UNIX Power Tools

New Lay-flat Binding

sed & awk

A NUTSHELL HANDBOOK

by Dale Dougherty

O'Reilly & Associates, Inc.

sed & awk

Dale Dougherty

O'Reilly & Associates, Inc.
103 Morris Street, Suite A
Sebastopol, CA 95472

sed & awk
by Dale Dougherty

Editor: Tim O'Reilly

Printing History:

November 1990:	First edition.
March 1991:	Minor corrections.
July 1992:	Minor corrections.
November 1992:	Minor corrections.

This book is printed on acid-free paper with 50% recycled content, 10-15% post-consumer waste. O'Reilly & Associates is committed to using paper with the highest recycled content available consistent with high quality.

ISBN: 0-937175-59-5 [6/93]

The following is a list of the programs in the book, with a brief description of each:

README — this file.
acronyms — program to read and substitute acronyms; Chapter 7 & 8.
awkro — acronym processor from Chapter 8.
bitmap.awk — awk script that demonstrates multi-dimensional arrays; Chapter 8.
contrib/ — subdirectory containing programs from Chapter 12.
contrib2/ — subdirectory containing additional contributed programs.
date-month — convert numeric month to name of month; Chapter 8.
do.outline.awk — awk script to create numbered outline.
do.outline.sed — sed script to extract outline from document; Chapter 4.
factorial — program to derive factorials; Chapter 8.
filesum — program to list size of files in directory; Chapter 7.
getmac — program for extracting troff macro definitions; Chapter 4.
glossary — sample glossary file for lookup program.
grades.awk — script to average student grades; Chapter 8.
gres — sed script for command line substitution command; Chapter 3.
index.edit — sed script for correcting index entries; Chapter 5.
index/ — subdirectory that contains index programs in Chapter 11.
invoke — nawk menu-based command generator; Chapter 10.
lookup — program to read glossary file and print definition; Chapter 8.
lotto — quick-pick lottery program; Chapter 9.
match — nawk program that outputs string matched by regular expresssion; Chapter 9.
phone — nawk program that looks up phone number; Chapter 8.
phones.data — data file for phone program.
phrase — sed script to match more than one word over more than one line; Chapter 6.
romanum — decimal to Roman numeral conversion; Chapter 8.
runsed — shell script to invoke sed; Chapter 4.
sedman — sed script to strip nroff'd man pages; Chapter 5.
soelim.awk — nawk script that simulates soelim program; Chapter 10.
spellcheck — shell script that involves spellcheck.awk; Chapter 11.
spellcheck.awk — nawk script for interactive spellchecker; Chapter 11.
words — awk program for stripping macros prior to word count; Chapter 10.

Preface

This book is about a set of uncommonly named UNIX utilities, **sed** and **awk**. These utilities have many things in common, including the use of regular expressions for pattern matching. Since pattern matching is such an important part of the use of these utilities, this book explains UNIX regular expression syntax very thoroughly. Because there is a natural progression in learning from **grep** to sed to awk, we will be covering all three programs, although the focus is on sed and awk.

Sed and awk are tools used by users, programmers, and system administrators —anyone working with text files. Sed, so called because it is a stream editor, is perfect for applying a series of edits to a number of files. Awk, named after its developers Aho, Weinberger, and Kernighan, is a programming language that permits easy manipulation of structured data and the generation of formatted reports. This book covers the original awk; the new awk, which became available as **nawk** starting with UNIX System V Release 3; and the Free Software Foundation GNU project's version of awk, **gawk**.

The focus of this book is on writing scripts for sed and awk that quickly solve an assortment of problems for the user. Many of these scripts could be called "quick-fixes." In addition, we'll cover scripts that solve larger problems that require more careful design and development.

Scope of this Handbook

Chapter 1, *Power Tools for Editing*, is an overview of the features and capabilities of sed and awk.

Chapter 2, *Understanding Basic Operations*, demonstrates the basic operations of sed and awk, showing a progression in functionality from sed to awk. Both share a similar command line syntax, accepting user instructions in the form of a script.

Chapter 3, *Understanding Regular Expression Syntax*, describes UNIX regular expression syntax in full detail. New users are often intimidated by these strange expressions, used for pattern matching. It is important to master regular expression syntax to get the most from sed and awk. The pattern-matching examples in this chapter largely rely on **grep** and **egrep**.

Chapter 4, *Writing Sed Scripts*, begins a three-chapter section on sed. This chapter covers the basic elements of writing a sed script using only a few sed commands. It also presents a shell script that simplifies invoking sed scripts.

Chapter 5, *Basic Sed Commands*, and Chapter 6, *Advanced Sed Commands*, divide the sed command set into basic and advanced commands. The basic commands are ones that parallel manual editing actions, while the advanced commands introduce simple programming capabilities. Among the advanced commands are those that manipulate the hold space, a set-aside temporary buffer.

Chapter 7, *Writing Scripts for Awk*, begins a four-chapter section on awk. This chapter presents the primary features of this scripting language. A number of scripts are explained, including one that modifies the output of the **ls** command.

Chapter 8, *Conditionals, Loops, and Arrays*, describes how to use common programming constructs such as conditionals, loops, and arrays.

Chapter 9, *Functions*, describes how to use awk's built-in functions as well as how to write user-defined functions (a feature introduced in new awk).

Chapter 10, *The Bottom Drawer*, covers a set of miscellaneous awk topics. It describes how to execute UNIX commands from an awk script and how to direct

output to files and pipes. It presents **awkcc**, an awk-to-C translator. This chapter also offers some (meager) advice on debugging awk scripts.

Chapter 11, *Full-featured Scripts*, presents two longer, more complex awk scripts that together demonstrate nearly all the features of the language. The first script is an interactive spell checker. The second script processes and formats the index for a book or a master index for a set of books.

Chapter 12, *A Miscellany of Scripts*, presents a number of user-contributed scripts that show different styles and techniques of writing scripts for sed and awk.

Appendix A, *Sed Quick Reference*, is a quick reference describing sed's commands and command-line options.

Appendix B, *Awk Quick Reference*, is a quick reference to awk's command-line options and a full description of its scripting language.

Appendix C, *Supplement for Chapter 11*, presents the full listings for the **spellcheck.awk** script and the **masterindex** shell script described in Chapter 11, *Full-featured Applications*.

Availability of Sed and Awk

Sed and awk were part of UNIX's seventh edition and have been part of the standard distribution ever since. Sed has been unchanged since it was introduced.

In 1985, the authors of awk extended the language, adding many useful features. Unfortunately, this new version remained inside AT&T for several years. Now it is finally part of AT&T's System V as of Release 3.1. It can be found under the name of nawk, for new awk; the older version still exists under its original name. Otherwise, source code is available for a licensing fee from AT&T's UNIX Toolchest.* The Free Software Foundation GNU project's version of awk, gawk, implements all the features of the new awk. It is freely available, although not technically in the public domain. If your UNIX system does not offer nawk, I highly recommend that you obtain gawk.†

*The UNIX Toolchest offers source code for a variety of programs. There is a registration fee for non-System V licensees and programs are individually priced. You can reach the UNIX Toolchest (in New Jersey) via modem at 908-522-6900; login as *guest*. In Europe, the number is 44-1-567-7711 (London). In the Far East, dial 81-3-431-3670 (Tokyo). The UNIX Toolchest also carries **awkcc**, an awk-to-C translator for producing compiled programs from awk scripts.

†For information on obtaining gawk, contact the Free Software Foundation, Inc., 675 Massachusetts Avenue, Cambridge, MA 02139. The telephone number is 617-876-3296.

NOTE

In this book, you can assume that what is true for nawk is true for gawk, unless gawk is explicitly called out. Scripts written for nawk are 100 percent compatible with gawk. There are a few areas where gawk has introduced gawk-specific features; however, recent versions of nawk support many of these features, suggesting that the remaining differences are really very minor.

DOS Versions

egrep, sed, and awk are available for MS-DOS-based machines as part of the MKS Toolkit (Mortice Kern Systems, Inc., Ontario, Canada). Their implementation of awk supports the features of nawk.

The MKS Toolkit also includes the Korn Shell, which means that many shell scripts written for the Bourne shell on UNIX systems can be run on a PC. While most users of the MKS Toolkit have probably already discovered these tools in UNIX, we hope that the benefits of these programs will be obvious to PC users who have not ventured into UNIX.

We have used a PC on occasion because Ventura Publisher is a terrific formatting package. One of the reasons we like it is that we can continue to use vi to create and edit the text files and use sed for writing editing scripts. We have used sed to write conversion programs that translate **troff** macros codes into Ventura stylesheet tags. We have also used it to insert tags in batch mode. This can save having to tag manually repeated elements in a file.

Sed and awk are also useful for writing conversion programs that handle different file formats.

Other Sources of Information About Sed and Awk

For a long time, the main source of information on these utilities was two articles contained in Volume 2 of the *UNIX Programmer's Guide*. The article *awk—A Pattern Scanning and Processing Language* (September 1, 1978) was written by the language's three authors. In ten pages, it offers a brief tutorial and discusses several design and implementation issues. The article *SED—A Non-interactive Text Editor* (August 15, 1978) was written by Lee E. McMahon. It is a reference that gives a full description of each function and includes some useful examples (using Coleridge's *Xanadu* as sample input). Homogenized versions of both these articles are presently distributed with most UNIX systems.

In trade books, the most significant treatment of sed and awk appears in *The UNIX Programming Environment* by Brian W. Kernighan and Rob Pike

(Prentice-Hall, 1984). The chapter entitled "Filters" not only explains how these programs work but shows how they can work together to build useful applications.

The authors of awk collaborated on a book describing the enhanced version: *The AWK Programming Language* (Addison-Wesley, 1988). It contains many full examples and demonstrates the broad range of areas where awk can be applied. It follows in the style of the *UNIX Programming Environment*, which at times makes it too dense for some readers who are new users

Most general introductions to UNIX introduce sed and awk in a long parade of utilities. Of these books, Henry McGilton and Rachel Morgan's *Introducing the UNIX System* offers the best treatment of basic editing skills, including use of all UNIX text editors.

UNIX Text Processing (Hayden Books, 1987) by the author of this handbook and Tim O'Reilly, covers sed and awk in full, although we did not include the new version of awk. Readers of that book will find some parts that are the same as in this book, but in general a different approach has been taken here. Whereas in the textbook we treated sed and awk separately, expecting advanced users only to tackle awk, here we try to present both programs in relation to one another. They are different tools that can be used individually or together to provide interesting opportunities for text processing.

Sample Programs

The sample programs in this book were written and tested on a Mac IIci running A/UX 2.0 (UNIX System V Release 2) and a SparcStation 1 running SunOS 4.0. Programs requiring nawk were generally tested using gawk (2.10beta 07 Apr 1989) as well as nawk.

The full-length example programs in this book are available free of charge from UUNET (that is, except for UUNET's connect-time charges). If you have access to UUNET, you can retrieve the source code using **uucp** or **ftp**.

For uucp transfer, the filename is *~/published/oreilly/nutshell/sedawk/progs.tar .Z*. For ftp transfer, **cd** to */published/oreilly/nutshell/sedawk* , specify binary transfer and get */progs.tar.Z*. You will need to uncompress the files and extract them from the archive.

Obtaining the Example Programs

The example programs in this book are available electronically in a number of ways, including ftp and uucp. Use *ftp* if you are directly on the Internet. Use UUCP if you have a modem.

FTP

To use FTP, you need a machine with direct access to the Internet. A sample session is shown, with what you should type in boldface.

```
% ftp ftp.uu.net
Connected to ftp.uu.net.
220 FTP server (Version 6.21 Tue Mar 10 22:09:55 EST 1992) ready.
Name (ftp.uu.net:kismet): anonymous
331 Guest login ok, send domain style e-mail address as password.
Password: kismet@ora.com (use your user name and host here)
230 Guest login ok, access restrictions apply.
ftp> cd /published/oreilly/nutshell/sedawk
250 CWD command successful.
ftp> binary (Very important! You must specify binary transfer for compressed files.)
200 Type set to I.
ftp> get progs.tar.Z
200 PORT command successful.
150 Opening BINARY mode data connection for xlibprgs2.tar.Z.
226 Transfer complete.
ftp> quit
221 Goodbye.
%
```

If the file is a compressed tar archive, extract the files from the archive by typing:

```
% zcat progs.tar.Z | tar xf -
```

System V systems require that you add the *o* option to the tar command.

If *zcat* is not available on your system, use separate uncompress and tar commands.

UUCP

You can get the examples from UUNET whether you have an account or not. If you or your company has an account with UUNET, you will have a system with a direct UUCP connection to UUNET. The basic syntax for uucp is:

```
uucp source yourhost!~/yourname/
```

The *source* is a pathname to the file on the remote machine. In this case, it is:

```
uunet\!~/published/oreilly/nutshell/sedawk/progs.tar.Z/
```

The backslashes can be omitted if you use the Bourne shell (*sh*) instead of *csh*. The file should appear some time later (up to a day or more) in the directory

*/usr/spool/uucppublic/**yourname***. If you don't have an account but would like one so that you can get electronic mail, then contact UUNET at 703-204-8000.

If you don't have a UUNET account, you can set up a UUCP connection to UUNET using the phone number 1-900-468-7727. As of this writing, the cost is 50 cents per minute. The charges will appear on your next telephone bill. The login name is "uucp" with no password. For example, an *L.sys/Systems* entry might look like:

```
uunet Any ACU 19200 1-900-468-7727 ogin:--ogin: uucp
```

Your entry may vary depending on your UUCP configuration. If you have a PEP-capable modem, make sure s50=255s111=30 is set before calling.

Once you've got the desired file, follow the directions under FTP to extract the files from the archive.

Conventions Used in this Handbook

The following conventions are used in this book:

Bold	is used for statements and functions, identifiers, and program names.
Italic	is used for file and directory names when they appear in the body of a paragraph as well as for data types and to emphasize new terms and concepts when they are introduced.
`Constant Width`	is used in examples to show the contents of files or the output from commands.
`Constant Bold`	is used in examples to show command lines and options that should be typed literally by the user.
Quotes	are used to identify a code fragment in explanatory text. System messages, signs, and symbols are quoted as well.
$	is the Bourne Shell prompt.

%	is the C Shell prompt.
[]	surrounds optional elements in a description of program syntax. (The brackets themselves should never be typed.)

Acknowledgements

To say that this book has been long anticipated is no understatement. I published three articles on awk in UNIX/World in the spring and summer of 1987, making the mistake of saying that these articles were from the upcoming Nutshell Handbook, "Sed & Awk." I proposed to Tim O'Reilly that I adapt the articles and create a book as a project I could work on at home shortly after the birth of my son, Benjamin. I thought I'd finish it in several months. Well, my son turned three recently, around the time I was completing the first draft. Cathy Brennan and the customer service representatives have been patiently handling requests for the book ever since the UNIX/World articles appeared. Cathy said that she even had people call to order the book, swearing it was available because they knew other people who had read it. I owe a debt of gratitude to her and her staff and to the readers I've kept waiting.

My thanks to Tim O'Reilly for creating a great company in which one can easily get sidetracked by a number of interesting projects. As editor, he pushed me to complete the book but would not allow it to be complete without his writing all over it. As usual, his suggestions made me work to improve the book.

Thanks to all the writers and production editors at O'Reilly & Associates, who presented interesting problems to be solved with sed and awk. Thanks to Ellie Cutler who was the production editor for the book and also wrote the index. Thanks to Lenny Muellner for allowing me to quote him throughout the book. Thanks as well to Sue Willing and Donna Woonteiler for their efforts in getting the book into print. Thanks to Chris Reilly who did the illustrations. Thanks to the individual contributors of the sed and awk scripts in Chapter 12. Thanks also to Kevin C. Castner, Tim Irvin, Mark Schalz, Alex Humez, Glenn Saito, Geoff Hagel, Tony Hurson, Jerry Peek, Mike Tiller, and Lenny Muellner, who sent me mail pointing out typos and errors.

Finally, dearest thanks to Nancy and Katie, Ben and Glenda.

Table of Contents