A practical guide to learning GNU Awk

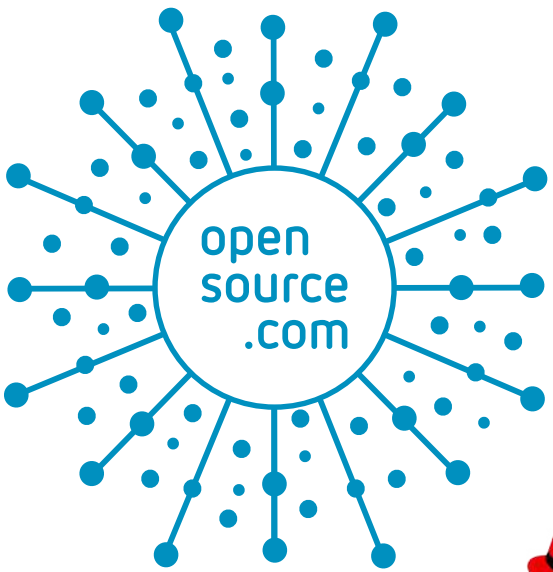
ABOUT OPENSOURCE.COM

What is Opensource.com?

OPENSOURCE.COM publishes stories about creating, adopting, and sharing open source solutions. Visit Opensource.com to learn more about how the open source way is improving technologies, education, business, government, health, law, entertainment, humanitarian efforts, and more.

Submit a story idea: opensource.com/story

Email us: open@opensource.com



Supported by
Red Hat

AUTHORS SETH KENLON, DAVE MORRISS, AND ROBERT YOUNG

SETH KENLON is an independent multimedia artist, free culture advocate, and UNIX geek. He has worked in the film and computing industry, often at the same time. He is one of the maintainers of the Slackware-based multimedia production project, http://slackermedia.info.

DAVE MORRISS is a retired IT Manager now contributing to the "Hacker Public Radio" community podcast (http://hackerpublicradio.org) as a podcast host and an administrator.

ROBERT YOUNG is the Owner and Principal Consultant at Lab Insights, LLC. He has led dozens of laboratory informatics and data manage projects over the last 10 years. Robert Holds a degree in Cell Biology/Biochemistry and a masters in Bioinformatics.

CONTRIBUTORS

Jim Hall

Lazarus Lazaridis

Dave Neary

Moshe Zadka

CONTENTS

CHAPTERS

LEARN

PRACTICE

CHEAT SHEET

What is awk?

awk is known for its robust ability to process and interpret data from text files.

AWK is a programming language and a POSIX [1] specification that originated at AT&T Bell Laboratories in 1977. Its name comes from the initials of its designers: Aho, Weinberger, and Kernighan. awk features user-defined functions, multiple input streams, TCP/IP networking access, and a rich set of regular expressions. It's often used to process raw text files, interpreting the data it finds as records and fields to be manipulated by the user.

At its most basic, awk searches files for some unit of text (usually lines terminated with an end-of-line character) containing some user-specified pattern. When a line matches one of the patterns, awk performs some set of user-defined actions on that line, then processes input lines until the end of the input files.

awk is used as a command as often as it is used as an interpreted script. One-liners are popular and useful ways of filtering output from files or output streams or as standalone commands. awk even has an interactive mode of sorts because, without input, it acts upon any line the user types into the terminal:

```
$ awk '/foo/ { print toupper($0); }'
This line contains bar.
This line contains foo.
THIS LINE CONTAINS FOO.
```

However, awk is a programming language with user-defined functions, loops, conditionals, flow control, and more. It's robust enough as a language that it has been used to program a wiki and even (believe it or not) a retargetable assembler for eight-bit microprocessors.

Why use awk?

awk may seem outdated in a world fortunate enough to have Python available by default on several major operating systems, but its longevity is well-earned. In many ways, programs written in awk are different from programs in other languages because awk is data-driven. That is, you describe to awk what data you want to work with and then what you want it to do when such data is found. There are no boilerplate constructors to create, no elaborate class structure to design, no stream objects to create. awk is built for a specific purpose, so there's a lot you can take for granted and allow awk to handle.

What's the difference between awk and gawk?

Awk is an open source POSIX specification, so anyone can (in theory) implement a version of the command and language. On Linux or any system that provides GNU awk [2], the command to invoke awk is **gawk**, but it's symlinked to the generic command **awk**. The same is true for systems that provide **nawk** or **mawk** or any other variety of awk implementation. Most versions of awk implement the core functionality and literal functions defined by the POSIX spec, although they may add special new features not present in others. For that reason, there's some risk of learning one implementation and coming to rely on a special feature, but this "problem" is tempered by the fact that most of them are open source, so they usually can be installed as needed.

Learning awk

There are many great resources for learning awk. The GNU awk manual, *GAWK: Effective awk programming* [3], is a definitive guide to the language. You can find many other tutorials for awk [4] on Opensource.com, including "Getting started with awk, a powerful text-parsing tool." [5]

Links

[1] https://opensource.com/article/19/7/what-posix-richard-stallman-explains
[2] https://www.gnu.org/software/gawk/
[3] https://www.gnu.org/software/gawk/manual/
[4] https://opensource.com/sitewide-search?search_api_views_fulltext=awk
[5] https://opensource.com/article/19/10/intro-awk