



AWK

interpreted programming language

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

This tutorial takes you through AWK, one of the most prominent text-processing utility on GNU/Linux. It is very powerful and uses simple programming language. It can solve complex text processing tasks with a few lines of code.

Starting with an overview of AWK, its environment, and workflow, the tutorial proceeds to explain the syntax, variables, operators, arrays, loops, and functions used in AWK. It also covers topics such as output redirection and pretty printing.

Audience

This tutorial will be useful for software developers, system administrators, or any enthusiastic reader inclined to learn how to do text processing and data extraction in Unix-like environment.

Prerequisites

You must have a basic understanding of GNU/Linux operating system and shell scripting.

Copyright & Disclaimer

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents.....	ii
1. OVERVIEW	1
Types of AWK.....	1
Typical Uses of AWK	1
2. AWK ENVIRONMENT.....	2
Installation Using Package Manager	2
Installation from Source Code.....	2
3. AWK WORKFLOW	4
Program Structure	5
Example	5
4. BASIC SYNTAX	7
AWK Command Line	7
AWK Program File.....	7
AWK Standard Options	8
5. BASIC EXAMPLES.....	13
Printing Column or Field	13
Printing All Lines	13
Printing Columns by Pattern	14
Printing Column in Any Order	14
Counting and Printing Matched Pattern.....	14
Printing Lines with More than 18 Characters	15

6. BUILT-IN VARIABLES.....	16
Standard AWK Variables	16
GNU AWK Specific Variables	20
7. OPERATORS	23
Arithmetic Operators	23
Increment and Decrement Operators	24
Assignment Operators	25
Relational Operators.....	27
Logical Operators.....	29
Ternary Operator	30
Unary Operators	31
Exponential Operators	31
String Concatenation Operator	32
Array Membership Operator.....	32
Regular Expression Operators	32
8. REGULAR EXPRESSIONS	34
Dot.....	34
Start of Line	34
End of Line	34
Match Character Set	35
Exclusive Set	35
Alteration	35
Zero or One Occurrence	36
Zero or More Occurrence	36
One or More Occurrence	36
Grouping.....	37

9. ARRAYS	38
Creating Array	38
Deleting Array Elements	38
Multi-Dimensional Arrays	39
10. CONTROL FLOW	41
If Statement	41
If-Else Statement	41
If-Else-If Ladder	42
11. LOOPS	43
For Loop	43
While Loop	43
Do-While Loop	44
Break Statement	44
Continue Statement	45
Exit Statement	46
12. BUILT-IN FUNCTIONS	47
Arithmetic Functions	47
String Functions	51
Time Functions	57
Bit Manipulation Functions	60
Miscellaneous Functions	62
13. USER-DEFINED FUNCTIONS	68
14. OUTPUT REDIRECTION	70
Redirection Operator	70
Append Operator	70
Pipe	71

Two-Way Communication.....71

15. PRETTY PRINTING.....73

 Escape Sequences73

 Format Specifier.....75

 Optional Parameters with %78

1. OVERVIEW

AWK is an interpreted programming language. It is very powerful and specially designed for text processing. Its name is derived from the family names of its authors – **Alfred Aho, Peter Weinberger, and Brian Kernighan.**

The version of AWK that GNU/Linux distributes is written and maintained by the Free Software Foundation (FSF); it is often referred to as **GNU AWK.**

Types of AWK

Following are the variants of AWK:

AWK - Original AWK from AT & T Laboratory.

NAWK - Newer and improved version of AWK from AT & T Laboratory.

GAWK - It is GNU AWK. All GNU/Linux distributions ship GAWK. It is fully compatible with AWK and NAWK.

Typical Uses of AWK

Myriad of tasks can be done with AWK. Listed below are just a few of them:

- Text processing
- Producing formatted text reports
- Performing arithmetic operations
- Performing string operations, and many more

2. AWK ENVIRONMENT

This chapter describes how to set up the AWK environment on your GNU/Linux system.

Installation Using Package Manager

Generally, AWK is available by default on most GNU/Linux distributions. You can use **which** command to check whether it is present on your system or not. In case you don't have AWK, then install it on Debian based GNU/Linux using Advance Package Tool (**APT**) package manager as follows:

```
[jerry]$ sudo apt-get update
[jerry]$ sudo apt-get install gawk
```

Similarly, to install AWK on RPM based GNU/Linux, use Yellowdog Updator Modifier (**YUM**) package manager as follows:

```
[root]# yum install gawk
```

After installation, ensure that AWK is accessible via command line.

```
[jerry]$ which awk
```

On executing the above code, you get the following result:

```
/usr/bin/awk
```

Installation from Source Code

As GNU AWK is a part of the GNU project, its source code is available for free download. We have already seen how to install AWK using package manager. Let us now understand how to install AWK from its source code.

The following installation is applicable to any GNU/Linux software, and for most other freely-available programs as well. Here are the installation steps:

Step 1 - Download the source code from an authentic place. The command-line utility **wget** serves this purpose.

```
[jerry]$ wget http://ftp.gnu.org/gnu/gawk/gawk-4.1.1.tar.xz
```


Step 2 - Decompress and extract the downloaded source code.

```
[jerry]$ tar xvf gawk-4.1.1.tar.xz
```

Step 3 - Change into the directory and run configure.

```
[jerry]$ ./configure
```

Step 4 - Upon successful completion, the **configure** generates Makefile. To compile the source code, execute a **make** command.

```
[jerry]$ make
```

Step 5 - You can run the test suite to ensure the build is clean. This is an optional step.

```
[jerry]$ make check
```

Step 6 - Finally, install AWK. Make sure you have super-user privileges.

```
[jerry]$ sudo make install
```

That is it! You have successfully compiled and installed AWK. Verify it by executing the **awk** command as follows:

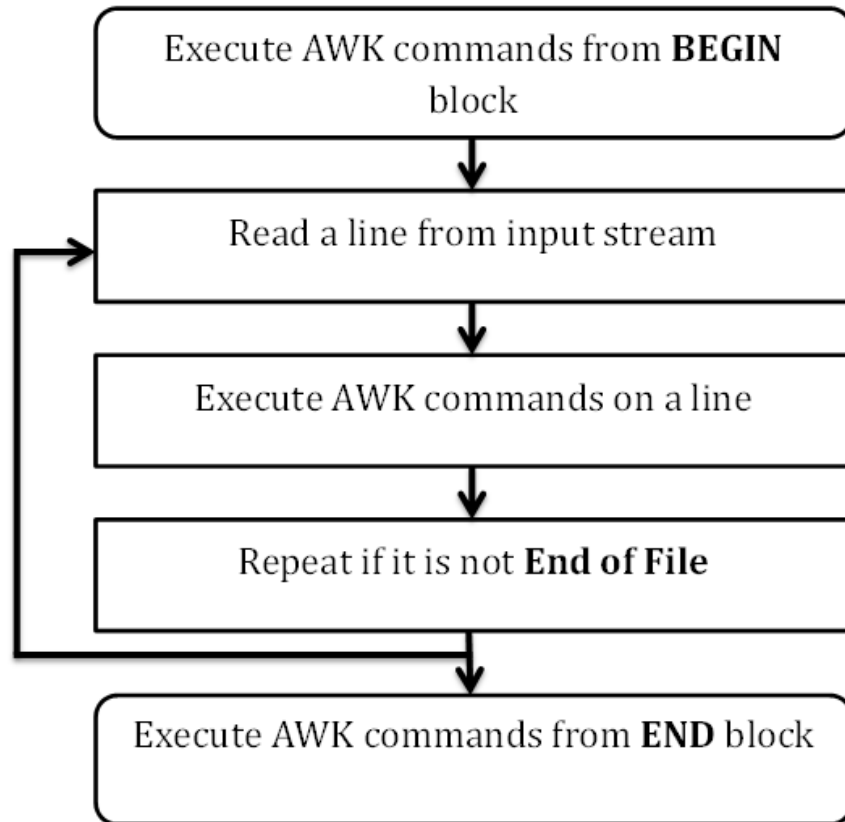
```
[jerry]$ which awk
```

On executing this code, you get the following result:

```
/usr/bin/awk
```

3. AWK WORKFLOW

To become an expert AWK programmer, you need to know its internals. AWK follows a simple workflow: Read, Execute, and Repeat. The following diagram depicts the workflow of AWK:



Read

AWK reads a line from the input stream (file, pipe, or stdin) and stores it in memory.

Execute

All AWK commands are applied sequentially on the input. By default, AWK executes commands on every line. We can restrict this by providing patterns.

Repeat

This process repeats until the file reaches its end.

End of ebook preview
If you liked what you saw...
Buy it from our store @ **<https://store.tutorialspoint.com>**