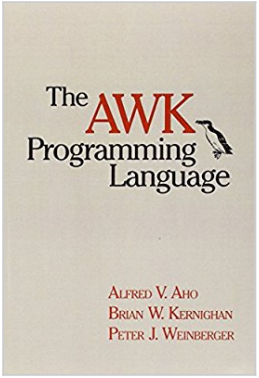

The AWK Programming Language

By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger

Originally developed by Alfred Aho, Brian Kernighan, and Peter Weinberger in 1977, AWK is a pattern-matching language for writing short programs to perform common data-manipulation tasks. In 1985, a new version of the language was developed, incorporating additional features such as multiple input files, dynamic regular expressions, and user-defined funcitons. This new version is available for both UNIX and MS-DOS.

The AWK Programming Language

By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger

Originally developed by Alfred Aho, Brian Kernighan, and Peter Weinberger in 1977, AWK is a pattern-matching language for writing short programs to perform common data-manipulation tasks. In 1985, a new version of the language was developed, incorporating additional features such as multiple input files, dynamic regular expressions, and user-defined funcitons. This new version is available for both UNIX and MS-DOS.

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger Bibliography

- Sales Rank: #253683 in Books
- Published on: 1988-01-11
- Original language: English
- Number of items: 1
- Dimensions: 9.00" h x .80" w x 6.20" l, .65 pounds
- Binding: Paperback
- 210 pages

Editorial Review

From the Back Cover

Originally developed by **Alfred Aho, Brian Kernighan,** and **Peter Weinberger** in 1977, AWK is a pattern-matching language for writing short programs to perform common data-manipulation tasks. In 1985, a new version of the language was developed, incorporating additional features such as multiple input files, dynamic regular expressions, and user-defined functions. This new version is available for both **Unix** and **MS-DOS.**

This is the first book on AWK. It begins with a tutorial that shows how easy AWK is to use. The tutorial is followed by a comprehensive manual for the new version of AWK. Subsequent chapters illustrate the language by a range of useful applications, such as:

- Retrieving, transforming, reducing, and validating data
- Managing small, personal databases
- Text processing
- Little languages
- Experimenting with algorithms

The examples illustrates the book's three themes: showing how to use AWK well, demonstrating AWK's versatility, and explaining how common computing operations are done. In addition, the book contains two appendixes: summary of the language, and answers to selected exercises.

020107981XB04062001

About the Author

Brian W. Kernighan works in the Computing Science Research Center at Bell Laboratories, Lucent Technologies. He is Consulting Editor for Addison-Wesley's Professional Computing Series and the author, with Dennis Ritchie, of *The C Programming Language*.

020107981XAB04062001

> Computer users spend a lot of time doing simple mechanical data manipulation -- changing the format of data, checking its validity, finding items with some property, adding up numbers, printing reports, and the like. All of these jobs ought to be mechanized, but it's a real nuisance to have to write a special-purpose program in a standard language like C or Pascal each time such a task comes up.
>
> Awk is a programming language that makes it possible to handle such tasks with very short programs, often only one or two lines long. An awk program is a sequence of patterns and

actions that tell what to look for in the input data and what to do when it's found. Awk searches a set of files for lines matched by any of the patterns; when a matching line is found, the corresponding action is performed. A pattern can select lines by combinations of regular expressions and comparison operations on strings, numbers, fields, variables, and array elements. Actions may perform arbutrary processing on selected lines,; the action language looks like C but there are no declarations, and strings and numbers are built-in data types.

Awk scans the input files and splits each input line into fields automatically. Because so many things are automatic--input, field splitting, storage management, initialization--awk programs are usually much smaller than they would be in a more conventional language. Thus one common use of awk is for the kind of data manipulation suggested above. Programs, a line or two long, are composed at the keyboard, run once, then discarded. In effect, awk is a general-purpose programable tool that can replace a host of specialized tools or programs.

The same brevity of expression and convenience of operations make awk valuable for prototyping larger programs. One starts with a few lines, then refines the program unitil it doesth edesired job, experimenting with designs by trying alternatives quickly. Since programs are short, it's easy to get started, and easy to start over when experience suggests a different direction. And, it's straightforward to translate an awk program into another language once the design is right.

Organization of the Book

The first goal of this book is to teach you what awk is and how to use it effectively. Chapter 1 is a tutorial on the bare minimum necessary to get started; after reading even a few pages you should have enough information to begin writing useful programs. The examples in this chapter are very short and simple, typical othe interactive use of awk.

Chapter 2 covers the entire language in a systematic order. Although there are plenty of examples in this chapter, like most manuals it's long and a bit dry, so you will probably want to skim it on a first reading.

The rest of the book contains a wide variety of examples, chosen to show the breadth of applicability of awk and how to make good use of its facilities. Some of the programs are in regular use in our environment; others show ideas but are not intended for production use; a few are included just because they are fun.

The emphasis in Chapter 3 is on retrieval, transformation, reduction and validation of data--the tasks that awk was originally designed for. There is also a discussion of how to handle data like address lists that naturally comes in multiline chunks.

Awk is a good language for managing small, personal databases. Chapter 4 discusses the generation of reports from databases, and builds a simple relational database system and query language for data stored in multiple files.

Awk handles text with much the same convenience that most languages handle numbers, so it often finds application in text processing. Chapter 5 describes programs for generating text, and some that help with document preparation. One of the examples is an indexing program based on the one we used for this book.

Chapter 6 is about "little languages," that is, specialized languages that focus on a narrow domain. Awk is convenient for writing small translators because it's basic operations support many of the lexical and table- management tasks encountered in translation. The chapter includes an assembler, a graphics language, and several calculators.

Awk is a good language for expressing certain kinds of algorithms. Because there are no declarations and because storage management is easy, an awk program has many of the advantages of pseudo-code but awk programs can be run, which is not true of pseudo-code. The focus in Chapter 7 is on experimentation with algorithms, including testing and performance evaluation. It shows several sorting algorithms, and culminates in a version of the Unix make program.

Chapter 8 describes some of the historical reasons why awk is as it is and offers some suggestions on what to do when it is too slow or too confining.

Appendix A is a summary of the language; Appendix B contains answers to selected exercises.

You shoul begin by reading Chapter 1, and trying some small examples of your own. Go through Chapter 2 quickly, concentrating on the summaries and tables; don't get bogged down in the details. Then read as far into each of the subsequent chapters as your interest takes you. These chapters are nearly independent of each other, so the order doesn't matter much.

The Examples

There are several themes in the examples. The primary one, of course, is to show how to use awk well. We have tried to include a wide variety of useful constructions, and we have stressed particular aspects like associative arrays and regular expressions that typify awk programming.

A second theme is to show awk's versatility. Awk programs have been used from databades to circuit design, from numerical analysis to graphics, from compilers to system adminstration, from a first language for nonprogrammers to the implementation language for software engineering courses. We hope that the diversity of applications illustrated in the book will suggest new possibilities to you as well.

A third theme is to show how common computing operations are done. The book contains a relational database system, an assembler and interpreter for a toy computer, a graph-drawing language, a recursive-descent parser for an awk subset, a file-update program based on make and many other examples. In each case, a short awk program conveys the essence of how something works in a form that you can understand and play with.

We have also tried to illustrate a spectrum of ways to attack programming problems. Rapid prototyping is an approach that awk supports well. A less obvious strategy is divide and conquer; breaking a big job into small components, each concentrating on one aspect of the problem. Another is writing programs that create other programs. Little languages define a good user interface and often suggest a sound implementation. Although these ideas are presented here in the context of awk, they are much more generally applicable and ought to part of every programmer's repertoire.

The examples have all been tested directly from the text, which is in machine-readable form.

We have tried to make the programs error-free, but we have not added features nor made them proof against all possible invalid inputs, preferring to concentrate on conveying the essentials.

Evolution of the AWK language

Awk was originally designed and implemented by the authors in 1977, in part as an experiment to see how the Unix gools grep and sed could be generalized to deal with numbers as well as text. It was based on our interests in regular expressions and programmable editors. Although it was meant for writing very short programs, its combination of facilities soon attracted users who wrote significantly larger programs. These larger programs needed features that had not been part of the original implementation, so awk was enhanced in a new version made available in 1985.

Other enhancements include dynamic regular expressions, with text substitution and pattern-matching functions; additional built-in functions and variables; some new operators and statements; iput from multiple files; and access to command-line arguments. Error messages have also been improved. The examples in Chapter 1 use only facilities of the original version; many examples in later chapters take advantage of new features.

The version of awk is partof Unix System V Release 3.1. Source code for this version is also available through AT&T's Unix System Toolchest software distribution system; call 1-201-522-6900 and log in as guest. In Europe, contact AT&T Unix Europe in London (44-1-567-7711); in the Far EASt, contact AT&T Unix Pacific in Tokyo (81-3-431-3670).

Since awk was developed under Unix, some of its features reflect capabilities usually found only there; these features are used in some of our examples. Furthermore, we have assumed the existence of some Unix utilities, particularly sort, for which exact equivalents may not exist elsewhere. Aside from these limitations, however, awk should be useful in any environment; in particular, it runs on MS-DOS. Further information is available from Addison-Wesley.

Awk is certainly not perfect; it has its share of irregularities, omissions, and just plain bad ideas, and it's sometimes painfully slow. But it's also a rich and versatile language, useful in a remarkable number of cases. We hope you'll find it as valuable as we do.

Acknowledgments

We are deeply indebted to friends who made comments and suggestions on drafts of this book. We are particularly grateful to Jon Bentley, whose enthusiasm has been an inspiration for years. Jon contributed many ideas and programs derived from his experience using and teaching awk; he also read several drafts with great care. Doug McIlroy also deserves special recognition; his peerless talent as a reader greatly improved the structure and content of the whole book. Others who made helpful comments on the manuscript include Susan Aho, Jaap Akkerhuis, Lorinda Cherry, Chris Fraser, Eric Grosse, Ricardo Gusella, Bob Herbst, Mark Kernighan, John Linderman, Bob Martin, Howard Moscovitz, Gerard Schmitt, Don Swartwout, Howard Trickey, Peter van Eijk, Chris Van Wyk, and Mihalis Yannakakis. We thank them all.

Alfred V. Aho

Brian W. Kernighan
Peter J. Weinberger

020107981XP04062001

Users Review

From reader reviews:

Gracie Thomas:

Do you have favorite book? Should you have, what is your favorite's book? E-book is very important thing for us to know everything in the world. Each e-book has different aim or even goal; it means that guide has different type. Some people truly feel enjoy to spend their time and energy to read a book. They may be reading whatever they take because their hobby is reading a book. How about the person who don't like reading through a book? Sometime, man or woman feel need book whenever they found difficult problem or exercise. Well, probably you should have this The AWK Programming Language.

Inez Tuller:

Reading a reserve can be one of a lot of task that everyone in the world really likes. Do you like reading book thus. There are a lot of reasons why people enjoy it. First reading a e-book will give you a lot of new information. When you read a guide you will get new information because book is one of many ways to share the information or perhaps their idea. Second, reading through a book will make you more imaginative. When you reading a book especially tale fantasy book the author will bring someone to imagine the story how the characters do it anything. Third, you are able to share your knowledge to other individuals. When you read this The AWK Programming Language, it is possible to tells your family, friends and also soon about yours book. Your knowledge can inspire the others, make them reading a book.

Angela Thomas:

Many people spending their time by playing outside with friends, fun activity with family or just watching TV all day long. You can have new activity to invest your whole day by reading through a book. Ugh, think reading a book can really hard because you have to take the book everywhere? It fine you can have the e-book, delivering everywhere you want in your Touch screen phone. Like The AWK Programming Language which is having the e-book version. So , try out this book? Let's find.

Donald Chen:

Within this era which is the greater man or woman or who has ability to do something more are

more special than other. Do you want to become among it? It is just simple strategy to have that. What you should do is just spending your time not very much but quite enough to experience a look at some books. One of several books in the top checklist in your reading list is The AWK Programming Language. This book which is qualified as The Hungry Hills can get you closer in turning out to be precious person. By looking right up and review this publication you can get many advantages.

Read The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger for online ebook

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger books to read online.

Online The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger ebook PDF download

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger Doc

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger Mobipocket

The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger EPub

IWTSL5RJUH6: The AWK Programming Language By Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger