Jason De Leon

Awk Lab

00305640

3/30/2023

Awk Lab

The reason why someone would use awk is used to search and find data for those who request it.

File Transfer:

In order to do any anything please remember to transfer the file over to the Virtual Machine. To do this you would need to the following.

The file is provided by the teacher on blackboard.

*“scp AwkLab.data* [*jason@192.168.1.194:~/AwkLab.data*](mailto:jason@192.168.1.194:~/AwkLab.data)*”*

Text

Description automatically generated

This is what it looks like on the VM side:



Part 1:

We will need to print all the first names.

This is the command we would be using:

The $ is used to identify the field while the ‘ make is the command understands that this means ONE command and not two.

*“awk ‘{print $1}’ AwkLab.data”*

$1 means the first field. Each field end and begins with :

This is the outcome of the command:

Text

Description automatically generated

Part 2: We will be printing the phone number for the following people, Tom and Frodo

The command we will be using is the following:

*“Awk -F: ‘/Tom|Frodo/{print $1$2}’ AwkLab.data”*

This is what the command means. /Tom|Frodo/ searches for those two people then print $1 and $2 means to please search for their names and what follows and this would be phone numbers.

The reason why I used “|” was because it means “or”. We are looking for not just one person.

This is the following result:



Part 3: Print Phone Number Area code and name as well for only Peregrin:

***(Awk ‘/Peregrin/{print $1 $2}’ FS=, AwkLab.data)***

The command will search for Peregrin and the {print $1$2} will print the first and second field.



Part 4: Print all Cell numbers in the 123 area code and names

***(Awk ‘/(123)/{print $1 $2 $3}’ FS=- AwkLab.data)***

The (123) is the part of the program where it looks for anything that contains 123 in the same sequence then {print $1$2$3} prints first to third field.



Part 5: Print last names that only start with T or D

***(Awk -F ‘[: ]’ ‘$3$2~/(T|D)/{print}’ AwkLab.data)***

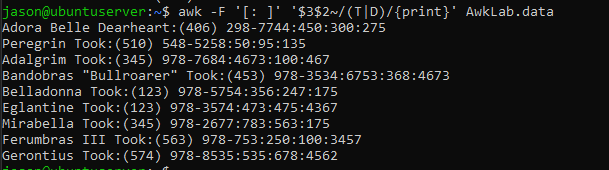
The reason why we include a -F ‘[: ]’ is for those with middle names. They become parameter 2 and their last names become parameter 3 or less it’ll thing middle name and last name are the same thing causing us to miss Adora.

We do ‘$3$2’ so I can look at the last name for those who have a middle name. It’ll skip those without middle names since their parameter 3 doesn’t have either letter.

$2 will skip the middle names since they don’t have the letters then go for the last name which or $2 for them.

If you were to remove the $3$2~ you will see everyone who start with T or D in either their first names or last names.

Now T|D just means that you would like to have both searched.



Part 6: Print all first name with four or less characters:

***(awk ‘{if(length($1)<=4){print $1}}’ AwkLab.data)***

This one works like the way you read it. IF the length of first field is less than or equal to 4 then print first field.

Text

Description automatically generated

Part 7: Print first names and those in the 916 Area code:

Similar to step 4 we tell the command what numbers we want by stating it first.

(916) tells it that this is what we want and then / tells it to end it since we’ve place one at the biggening. Print {$1$2} gives us the 1st and 2nd fields. You can add more friends by doing $3.

***(Awk ‘/(916)/{print $1 $2}’ AwkLab.data)***

Text

Description automatically generated

Part 8:Print Sacharissa’s campaign contributions following her name:

***(Awk -F: ‘/Sacharissa/{print $1 “ $”$3 “ $”$4 ” $”$5”}’ AwkLab.data)***

In this command we search for Sacharissa then we print the parameters 1 being their name and then 3-5 which contain their contributions. By us putting “ $” before the parameters we create a space between the numbers and add the $ sign next to it.

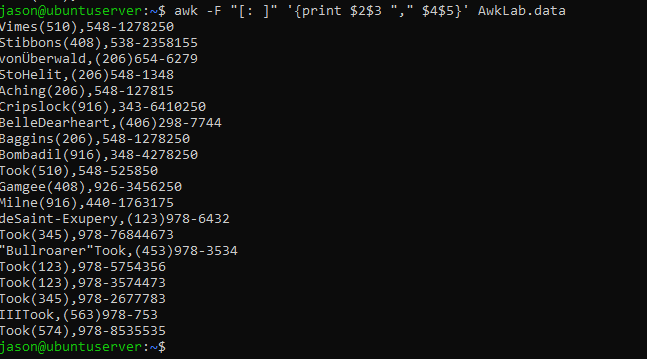


Par 9: Print last names followed by a comma and the phone number:

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

The reason why this time $2$3 prints the last name ($3 for those that have a last name) before the numbers is because of what we did at the beginning of the command. “[:]” essentially makes the first name become parameter 1 and the last name number 2.

To make the command add the “,” we must include it after the parameter number 2 as that follows the last name and then 4 and 5 are their cell numbers.



Part 10: Print those to contributed over $110 in the last month and to include their full name before the given amount:

***(awk -F ‘[: ]’ ‘{if(($7)>+110){print $1,$2,$7}}’ AwkLab.data***

Since we are doing addition here we would like to create a new parameter and make it so we can call it constantly in order to do this we make the first name a parameter “[:]” and create another parameter called 7.

If parameter 7 more than 110 then we shall print $1 First name, $2 Last name and finally the total of their contributions $7.

Text

Description automatically generated

Part 11: Print Last name, cell and first month contribution less than 150 the first month:

***(awk -F ‘[: ]’ ‘{if(($5)<=150){print $2,$3$4,$5}}’ AwkLab.data)***

Since we want only the last name and not the full name we will turn the first name into a parameter itself.

This means that the first month contributions will become number 5 instead of 4.

Now the If statement means the following. If Parameter 5 aka first month is less then or equal to 150 we will print the Last name $2 and their full cell number $3$4 and finally their first month contributions which is $5

Text

Description automatically generated

Part 12: Print first name of those who contributed between 10-200:

***(awk -F ‘:’ ‘$5<=200 && $5>=10 {print $1,$5}’ AwkLab.data)***

The command does the following it looks at the parameter 5 and checks if the number itself is under 200 but at the same time if it’s bigger than 10. This is done in this part of the command $5<=200 && $5>=10

Text

Description automatically generated

Part 13: Print full name and contributions less than $700 over 3 months:

***(awk -F ‘[: ]’ ‘{if(($5+$6+$7)<700) {print $1,$2,($5+$6+$7)}}’ AwkLab.data)***

The following command does this.

If Parameters 5-7 added up is than 700 then it shall print both parameters 1 and 2. Then it will add up the Parameters of 5-7 and print them. If the addition is more than 700 it will be skipped for that line.

Text

Description automatically generated

Part 14: Print first name and contributions that averages over 300:

***(awk -F ‘[: ]’*** ***‘($5+$6+$7)/3>300 {print $1,$5+$6+$7/3}’ AwkLab.data)***

‘($5+$6+$7)/3>300 does the following. It will add parameters 5-7 and then divide it by 3. Finally, it’ll check if the math is over 300.

If the final math is over 300 it will print Parameters of 1,5-7 and then divide it by three.

Text

Description automatically generated

Part 15: Print Last Name for those not in the 916 area Code:

***(awk -F ‘ [: ]’ ‘!($3==”(916)”){print $2, $3, $4}’ AwkLab.data)***

We add the ‘[: ]’ makes it so we can select and print the last name solely.

! means “NOT” so if parameter 3 is NOT equal (==) 916 then print Last name ($2) and the rest of their phone numbers ($3$4)

Text

Description automatically generated

Part 16: Print each record by numbers:

***(awk ‘{print NR, $0}’ AwkLab.data)***

NR basically tells the total numbers of lines and since we gave the command a parameter of $0

We basically are asking that the VERY beginning of the command please state the number of this lines.

A black screen with white text

Description automatically generated with low confidence

Part 17: Print the names and their Contributions:

***(awk -F ‘[: ]’ ‘{print $1, $2, $5+$6+$7}’ AwkLab.data)***

We are printing the first and and Last name then following by the total contribution made by each person. The math is done for us when we did this $5+$6+$7.

Text

Description automatically generated

Print 18: Print Tiffany’s full name with their first contribution and one with 10 extra:

***(awk -F ‘[: ]’ ‘ /^\<Tiffany Aching\>/{$5=$5+10;print}’ AwkLab.data)***

Here we will be searching for Tiffany Aching hence the reason why we have the name in the slashes. It’s to indicate the person we would like and what followings.

If you try by doing $1 == Tiffany nothing will print.

Then by selecting the first contribution with $5 and doing the following $5 =$5 +10 we are telling it that $5 is now 10 more than it previously was.



Part 19: Name change for Samwise to Sean Austin:

***(awk -F ‘[: ]’ ‘$1==”Samwise” && $2==”Gamgee”{$1=”Sean”;$2=”Austin”; print$1,$2,$3}’ AwkLab.data)***

Like the pervious one in terms of changing data.

Here we will be changing the name of person and in order do that we will have to create a new parameter which is done by [: ]. This tells the AWK command that the first and last names are their own parameters and can be called upon.

$1 == “Samwise” is the first name

&& means AND which tells it that both things are crucial, and it must keep it cached.

$2 == “Gamgee” is the second name

{$1 =”Sean;$2 =”Austin”;}

The colons is a separator. Tells it that the command that it most be done step by step.

Finally the print will print out the new name since we established that $1and $2 are no longer Samwise Gamgee but now Sean Austin.



Part 20: Write an awk script to meet the requirements (not bash or commands in the command line):

(a) Prints first name of the all the Tooks followed by their total campaign contributions.

(b) Print the full names and contributions of anyone who contributed between $10 and $200 in the last contribution

(c) Prints the full names and average contribution of those who contributed less than $300 on average

To Create the script I recommend to use text editor. This is the command for creating the file. I named mine ***awkscript.txt.***

***(vi awkscript.txt***)

To make any change please click “INS” or “Insert” on your keyboard.

To exit edit mode you have to hit “Cntr + C”

In order to Save you have to the previous step and the type this: (:wq) it will do the following WRITE and QUIT.

Please reference the picture in order to write the script.

Text

Description automatically generated

Finally, to run the script you must type the following command:

***(awk -f awkscript.txt AwkLab.data)***

Text

Description automatically generated