AWK Lab

By Nicholas Mueller

Sources used :

“Learning Awk Is Essential For Linux Users”

<https://www.youtube.com/watch?v=9YOZmI-zWok&t=655s>

“Scripts 1.4 - AWK : remplacement avec gsub | tutos fr” – note the language is French in this video.

<https://www.youtube.com/watch?v=QE17uz_movY>

“EVERYONE Needs to Learn a Little Bit of AWK!”

<https://www.youtube.com/watch?v=jJ02kEETw70>

The database contains the names , phone numbers , and money contributions to the party campaign for the past three months.

-1 Print all the First Names.

This command tells awk to print the first column which is first names denoted by $1 the command its enclosed in ‘quotation’ marks and curly brackets, following is the file name AwkLab.data.

Command: awk '{print $1}' AwkLab.data

Graphical user interface, text

Description automatically generated

-2 Print phone numbers for Tom and Frodo

For this problem I use -F: which tells awk to use a colon as the field separator then within the awk command the /Tom/ is a regular expression that searches for the pattern Tom I do the same thing for the pattern Frodo with /Frodo/. I then tell awk to print fields 1 and 2. $1 and $2 inside of the curly brackets.

Command: awk -F: '/Tom/{print$1 $2}/Frodo/{print $1 $2}' AwkLab.data

A black background with white text

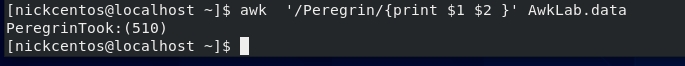
Description automatically generated with low confidence



-3 Print Peregrin's name and phone number area code.

With this awk command I tell awk to look for the pattern Peregrin with /Peregrin/ then I tell awk to print the first two fields of the matching line

Command: awk '/Peregrin/{print $1 $2 }' AwkLab.data



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

Here I tell awk to search for the pattern 408 using -F: to make a colon the field separator, and I print the first and second fields using {print $1 $2 }.

Command: awk -F: '/(408)/{print $1 $2 }' AwkLab.data



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

This problem I use /B/ and /D/ searching for the pattern B and D and printing the second and third fields, there are some extraneous results in the output.

Command: awk '/B/{print $2}/D/{print $3}' AwkLab.data

Text

Description automatically generated

-6 Print all first names containing four or less characters.

To accomplish this I tell awk to print the first field (first names) that are 4 or less characters denoted with the length <=4 syntax.

Command: awk 'length($1) <=4' AwkLab.data

Graphical user interface, text

Description automatically generated with medium confidence

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

I search for the pattern 916 using /916/ and tell awk to print the first field $1.

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

Text

Description automatically generated

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

In this command I tell awk that the field separator is a colon with -F: then I search the pattern /Sacharissa/ which I then print the 3rd 4th and 5th fields. These fields are preceded by a $ which I tell awk to print by putting in the “$”.

Command: awk -F: '/Sacharissa/{print"$" $3, "$" $4, "$" $5 }' AwkLab.data

Graphical user interface, text, application

Description automatically generated

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

In this problem I tell awk to use a colon to separate the fields, I print field $1 then I tell awk to put in a comma with “,” and print field $2.

I also am printing out the first and middle names I could not find a way to sperate them from the rest of the line.

Command: awk -F: '{print $1,","$2}' AwkLab.data

Text

Description automatically generated

-10 Write an awk script called facts to do the following (MUST be an awk script not just a bash script or commands on the commandline)

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

Prints "Bullroarer"'s contributions.

Prints all those who contributed over $175 for their last contribution

\*For this problem I wrote a script that I think does all of these.

Shown on next page

Script Graphical user interface, text, application

Description automatically generated

For the script I set the field separator to be a colon

I tell awk to look for the pattern Took: and then print the first field then the sum of the 3rd 4th and 5th fields

Next I tell awk to print field 5 if it’s greater than 175 for this the full line is shown for those contributors

Command: ./Awkscript AwkLab.data

Text

Description automatically generated

Awk (use the AwkLab.data file)

The database contains the names , phone numbers , and money contributions to the party campaign for the past three months.

-11 Print the first and last names of those who contributed more than $110 in the last month.

I tell awk to use a colon as the field separator, then I tell awk to look at field 5 with $5 if it’ greater than the value of 110 awk is told to print the first and last names field 1 or $1.

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

Text

Description automatically generated

-12 Print the last names and phone numbers of those who contributed less than $75 in the first month.

For 12 I tell awk to use a colon as a field separator, then check field 3 if it’s less than 75 the first and second fields are then printed displaying the last name and phone number.

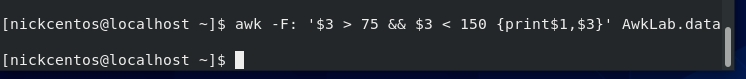
Command: awk -F: '$3 <75 {print$1,$2}' AwkLab.dataGraphical user interface, text, application

Description automatically generated

-13 Print the first names of those who contributed between $75 and $150 in the first month.

For this problem I tell awk to use a colon as the delimiter and print fields 1 and 3 if they are greater than 75 and less than 150 (No results were found)

Command: awk -F: '$3 > 75 && $3 < 150 {print$1,$3}' AwkLab.data



-14 Print the first and last names of those who contributed less than $700 over the three-month period.

For this problem I tell awk to use a colon as a field separator, then if fields 3,4 and 5 added together are less than 700 field 1 is printed.

Command: awk -F: '$3+$4+$5 < 700 {print$1}' AwkLab.data

Text

Description automatically generated

-15 Print the first names and first letter of the last name of those with an average monthly contribution greater than $100 .

For this problem the average is over $100 for everyone except Peregrin Took which is line 10. So I tell awk that I want to use a lower case character [a-z] and more than 1 character with \*. Then a colon all as my field separator, next I don’t want Peregrin Took in my output because he’s less than $100 per month average so I tell awk not to print line 10 with the command NR(number of records) != not equal to 10 finally I tell awk to print field 1 which is the first name and the first letter of the last name of those with $100+ monthly average.

Command: awk -F"[a-z]\*:" 'NR!=10{print$1}' AwkLab.data

Text

Description automatically generated

-16 Print the last name of those not in the 916 area code.

I tell awk that if the second field does not match with !~ the pattern (916) then print the second and third fields showing last name and that the area code is not 916.

Command: awk '$2 !~(916) {print$2,$3}' AwkLab.data

Text

Description automatically generated

-17 Print each record preceded by the number of the record.

This command tells awk to print the NR (number of records) along with the entire line in the file after that record number.

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

Text

Description automatically generated

18 Print the name and total contribution of each person.

I’m telling awk to use a colon as a field separator then I tell awk to print the first field then add a dollar sign to the output, and print the sum of columns 3,4, and 5.

Command: awk -F: '{print$1,"$"$3+$4+$5}' AwkLab.data

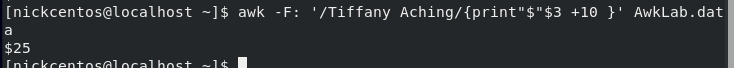
Text

Description automatically generated

-19 Add $10 to Tiffany Aching's first contribution.

I tell awk to use a : as the field separator, then to look for the pattern Tiffany Aching in the file. Next I tell awk to add a dollar sign and print the 3rd column’s value plus 10 .

Command: awk -F: '/Tiffany Aching/{print"$"$3 +10 }' AwkLab.data



-20 Change Samwise Gamgee's name to Sean Astin

For this command I tell awk using gsub in my command; to substitute the pattern Samwise Gamgee with Sean Austin. I then tell awk to print columns 1 and 2 a semicolon separates the gsub from the print command.

Command: awk '{gsub("Samwise Gamgee","Sean Austin");print $1,$2}' AwkLab.data

Text

Description automatically generated