The Awk Family

Current Date: 2022-03-23

David Belmont
ACME CORP

Awk Overview 👀

Awk is a UNIX programming language used for manipulating data and generating reports . Nawk is a newer version of awk , and gawk is the GNU version used on Linux.

The data may come from standard input, one or more files, or as output from a process. Awk can be used at the command line for simple operations, or it can be written into programs for larger applications. Because awk can manipulate data, it is an indispensable tool used in shell scripts and for managing small databases.

Awk scans a file (or input) line by line, from the first to the last line, searching for lines that match a specified pattern and performing selected actions (enclosed in curly braces) on those lines. If there is a pattern with no specific action, all lines that match the pattern are displayed; if there is an action with no pattern, all input lines specified by the action are executed upon.

Objectives 🎯

This tutorial will teach you how to write awk commands that will be used on your Linux server. Before getting started, make sure you know the filename and where it is in the Linux directory.

> ⓘ cd /directory/file

Table of Contents ⚓

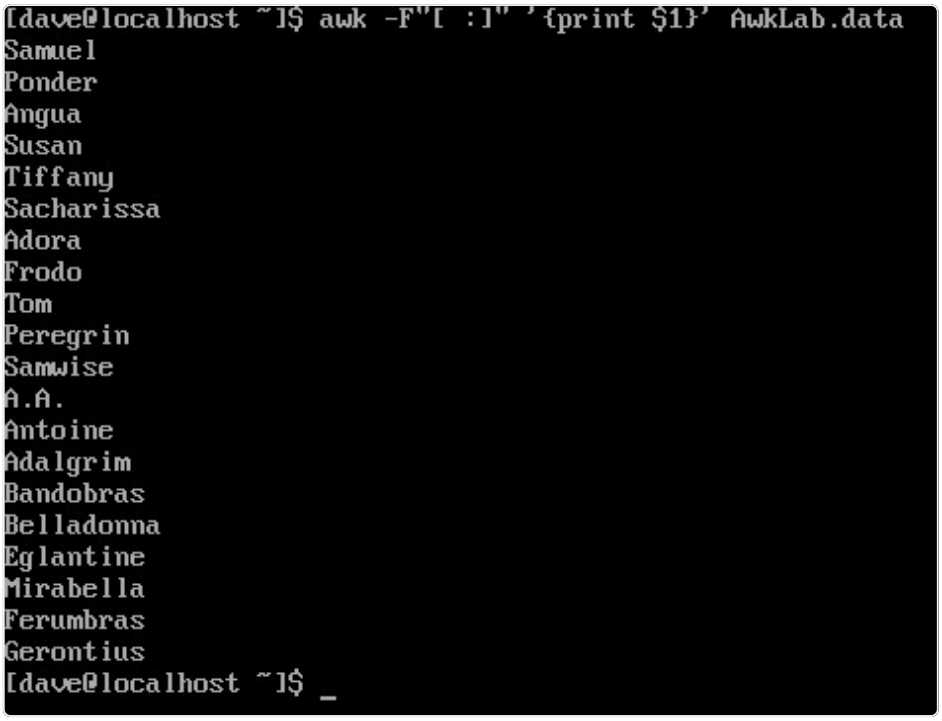
TABLE OF CONTENTS

1: Print all the First Names.✔️

> ⓘ awk -F"[:]" '{print $1}' AwkLab.data

- awk is the command.
- -F file separator.

- : separates file
- print $1 string before :
- AwkLab.data is the file we are using.



Example of command line code

2: Print phone numbers for Tom and Frodo after their names.✔

> awk -F '[:]' '/^Tom/{print $3$4 }' AwkLab.data && awk -F '[:]' '/^Frodo/{print $3$4 }' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- ^ Starting with name
- print $3$4 prints column 3 and 4 since separated
- && adds another command for Frodo
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F"[: ]" '/^Tom/{print $3$4 }' AwkLab.data && awk -F"[: ]" '/^Frodo/{print $
3$4 }' AwkLab.data
(916)348-4278
(206)548-1278
[dave@localhost ~]$ _
```

Example of command line code

3: Print Peregrin's full name and phone number area code only.✔

awk -F '[:]' '/^Peregrin/{print $1$2$3 }' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- ^ Starting with name
- print $1$2$3 prints column 1 2 3
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F"[: ]" '/^Peregrin/{print $1$2$3 }' AwkLab.data
PeregrinTook(510)
[dave@localhost ~]$ _
```

Example of command line code

4: Print all phone numbers in the 123 area code along with the names✔

cat AwkLab.data

- First we need to see how the names and numbers are formatted.
- We now have to adjust our code so we don't miss Mr. Saint-Exupery.

awk -F "[:]" '$3=="(123)"{print $1$2$3}' AwkLab.data && awk -F "[:]" '$4=="(123)" {print $1$2$3$4}' AwkLab.data

- awk is the command.

- -F file separator.
- : separates file
- $3 3rd column equals 123
- print 1 2 3 columns
- && another line of code to include the name with the hyphen
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[: ]" '$3=="(123)"{print $1$2$3}' AwkLab.data && awk -F "[: ]" '$4=="(12
3)"{print $1$2$3$4}' AwkLab.data
BelladonnaTook(123)
EglantineTook(123)
AntoinedeSaint-Exupery(123)
[dave@localhost ~]$
```

Example of command line code

5: Print all Last names beginning with either a T or D.✔

cat AwkLab.data

- First we need to see how the names and numbers are formatted.
- We now have to adjust our code so we don't miss a person with a middle name.

awk -F "[:]" '$2 ~/^T/{print $2}' AwkLab.data && awk -F "[:]" '$3 ~/^T/{print $3}' AwkLab.data && awk -F "[:]" '$3 ~/^D/{print $3}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- $2 2nd column starts with T
- print second column (last name)
- && another line of code same format to include names with middle names
- && another line of code same format to include last names that start with D with middle names
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[: ]" '$2 ~/^T/{print $2}' AwkLab.data && awk -F "[: ]" '$3 ~/^T/{print
$3}' AwkLab.data && awk -F "[: ]" '$3 ~/^D/{print $3}' AwkLab.data
Took
Took
Took
Took
Took
Took
Took
Took
Dearheart
[dave@localhost ~]$
```

Example of command line code

6: Print all first names containing four or less characters.✔

awk 'length($1) <5 {print $1}' AwkLab.data

- awk is the command.
- length($1) length of first column
- Any name less than 5 in first column
- {print $1} prints first column if above lines match
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk 'length($1) <5 {print $1}' AwkLab.data
Tom
A.A.
[dave@localhost ~]$ _
```

Example of command line code

7: Print the first names and area codes of all those in the 916 area code. ✔

awk -F "[:]" '$3 ~/(916)/ {print $1$3}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- $3 on column 3 match 916
- print $1$3 prints first name and area code
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[: ]" '$3 ~/(916)/ {print $1$3}' AwkLab.data
Sacharissa(916)
Tom(916)
A.A.(916)
[dave@localhost ~]$
```

Example of command line code

8: Print Sacharissa's campaign contributions following her name. Each value should be printed with a leading dollar sign; e.g., $250 $100 $175. ✔

> awk -F "[:]" '1$ ~/Sacharissa/ {print "
> quot;$5, "
> quot;$6, "
> quot;$7}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- $1 ~/Sacharissa/ first column contains Sacharissa
- print "
  quot; will print out a dollar sign before number
- $5 $6 $7 are columns that contain number amount
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[: ]" '$1 ~/Sacharissa/ {print "$"$5, "$"$6, "$"$7}' AwkLab.data
$250 $100 $175
[dave@localhost ~]$ _
```

Example of command line code

9: Print last names followed by a comma and the phone number. Be careful of the last name's format.✔

> awk -F "[:]" '{print $2$3 ", " $4}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file

- print $3$4 prints columns with last name
- ", " adds comma
- $4 contains phone number
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[ :]" '{print $2$3", " $4}' AwkLab.data
Vimes(510), 548-1278
Stibbons(408), 538-2358
vonÜberwald, (206)
StoHelit, (206)
Aching(206), 548-1278
Cripslock(916), 343-6410
BelleDearheart, (406)
Baggins(206), 548-1278
Bombadil(916), 348-4278
Took(510), 548-5258
Gamgee(408), 926-3456
Milne(916), 440-1763
deSaint-Exupery, (123)
Took(345), 978-7684
"Bullroarer"Took, (453)
Took(123), 978-5754
Took(123), 978-3574
Took(345), 978-2677
IIITook, (563)
Took(574), 978-8535
[dave@localhost ~]$ _
```

Example of command line code

10: Print the first and last names of those who contributed more than $110 in the last month. Make sure to include their last month contribution amount after the name✔

> ⓘ awk -F "[:]" '$7 > 110 {print $1$2$7}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- If column 7 (last month contribution) is more than 110 print
- $1$2$7 prints names and amount if statement correct
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[ :]" '$7 > 110 {print $1$2$7}' AwkLab.data
SamuelVimes175
PonderStibbons201
TiffanyAching150
SacharissaCripslock175
AdoraBelle300
TomBombadil175
PeregrinTook135
SamwiseGamgee200
A.A.Milne300
AdalgrimTook467
Bandobras"Bullroarer"368
BelladonnaTook175
EglantineTook4367
MirabellaTook175
GerontiusTook4562
[dave@localhost ~]$ _
```

Example of command line code

11: Print the last names, phone numbers, and first month contribution of those who contributed less than $150 in the first month.✔

> awk -F "[:]" '$5 > 150 {print $2$3$4}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- If column 5 (first month contribution) is less than 150 print
- $2$3$4 prints last names and phone number if statement correct
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[ :]" '$5 < 150 {print $2$3$4}' AwkLab.data
Aching(206)548-1278
Took(510)548-5258
[dave@localhost ~]$ _
```

Example of command line code

12: Print the first names and contribution of those who contributed between $75 and $150 in the first month.✔

awk -F "[:]" '$5 < 150 && $5 < 75 {print $1}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- If column 5 (first month contribution) is less than 150 and more than 75 print
- $1 prints last names if statement correct
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[ :]" '$5 < 150 && $5 < 75 {print $1}' AwkLab.data
Tiffany
Peregrin
[dave@localhost ~]$ _
```

Example of command line code

13: Print the first and last names and total contributions of those who contributed less than $700 over the three-month period.✔

awk -F "[:]" '{sum=$5+$6+$7} sum < 700 {print $1$2}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- sum= new variable with those columns added together
- if sum is less than 700 it will print names
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[ :]" '{sum=$5+$6+$7} sum < 700 {print $1$2}' AwkLab.data
SamuelVimes
PonderStibbons
TiffanyAching
SacharissaCripslock
FrodoBaggins
TomBombadil
PeregrinTook
SamwiseGamgee
A.A.Milne
[dave@localhost ~]$ _
```

Example of command line code

14: Print the first names and first letter of the last name, and average contribution of those who had an average contribution of more then $300✔

> awk -F "[:]" '{average=$5$6$7/3} average > 300 {print $1 substr($2, 1, 1)}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- average= new variable with those columns added together divided by 3
- if average is more than 300 it will print names
- $1 is first name column
- substr($1, 1, 1) will print first letter of last names
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F "[ :]" '{average=$5+$6+$7/3} average > 300 {print $1 substr($2, 1, 1)}' A
wkLab.data
SamuelV
PonderS
Anguav
SusanS
SacharissaC
AdoraB
FrodoB
TomB
SamwiseG
A.A.M
Antoined
AdalgrimT
Bandobras"
BelladonnaT
EglantineT
MirabellaT
FerumbrasI
GerontiusT
[dave@localhost ~]$
```

Example of command line code

15: Print the last name of those not in the 916 area code.✔

> awk -F '[:]' '!($3=="(916)") {print $2}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- ! means "not" or "does not contain" (916)
- $2 will print last names

- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F '[: ]' '!($3=="(916)") {print $2}' AwkLab.data
Vimes
Stibbons
von
Sto
Aching
Belle
Baggins
Took
Gamgee
de
Took
"Bullroarer"
Took
Took
Took
III
Took
[dave@localhost ~]$ _
```

Example of command line code

16: Print each record preceded by the number of the record.✔

> awk '{print NR, $0}' AwkLab.data

- awk is the command.
- NR (number of record)
- $0 will print entire line
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk '{print NR, $0}' AwkLab.data
1 Samuel Vimes:(510) 548-1278:250:100:175
2 Ponder Stibbons:(408) 538-2358:155:90:201
3 Angua von Überwald:(206) 654-6279:250:60:50
4 Susan Sto Helit:(206) 548-1348:250:100:175
5 Tiffany Aching:(206) 548-1278:15:188:150
6 Sacharissa Cripslock:(916) 343-6410:250:100:175
7 Adora Belle Dearheart:(406) 298-7744:450:300:275
8 Frodo Baggins:(206) 548-1278:250:80:75
9 Tom Bombadil:(916) 348-4278:250:100:175
10 Peregrin Took:(510) 548-5258:50:95:135
11 Samwise Gamgee:(408) 926-3456:250:168:200
12 A.A. Milne:(916) 440-1763:175:75:300
13 Antoine de Saint-Exupery:(123) 978-6432:250:100:175
14 Adalgrim Took:(345) 978-7684:4673:100:467
15 Bandobras "Bullroarer" Took:(453) 978-3534:6753:368:4673
16 Belladonna Took:(123) 978-5754:356:247:175
17 Eglantine Took:(123) 978-3574:473:475:4367
18 Mirabella Took:(345) 978-2677:783:563:175
19 Ferumbras III Took:(563) 978-753:250:100:3457
20 Gerontius Took:(574) 978-8535:535:678:4562
[dave@localhost ~]$
```

Example of command line code

17: Print the name and total contribution of each person.✔

> ⓘ awk -F '[:] '{sum=$5+$6+$7} {print $1$2, sum}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- sum of columns 5, 6, and 7 are contributions
- $1 $2 sum will print names and total contributions
- AwkLab.data is the file we are using.

```
[dave@localhost ~]$ awk -F '[ :]' '{sum=$5+$6+$7} {print $1$2, sum}' AwkLab.data
SamuelVimes 525
PonderStibbons 446
Anguavon 964
SusanSto 898
TiffanyAching 353
SacharissaCripslock 525
AdoraBelle 1048
FrodoBaggins 405
TomBombadil 525
PeregrinTook 280
SamwiseGamgee 618
A.A.Milne 550
Antoinede 1328
AdalgrimTook 5240
Bandobras"Bullroarer" 8099
BelladonnaTook 778
EglantineTook 5315
MirabellaTook 1521
FerumbrasIII 1328
GerontiusTook 5775
[dave@localhost ~]$
```

Example of command line code

18: Add $10 to Tiffany Aching's first contribution.✔

> awk -F '[:] '/^Tiffany/ {print $1 $5 + 10}' AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- ^ selects Tiffany's line
- $1 $5 + 10 will print Tiffany with 10 + first contribution
- AwkLab.data is the file we are using.

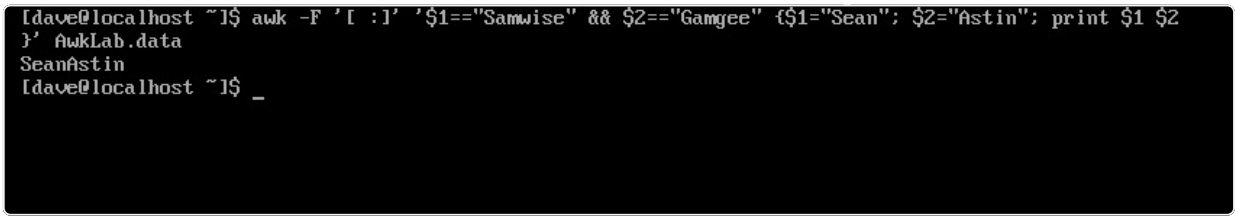
```
[dave@localhost ~]$ awk -F '[ :]' '/^Tiffany/ {print $1 $5 + 10}' AwkLab.data
Tiffany25
[dave@localhost ~]$ _
```

Example of command line code

19: Change Samwise Gamgee's name to Sean Astin✔

awk -F '[:] '$1=="Samwise" && $2=="Gamgee" {$1="Sean"; $2="Astin"; print $1$2 AwkLab.data

- awk is the command.
- -F file separator.
- : separates file
- $1=="Samwise $2==Gamgee will find line with name
- $1="Sean" $2="Astin changes the column text to Sean Astin
- $1$2 prints name to show changes
- AwkLab.data is the file we are using.



Example of command line code

20: Write an awk script to do the following (MUST be an awk script not just a bash script or commands on the commandline)✔️

1: Prints first name of the all the Tooks followed by their total campaign contributions .
✔️

2: Prints "Bullroarer"'s contributions after his name✔️

3: Prints all the names and last month's contribution of those who contributed over $175 for their last contribution✔️

Citations👩🏼‍💻

Awk Overview
https://flylib.com/books/en/4.356.1.53/1/
6.1

Awk Commands
https://www.gnu.org/software/gawk/manual/gawk.html

PDF Template
slite.com

Technical Support Contact Information 💬

David Belmont
00288849@student.necc.edu