是非常有益的。无论何时，包括在讲述 grep、sed 和 awk 以及使用正则表达式时，都假设您以前还没有遇到过正则表达式。如果您已经接触过正则表达式，那么您就可以忽略这些材料中的部分或全部内容。

下一部分描述 grep。grep 是一个最适合于文件查找的工具；因此最先讲述使用正则表达式的 grep。接下来将讲述 grep 和管道以及和其他 UNIX 程序的连接。最后讨论运行 grep 时需要用到的各种选项。

本书的下一部分将描述 sed，包括 sed 定址，以及 sed 中要用到的各种 sed 内部命令。sed 命令提供了比 grep 更强大的功能，可以用来完成更多的应用。最后还将讨论更多的 sed 高级命令，例如多行模式空间等。

awk 放在最后讲述。awk 与 grep 和 sed 不同，它比一个工具或者特定语言有更接近于一般语言的特征。原因是它拥有比一般语言更多的结构和特性。它们包括：

- 控制流转移到程序任何部分的能力
- 在用户自定义的变量中存储值的能力
- 执行算术运算的能力
- 编写函数的能力
- 以用户自定义格式执行输出的能力

在 sed 和 grep 中，或者简化了这些功能或者不包括它们。一般来说，大部分的书都把它当

作一个工具，一个采用 awk 特性的范例工具来讲述。在本书中把它当作一门语言而不是一个工具来讲述。因此，我们将讨论数据类型、变量、内部函数、数组、控制语句、输入/输出和函数等。把 awk 作为一个工具来讲述，并给出一些用 awk 解决问题的例子会产生一些问题。如果所有您需要解决的正是这些例子中出现过的，那么您马上就能用。但是，如果您需要的功能和所提供的不同，那么您必须从这些例子中自己去了解和理解该语言。比较好的办法是把它作为语言讲述，例子仅仅为了加深对语言的每个特性、使用规则和各种使用方法的理解。这种方法有多种好处。您不能期望使用 awk 的大部分方式。把它作为一门语言来学，帮助您更好地理解它的特性，也就达到了本书的目的，在解决问题时可更迅速地决定是否可以用 awk 来解决问题。

如果您把 awk 简单地作为工具来学习而不是作为语言，那么一旦发生了错误，尤其是通过作为范例工具学的，就很难确定哪里发生了错误。如果您理解了这种语言，通过查看有错误的程序就能很轻松地找到出错的地方，因为您知道语言结构是如何通过语句来组合的。很多人抱怨 awk 的错误信息系统的提示难于理解。我相信这个问题不是由于 awk 的错误信息系统——awk 的运行时错误信息系统的提示比 GNU C/C++ 的多得多——而是没有把 awk 当作一门语言。

各种语言都基于非常相似的概念。理解了一种语言能大大增强理解另一种语言的能力。理由是大多数语言共享了非常相似的原理和结构。例如函数、适用范围、强制转换、传值调用和引用调用(这些在本书中都会讲述)。如果您已经学过一门一般语言，例如 C，C++ 或者 JAVA，那么学习 awk 将简单多了。如果您没有学过一般语言，本书中提供的方法将帮助您学习一种新的语言。在 awk 中，和大多数语言一样，有一种确定的语言设计思想。一个重要的设计思想是正交性，简单地说就是必须存在一种以上表达一个动作的方式。正交性概念的意思是有多于一种语言结构(例如 FOR 循环和 WHILE 循环都可用来完成一个程序)能用来解答一个程序。正交性对程序员有好处，因为它给程序员提供了灵活性，在用语言解答任何给定问题时，可以寻找一种更有效，更紧凑并具有创造性的程序代码。虽然有多种结构可用来解答一个问题，但是对特定的程序有一种结构比其他的结构要好。真正精通一种语言实际表现为一种理解能力：什么时候用某种结构比用另一种结构更有效(使用更少的存储空间或更快的执行速度)，或更紧凑(使用更少的代码行或更便于阅读)。这需要大量实践。我希望在阅读本书时和读完本书后，您会练习写一些 awk 程序。还应该考虑用不只一个解答方法来解决一个问题，并写出 awk 程序来验证和比较那种方法更好(更易读、更短、更快和消耗更少的存储空间)。

我必须强调学习计算机语言和工具的最好方法是练习实践。尽量尝试提出您自己的疑问。另外，我发现有时最好的学习方法是利用自己所产生的错误。找出出错的地方，并研究出错的原因。最终您会有意想不到的惊喜。

本书适用读者

本书不是一本关于计算机语言的书。虽然本引言中提到了语言的细节和思想，不管您以前是否遇到过计算机语言，或是否学过计算机语言编程都不重要。而且，您也不需要对 UNIX 操作系统有很深的了解。本书是关于 grep，sed 和 awk 的编程语言教程。本书适用于没有用

过 grep,sed 和 awk 的 UNIX 的初学者。它也适用于使用过 grep,sed 和 awk,但还不精通的 UNIX 中级甚至高级用户。当然需要有如何从命令行方式执行程序的基本知识。

关于伴侣网站

本书的伴侣网站位于:

http://www.phptr.com/phptrinteractive/

此网站由多个模块组成,目的是帮助您加快揭开 grep,sed 和 awk 的神秘面纱。您会发现有一个学生休闲室,在那儿您会遇到本丛书的其他读者,并能分享各种小窍门和程序。还有一个作者角,在那儿您会找到本书的补充材料和作者的公告,勘误表和其他。专门有一个模块是关于本书各章"思考题"的答案。还有一个模块是有助于加深理解本书中提到的概念的"自测题"。

请经常访问此网站以共享和讨论您的答案。

作 者 简 介

Peter A. Patsis 是数字设备公司(DEC)的高级软件工程师,为检验该公司开发的 Alpha AXP 微处理器编写工具软件。他在位于 Potsdam 的纽约州立大学获得计算机科学学士学位,在位于 Atlanta Georgia 的 Georgia 理工学院获计算机科学硕士学位。他曾在 Smithsonian 研究所工作,任软件工程师,和 NASA,MIT 以及 TRW 合作从事 AXAF 计划。

他已经在 UNIX 操作系统方面工作了十多年。

目　录

试读结束：需要全本请在线购买： www.ertongbook.com

第 1 章　正则表达式和元字符

对正则表达式可能都会有这样的看法：每个人都知道它们，但不是每个人都能用好它们。

本章目标

在本章中，您将学到如下内容：

√句点和反斜杠元字符的使用

如果您以前没有接触过正则表达式，本章将帮助您理解如何正确并高效地使用正则表达式。如果您在 UNIX Shell 中接触并使用过正则表达式，请注意在 Shell 中正则表达式的用法与 grep，awk 和 sed 中的用法是有区别的。本章并不直接讲述这些区别，而是讨论并让您理解在 grep，sed 和 awk 中如何使用正则表达式。在任何需要的地方我们会讲述在 grep 与 sed 中使用正则表达式和在 awk 与 grep 中使用正则表达式的区别。即使您觉得在 grep，awk 和 sed 中会使用正则表达式，我们还是建议您回答一下后面的问题，做一做练习，以回顾一下这方面的内容。

实践 1.1 使用句点和反斜杠元字符

实践目标

在本实践中,您将学到如下内容:

√识别正则表达式中的运算符和运算数
√理解运算数和运算符组成的正则表达式的结果
√区别何时计算正则表达式
√理解把表达式当作模式计算的结果
√理解通配符元字符

本实践简单介绍什么叫正则表达式,并列出本书中要用到的所有元字符。学习正则表达式的最好办法是:学习元字符执行的功能和它们在表达式中使用的结果。因此最好的办法是通过例子来学习。在后面的章节中有大量使用了各种元字符的正则表达式的例子。

运算数和运算符

就像算术表达式一样,一个正则表达式中也包含运算符和运算数。因此可以把正则表达式理解为一个包含运算数和运算符的表达式。在算术表达式中运算数是数字,运算符是加号(+),减号(-)等。而在正则表达式中,运算数是字符串,运算符是各种元字符。

■ 实例

考虑如下正则表达式:

```
chapter *[0-9]+
```

运算数是字符串"chapter"、0 和 9。运算符是 * 、[]、- 和 + 等元字符。就像算术表达式一样,运算符用运算数来执行某种计算并产生某个结果。

正则表达式和算术表达式不同,算术表达式通常只产生一个值,而正则表达式的结果是一个子字符串列表,此子字符串列表是长度为 1(即一个字符)到计算机能存储的最大长度、包含任何可打印字符的所有子字符串的子集。因此,上述正则表达式的计算结果是如下所示的字符串列表:

```
{chapter 0,chapter 1,chapter 2,chapter 3,chapter 4,...,chapter 9,chapter0,...,chapter9}
```

这里,逗号之间的每个值是从表达式中计算得到的子字符串。因此,可以把正则表达式理解为一种确定子字符串子集的表示法,或等效于一个子字符串列表。换句话说,在我们的例子中,表示法 chapter *[0-9]+ 确定了一个子字符串列表:它们以文本 chapter 开头,接下来是一个可选的空格,最后是 0 到 9 之间的任意一个数字。

现在,让我们来考虑如下正则表达式:

```
yes
```

计算后,这个表达式的结果是一个单一的值:

```
yes
```

这个正则表达式生成一个由“y”、“e”和“s”组成的长度为 3 个字符的单一子字符串。在这些例子中,单一元素(在上例中,单一元素是 yes)是所有可以作为正则表达式的子字符串子集中的一个。

当一个正则表达式中包含元字符运算符时,此正则表达式叫元字符正则表达式,其结果包含多个子字符串。只有一个单一元素结果的正则表达式叫做文字正则表达式,也就是说它不包括元字符。在上述例子中,正则表达式 chapter *[0-9]+ 包含多个元字符,可看到其结果有多个子字符串。正则表达式 yes 中没有任何元字符,结果为单一子字符串 yes。因此正则表达式中最简单的表示法是确定一个单一元素,此单一元素是所有可能的子字符串中的一个或者是一个文字正则表达式。

但是把正则表达式简单理解为选择子字符串,那仅仅只是讲了一半,还没有说明它的用途和使用方法。正则表达式的结果用来匹配一组字符串的。被用来匹配的一组字符串通常是一行文本。

■ 实例

考虑如下正则表达式:

```
her
```

和下列文本行:

```
feathered
```

文字正则表达式 her 会匹配字符串 feathered(因为在 feathered 中的第 5 个字符处包含了字符串 her)。务必理解这个观点:正则表达式计算后得到的一个字符串或一组字符串都是用来匹配另外的字符串的(说一个字符串或一组字符串,是因为在使用元字符时,有多于一个的字符串结果)。同时必须知道,her 严格地和 her 匹配,不管它是否处于更长的子字符串中,例如在 feathered 中。

现在我们知道了正则表达式是用来匹配另一个子字符串的。接下来该让我们看看哪里需要用到正则表达式,什么时候正则表达式会被计算。在讲述每个工具(grep,sed 和 awk)时,我们会更详细地看到正则表达式用在什么地方。但是现在不详细展开讲,只举两个例子以便更清楚地理解它们的定义。在 grep 中,通常希望从一个输入文件的文本行中查找是否出现特定的字符串。因此在 grep 中,要查找的特定字符就是一个正则表达式,匹配字符串是文件中的每一行。

■ 实例

考虑如下行:

```
grep her input-file
```

这里,grep 把文字正则表达式 her 确定为一个正则表达式匹配参数,用来和“input-file”文件中的每一行比较。如果出现匹配,那么将整行打印出来。文字正则表达式被作为一个表达

试读结束:需要全本请在线购买:　www.ertongbook.com