Samnang San

Linux Administration

October 30, 2021

Awk assignment

Awk stands for the first initials of last names of the authors of the language. Awk is a UNIX language programming language used for manipulating data and generate reports. This command can manipulate data which is used for shell scripts and managing small databases. To better understand AWK, we will be going through some examples to help you better understand awk.

1. Print all the First Names.

Command: awk ‘{print $1}’ AwkLab.data

We begin with command awk followed by ‘{print$1}’, $ means beginning of the line and the 1 is the first item in the line. Followed by the name of the file. In this example, its AwkLab.data. Example below.

Graphical user interface, text, chat or text message

Description automatically generated

2. Print phone numbers for Tom and Frodo

Command: awk -F: ‘/Tom|Frodo/{print $2}’ AwkLab.data

-F is used to declare a separator, in this example we are using: as the separator. The | between Tom and Frodo would include both. {print $2} prints out the next item which is the phone number. If I type {print $1}, it will print out the first item which would be Frodo Baggins and Tom Bonbadil.



3.Print Peregrin’s name and phone number area code.

Command: awk -F: ‘/Peregrin/{print $1,$2}’ AwkLab.data

First command awk, followed by -F again, I explained it above. The: is the separator, search for Peregrin and print out first item and second item using {print $1, $2}



4. Print all phone numbers in the 408-area code.

Command: awk -F: ‘/408/{print $2}’ AwkLab.data

Awk command followed by -F, /408/ searches for anything that has 408 in it. {print $2} prints item from field 2.



5.Print all Last names beginning with either a B or a D.

Command: awk ‘$2~/^B|D/{print $2,$3}’ AwkLab.data

Awk followed $2~ which means the second item. ^ means the beginning. For /^B|D/, it will match anything for the letters B and D in the beginning of the last name. {print $2, $3} will print the match. Usually I would probably just do {print $2} but some of the names have middle names. Followed by the name of the file.



6. Print all First names containing four or less characters.

Command: awk ‘$1~/^[a-zA-Z]{,4}$/{print $1}’ AwkLab.data

As always, we begin with awk followed by print, $1~ is the first field followed by /^ which signifies the beginning. [a-zA-Z] stands for the letters/characters. {,4} is the number of characters. {print $1} prints out the first name. followed by name of file.



7.Print first names of all those in the 916-area code

Command: awk ‘/916/{print $1}’ AwkLab.data

Awk followed by /916/ which means anything containing in it, print the first name with {print $1}. Followed by name of file

Text

Description automatically generated

8.Print Sacharissa’s campaign contributions.

Command: awk -F: ‘/Sacharissa/{print $1,”$”$3,”$”$4,”$”,$5}’ AwkLab.data

Awk command followed by -F for separation by: . /Sacharissa/ =Anything with Sacharissa in it print name first then added “$” before the 3rd, 4th and 5th items. The $ character appears before the integers like below.



9.Print last name followed by a comma and the phone number.

Command: awk -F ‘[ :] ’ ‘/ ()/{print$2 “,” $3,$4,$5}’ AwkLab.data

This one was difficult, the only pattern I can think of was the parenthesis. I used the -F but there had to be a space and the: as the separator. {print$2 “,” $3,$4,$5} $2 is the last name followed by a string with the quotes “,” to separate the last name and the phone number. $3/4/5 was added to have the phone number printed.

Text

Description automatically generated

10.

a. Print first name of the Tooks followed by their total campaign contributions.

b. Print “Bullroarer”’s contributions.

c. Print all those who contributed over $175 for their last contribution

Text

Description automatically generated

Commands were tested ahead of adding them to a script.

Begin by typing on the command line: vim (name of file to make). Awk

You will be brought a screen like this one below, begin typing the letter “I” to insert

Graphical user interface

Description automatically generated

Then type in #!/usr/bin/awl -f like the one below. FS=’:”, I am just declaring the separator just like the -F commands. As you can see below, I have transferred the awk commands into this script. Once completed like below, Hit the Esc button on your keyboard. And type :wq, to save changes.

Text

Description automatically generated

To run the script type in command prompt awk -f (name of file).awk then the name of the file you want to run the script on. Another way to do this is if gives permission to user.

For example, I created a file called number10.awk

Once created with scripts, I can run command: chmod +x number10.awk, this will give more permissions (read, write and execute).

After that is complete, I just have to type in: ./number10.awk (name of file)

In this example, its ./number10.awk AwkLab.data

**Part 2**

1. Print first and last names of those who contributed more than $110 in the last month.

Command: awk -F: ‘$5>110{print $1}’ AwkLab.data

Awk followed by -F: which is separating fields by a : . $5 represents the last months field, >110 = greater than 110. If field 5 is greater than 100, print their first and last name.

Text

Description automatically generated

2.Print the last name and phone numbers of those who contributed less than $75 in the first month.

Command: awk -F: ‘$3<75{print$1 $2}’ AwkLab.data

Awk command, -F: field separation with : . $3 represents the first month, <75 represents less than 75. {prints $1,$2} represents name and phone number.



3. Print the first name of those who contributed between 75 and 150 in the first month.

Command line: awk -F: ‘$3>75 && $3<150 {print $1}’ AwkLab.data

Awk command, -F: as the separator. $3 represents the first month. If field 3 is greater than 75 and less than 150. {print$1} would print out the name. followed by the name of the file.

Text

Description automatically generated

4.Print first and last name of those who contributed less than 700 over the three months.

Command line: awk -F: ‘{sum=$3+$4+$5} sum <700{print $1}’ AwkLab.data

Awk command followed by field separator -F:. {sum=$3+$4+$5} is the total of three months. If some is less than 700 then print out first and last name. followed by name of file.

Graphical user interface, text

Description automatically generated

5.Print the first and last name of those with an average monthly contribution greater than $100.

Command line: awk -F: ‘{avg= ($3+$4+$5)/3} avg >100 {print$1}’ AwkLab.data

Awk followed by separator -F:, {avg= ($3+$4+$5)/3} gives me the avg of the 3 contributions. If the avg is over 100, {print$1} prints out person’s name. name of file at the end.

Graphical user interface, text

Description automatically generated with medium confidence

6.Print the last name of those not in the 916 area code.

Command line: awk ‘$2 !~/916/{print $2}’ AwkLab.data

!~ means not matching. Area code 916 doesn’t not match item 2, it prints out item 1 that doesn’t have anything with 916 in it.

Graphical user interface, text

Description automatically generated

7.Print each record preceded by the number of the record.

Command line: awk ‘{print NR,$0}’ AwkLab.data

NR means line number. So by adding the {print NR,$0}, it adds a number to each line.

Text

Description automatically generated

8.Print name and total contribution of each person.

Command line: awk -F: ‘{sum=$3+$4+$5} {print$1, sum}’ AwkLab.data

Awk followed by -F: as the separator. {sum=$3+$4+$5} gives us the answer for sum. Then {print$1, sum} prints out the name followed the calculated sum. Always have the file name at the end.

Text

Description automatically generated

9.add $10 to Tiffany Aching’s first contribution.

Command line: awk -F: ‘/Tiffany/{print $3 + 10}’ AwkLab.data

Awk followed by -F: as separator. /Tiffany/ is looking for any line that has tiffany in it. {print $3 + 10} finds her first contribution and adds 10 to it. Last but not least, don’t forget name of file. At times,I do forget then I wonder what I missed.



10. Change Samwise Gamgee’s name to Sean Astin

Command line: awk ‘{sub(/Samwise Gamgee/,”Sean Astin”);print}’ AwkLab.data

Sub is used to substitute followed by the string in quotes you want to change it to.

Awk followed by sub. /Samwise Gamgee/ is name we want to substitute. In this example, we are changing it to Sean Astin which has to be in quotations like this , “Sean Astin”. Then print followed by name of file.

Text

Description automatically generated

Sources used

https://zetcode.com/lang/awk/

<https://flylib.com/books/en/4.356.1.53/1/>

<https://www.youtube.com/watch?v=0dk-WxxTJdw>

<https://riptutorial.com/awk/example/14013/string-text-substitution>