Brian Santana

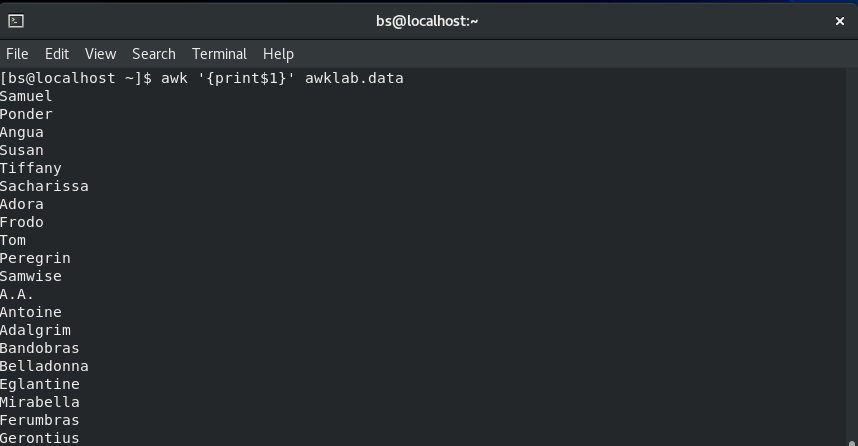
Linux Administration

**Lab 4 Awk**

1. Print all the First Names.

To print all the first names you need to input $1 since the fields are divided by spaces and the first field is where the names are located

Input: awk ‘{print$1}’ awklab.data



1. Print phone numbers for Tom and Frodo

We use -F: as a separator with Tom|Frodo being the search term and print$2 for where the phone number is located

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



1. Print Peregrin's name and phone number area code.

Pretty much the same with 2 except this time we put a ^ infront of peregrin to signify the the beginning of the line and print $1 to put the name in

Input: awk ‘/^Peregrin/ {print$1$2}’ awklab.data



1. Print all phone numbers in the 408 area code.

Similar to 2 but this time you won’t use the ^ to signify the start of the line

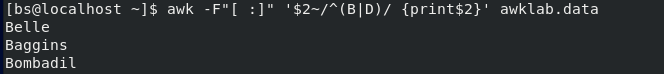
Input: awk -F: ‘/(408)/ {print$2}’ awklab.data



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

Use -F as a separator again then use $2 with a ~ to signify a match in which column then add the ^ with (B|D) so it knows to search for last name letters b and d

Input: awk -F”[ :]” ‘$2~/^(B|D)/ {print$2}’ awklab.data



1. Print all first names containing four or less characters.

We use length($1) to return the value of the all the names then we use <=4 to print anything less than or equal to 4

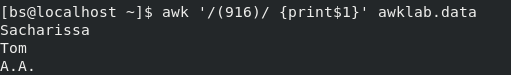
Input: awk ‘ length($1) <=4 {print $1}’ awklab



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

You’ll be doing something similar to 4 except this time you won’t need -F for the seperator

Input: awk ‘/(916)/ {print$1}’ awklab.data



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

We use -F as a separator again then enter Sacharissa as the search term then add “$” behind each numbered $ to give it a dollar sign with the campaign contributions in each respective column

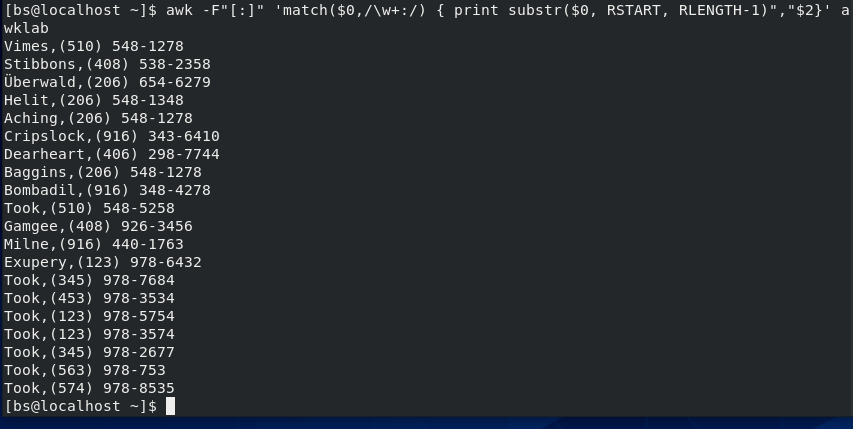
Input: awk -F: ‘/Sacharissa/ {print”$”$3,”$”$4,”$”$5}’ awklab.data



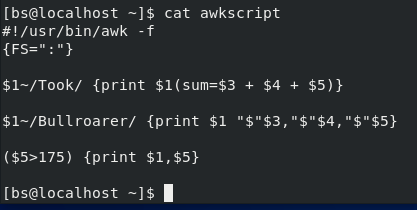
1. Print last names followed by a comma and the phone number.  Be careful of the last names's format.

We use match to search the entire line for the last names \w to search for the characters of the last name substring to print out the last name rstart stores the location of the string and rlength the strings length

Input: awk -F”[:]” ‘match($0,/\w+:/) { print substr($0, RSTART, RLENGTH-1)”,”$2}’ awklab

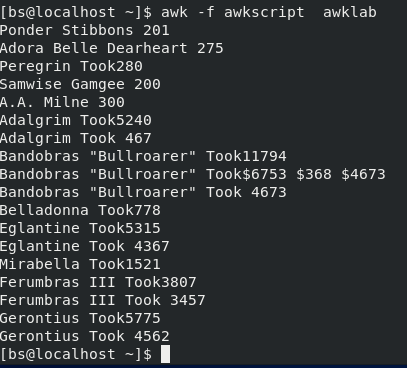


1. Write an awk script called facts to do the following (MUST be an awk script not just a bash script or commands on the commandline)

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

Similar to 8 Instead enter $1~ to match Took with // then print “$” $3 + $4 + $5 so it prints out the sum of their contributions  
  
        Prints "Bullroarer"'s contributions.

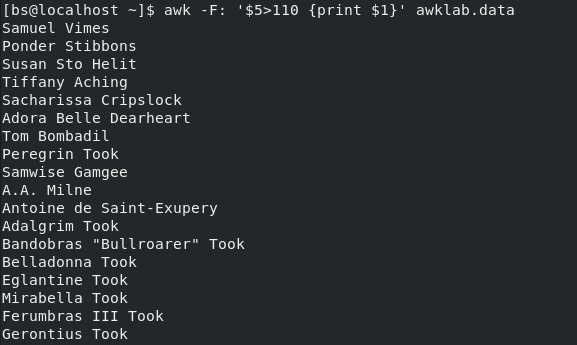
Also similar to 8 and the previous entry enter $1~ /bullroarer/ instead of sacharissa in the //   
  
        Prints all those who contributed over $175 for their last contribution

The last contributions are located in the 5th column so you’re going to put a greater than sign over so it only prints out that number if its larger than 175 and it should look something like this ($5>175){print$1,$5}  


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

Similar to the last part of question 10 we need a greater than symbol in front of $5 the only difference is we don’t need to print out the contributions for the last month so it looks like the input below

Input: awk -F: ‘($5>110) {print $1}’ awklab.data



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

Similar to the problem above but this time around you put $3 instead of a $5 since that’s where the first month contributions are and when printing we add a $2 so it adds the phone number of the person

Input: awk -F: ‘$3<75{print $1,$2}’ awklab.data



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

For this problem you’ll need to use an and statement which is && for $3>75 and $3<150 to get the values in between 75 and 150 though there won’t be any results since nothing in the document matches the criteria

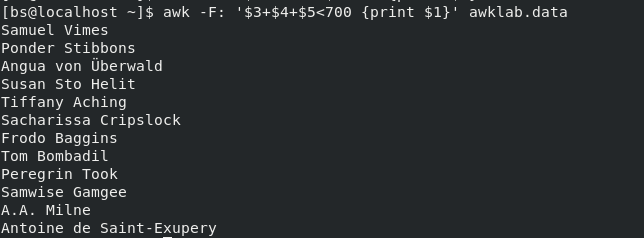
Input: awk -F ‘$3>75 && $3<150 {print $1}’ awklab.data



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

Similar to the first part of 10 we need to add up all the contributions and see who got less than 700 by putting a less than sign and printing their names like the input below

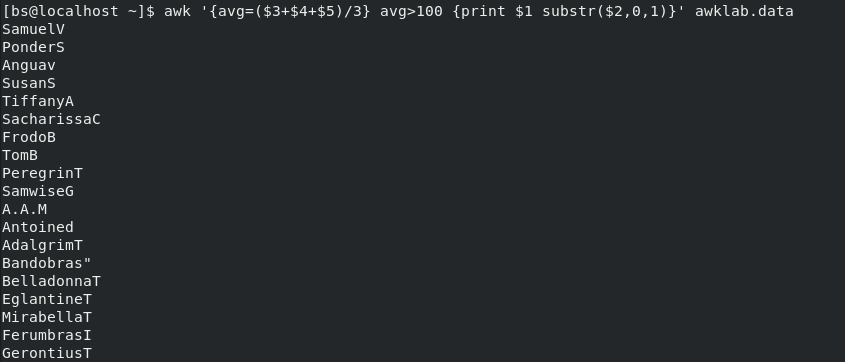
Input: awk -F: ‘$3+$4+$5<700 {print$1}’ awklab.data



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

First we need to make a variable for average by adding all 3 contributions then dividing them by 3 so it looks like this {avg=($3+$4+$5)/3} then compare the average value with a 100 then have it print the first name and last name initial with substr

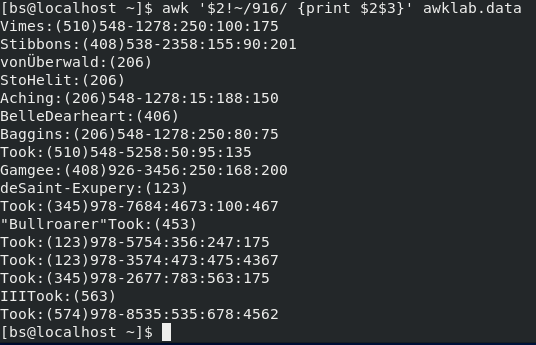
Input: awk ‘{avg=($3+$4+$5)/3} avg>100 {print $1 substr($2,0,1)}’ awklab.data



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

Same deal as seven but this time we put $2!~ to signify not matching and print all those results

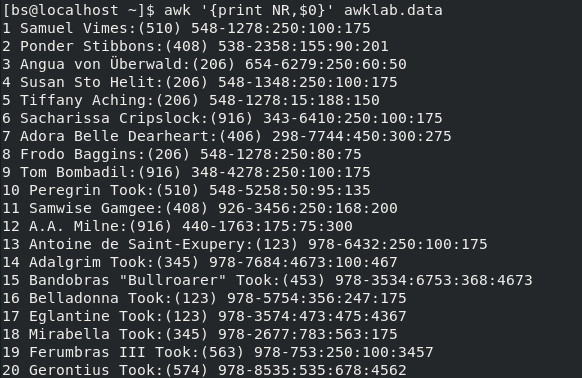
Input: awk ‘$2!~/916/ {print $2$3}’ awklab.data



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

This one is pretty straight forward enter {print NR, then $0} NR counts the number of lines

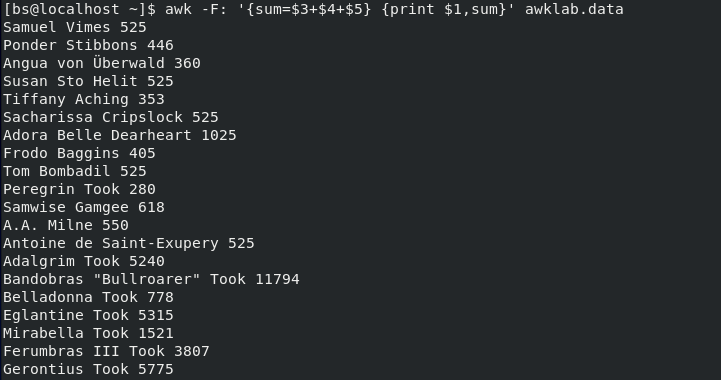
Input: awk ‘{print NR,$0}’ awklab.data



1. Print the name and total contribution of each person.

Similar to 5 we create a variable in this case sum add a separator -F: then we add each contribution under the name of the variable and print $1

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



1. Add $10 to Tiffany Aching's first contribution.

We need to use a separator -F: and the ~ to get the exact match to add we just use $3 +10 since the first contribution is located in the 3rd column

Input: awk -F: ‘$1~/Tiffany/ {print $3 + 10}’ awklab.data



1. Change Samwise Gamgee's name to Sean Astin

To change the name we use the sub command with /Samwise Gamgee/,”Sean Astin”);print

Input: awk ‘{sub(/Samwise Gamgee/,”Sean Astin”);print $1$2}’ awklab.data



***Sources:***

https://flylib.com/books/en/4.356.1.71/1/

https://flylib.com/books/en/4.356.1.80/1/

<https://www.youtube.com/watch?v=4HpWO_RAMq4>

<https://phoenixnap.com/kb/awk-command-in-linux>

<https://linux.die.net/man/1/awk>