**Name: Oscar Ynoa**

**Student Id: 00308560**

**Course: Linux Administration (CIS-245-O1A)**

**Subject: Awk Assignment**

**Professor: Adrianna Holden-Gouveia**

**Due Date: 10/03/2021**

**Important info: While doing this assignment it is important to move to the directory that has the datebook file and type correctly the name of everything. For example, if the file name is called Awklab and you type awklab is another different file.**

1. **Print all the First Names.**

**The command is awk '{print$1}' Awklab**

**To print first names, we need to know how our file is structured.**

**The Name is the first field. The Phone Number is the second field. And the remaining 3 fields are contributions made.**

**That means $1 equals first names. That is also the first field.**

**Texto

Descripción generada automáticamente**

1. **Print phone numbers for Tom and Frodo**

**The command is** **awk -F: '/^Tom|Frodo/{print $2}' Awklab**

**We use -F parameter to declare a separator. In this example we use the colons as the field separator. Same concept as the previous example. We are printing the second field $2 because the phone numbers are located there. The only difference is that we need to write the person’s name’/ Tom|Frodo/’ enclosed in backslash to search for their numbers.**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

1. **Print Peregrin name and phone number area code.**

**The command is awk '/^Peregrin/{print $1" "$2 }' Awklab**

**$1 means the first field. That means that Peregrin name is located there. $2 as the second field. Numbers. The incorporation of the quotations “ “ is to separate them so they don’t stick together. It adds a space between them. So, the outcome between $1 and $2 comes separated.**

****

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

**The command is awk -F: '/(408)/ {print $2}' Awklab**

**We use -F parameter to declare a separator with a colon. We enclosed the (408) between parentheses because we are looking for that certain combination to work with. Then we print the $2 because that is the field were the numbers are located. In summary, we are telling awk to grab the code area (408) and print the rest of the field that is $2.**

****

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

**The command is awk -F"[ :]" '$2~/^(B|D)/{print $2}' Awklab**

**We use -F to create a** **separator by either a space or colon. Printed the $2 because it is now limited by the separator, so this means is going to print the second match allowing us to print the last names with B or D.**

**^: Indicates the beginning of the line**

**~ : Means match**

**Texto

Descripción generada automáticamente**

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

**The command is awk '4 >= length($1) {print$1}' Awklab**

**For this question we need to know what we are looking for. We are going to start this problem delimitating the range of the search with 4 >= length ($1). This expression means look for the first four or less carachters in the first field that the first names are located or $1. Later, for first names we use {print$1} to print the field were the first names are located and going to be searched.**

**Texto

Descripción generada automáticamente**

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

**The command is awk '/(916)/{print $1}' Awklab**

**We identified the code by enclosing it between parenthesis (916). Then printing the first field $1. So, then this will print the first names of the ones containing the area code 916.**

**Texto

Descripción generada automáticamente**

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

**The command is awk -F: '/Sacharissa/{print "$"$3,"$"$4,"$"$5}' Awklab**

**We need awk with the -F parameter to declare a separator by either a space or colon . In this example we use the colons as the field separator. Then we start by searching the name Sacharissa. Then we want awk to print dollars sign for each field. To add them we write a dollar sign “$“. If we do not add the field separator expression, it will only print a dollar sign for only one field. So, we need to use it as this /{print “$”$3, “$”$4, “$”$5’}. This expression means, to print a string $ for the all this fields.**

**Texto

Descripción generada automáticamente**

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

**The command is awk -F: 'match($0,/\w+:/){print substr($0,RSTART,RLENGTH-1),$2}' Awklab**

**Explanation: We need awk with -F to create a separator** **by either a space or colon. Then awk matches the entire line to get the last name of the person. Then \w to identify any character as part of the last name.**

**Substr : can be used to select a substring from the input.**

**RSTART: This stores the starting location of the search string/pattern.**

**RLENGTH: This stores the length of the search string/pattern.**

**Texto

Descripción generada automáticamente**

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 command line)**

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

**We have a similar example in the awk assignment that states this problem.**

**Question Number 8 first part of the assignment. Instead of Sacharissa I took the name of Tooks. Then I proceed to change the form of the expression to add the value or the sum of them. I know the last 3 fields are contributions. Thus, we have the answer of our first question in the script.   
  
 Prints "Bullroarer’s contributions.**

**Same as the past one. Instead of adding. Showing the value. I put the name between // and then I proceed to print their contributions. To add them we write a dollar sign “$“. If we do not add the field separator expression, it will only print a dollar sign for only one field. So, we need to use it as this /{print “$”$3, “$”$4, “$”$5’}. This expression means, to print a string $ for all this fields.  
  
 Prints all those who contributed over $175 for their last contribution.**

**I know the last contribution is in field 5. I going to try to do it like problem number 2 of the second part of the awk assignment. Instead of looking for a minor than < I look for a bigger than > Field $5 Represents the last month. If the is a number bigger or equal >= than 175. It is going to print the value.  
  
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.**

**vi filename.awk** to create an awk script

At the beginning of the editor, we type **#!/usr/bin/awk -f** at the top

**I put the FS=”:” that means FS** : **Field Separator . Same as -F to separate as either a space or colon**

**to run the script :** **awk -f Name of the file + Awklab**

**Note: We don’t need to use “awk” at the begging of an expression.**

**Texto

Descripción generada automáticamente**

**First things firsts. Simple is better for me.**

**Texto

Descripción generada automáticamente**

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

**The command is awk '$4>110 {print$1}' FS=':' Awklab**

**$4 represents the field 4 or the second month. If there is a number “>”greater than 110 in the field 4, print field $1 which correspond to the person full names.**

**Texto

Descripción generada automáticamente**

1. **Print the last names and phone numbers of those who contributed less than $75 in the first month. The command is awk -F':' '{if ($3 < 75) print $1 " " $2}' Awklab | awk '{for (i=2; i<=NF;i++) printf $i " " ;print " "}'**

**For the first part I needed -F to separate** **by either a space or colon. The next part is printing the contributions of the first month of the people that contributed less than 75. We use {if ($3 < 75) to print the first month and the contribution less than 75.**

**Later we use print $1 " " $2 as a subsequence of printing the first and second field were the name, and the number are located. awk '{for (i=2; i<=NF;i++) printf $i " ";print " "}' This command is used to eliminate the first name and allow to print the last name with the phone number.**

**Texto

Descripción generada automáticamente**

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

**The command is awk '$3>75 && $3<150 {print$1}' FS=':' Awklab**

**No records were found. I did not get an outcome from this. I looked at the file and nobody had that range of contribution .**

**FS** : **Field Separator . Same as -F to separate as either a space or colon.**

**&&** = **and**

**Texto

Descripción generada automáticamente**

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

**The command is awk '{x=$3+$4+$5}x<700 {print $1}' FS=':' Awklab**

**Comments: For this method we need to add all the months . But first, we must create a variable. Let’s use as a variable “x”. Within the variable, we added field $3,$4, $5 to identify it as the 3 months. Then in the expression, we said if x is less than < 700, print field $1 which correspond to the person names.**

**Texto

Descripción generada automáticamente**

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

**The command is awk -F: 'match($0,/\w+:/){avg=($3+$4+$5)/3;if(avg>100){print substr($0,0,RSTART)}}' Awklab**

**We use -F to separate** **by either a space or colon. Then we need to match the first letter of their last name. The average variable will be used to combine the contribution and divide it by the amount of contribution. If someone have an average larger than a 100. The name will be printed with the first letter of their last name.**

**Substr : can be used to select a substring from the input.**

**RSTART: This stores the starting location of the search string/pattern.**

**Texto

Descripción generada automáticamente**

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

**The command is awk -F: '!/(916)/{match($0, /\w+:/);print substr ($0,RSTART,RLENGTH-1)}' Awklab**

**We need to use -F to separate as either a space or colon. Then the ! will be used to don’t allow the (916) shows in the expressions of the body. Then for the match to get the last name index and printed it using substring 916 was not their area code.**

**Substr : can be used to select a substring from the input.**

**RSTART: This stores the starting location of the search string/pattern.**

**RLENGTH: This stores the length of the search string/pattern.**

**Texto

Descripción generada automáticamente**

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

**The command is awk '{print NR,$0}' Awklab**

**After doing some research I find the expression to solve this. So, this will display the number of the records in a file.**

**NR** : **The number of records.**

**Texto

Descripción generada automáticamente**

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

**The command is awk '{x=$3+$4+$5}{print $1, x}' FS=':' Awklab**

**We must make a variable and then add all the contributions fields to find the total.**

**FS** : **Field Separator . Same as -F to separate as either a space or colon.**

**We create a variable call it “x” . We add the fields that contain the contributions that are $3,$4, and $5. Then it adds together at the end.**

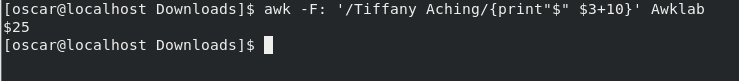
**Texto

Descripción generada automáticamente**

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

**The command is awk -F: '/Tiffany Aching/{print"$" $3+10}' Awklab**

**We first start with awk -F to separate either by a space or colon. $3 represents the first contribution field. Print the first contribution field which is $3 plus 10.**

****

1. **Change Samwise Gamgee's name to Sean Astin**

**The command is awk '{gsub(/Samwise Gamgee/,"Sean Astin");print $0}' Awklab**

**Replace the old name (Samwise Gamgee) with the new name (Sean Astin)**

**gsub**: **Global substitution**. **This one is the function that let us make these changes from a previous name to new one.**

**Texto

Descripción generada automáticamente**

**References**

[**https://flylib.com/books/en/4.356.1.53/1/**](https://flylib.com/books/en/4.356.1.53/1/)

[**https://docstore.mik.ua/orelly/unix3/sedawk/ch09\_02.htm**](https://docstore.mik.ua/orelly/unix3/sedawk/ch09_02.htm)

[**https://www.youtube.com/watch?v=4HpWO\_RAMq4&ab\_channel=AdriannaHolden-Gouveia**](https://www.youtube.com/watch?v=4HpWO_RAMq4&ab_channel=AdriannaHolden-Gouveia)