Created a file name grep.txt with the command:

Echo “contain here” >> grep.txt

1. **Case insensitive search :**The -i option enables to search for a string case insensitively in the give file. It matches the words like “UNIX”, “Unix”, “unix”.

**$grep -i "UNix" geekfile.txt**

1. **Displaying the count of number of matches :** We can find the number of lines that matches the given string/pattern

**$grep -c "unix" geekfile.txt**

1. **3. Display the file names that matches the pattern :**We can just display the files that contains the given string/pattern.
2. **$grep -l "unix" \***
3. **or**
5. **$grep -l "unix" f1.txt f2.txt f3.xt f4.txt**

**4. Checking for the whole words in a file :**By default, grep matches the given string/pattern even if it found as a substring in a file. The -w option to grep makes it match only the whole words.

**$ grep -w "unix" geekfile.txt**

**5. Displaying only the matched pattern :**By default, grep displays the entire line which has the matched string. We can make the grep to display only the matched string by using the -o option.

**$ grep -o "unix" geekfile.txt**

**6. Show line number while displaying the output using grep -n :**To show the line number of file with the line matched.

**$ grep -n "unix" geekfile.txt**

**7. Inverting the pattern match :**You can display the lines that are not matched with the specified search sting pattern using the -v option.

**$ grep -v "unix" geekfile.txt**

**8. Matching the lines that start with a string :**The ^ regular expression pattern specifies the start of a line. This can be used in grep to match the lines which start with the given string or pattern.

**$ grep "^unix" geekfile.txt**

**9. Matching the lines that end with a string :**The $ regular expression pattern specifies the end of a line. This can be used in grep to match the lines which end with the given string or pattern.

**$ grep "os$" geekfile.txt**

Answer:

1. How to search multiple words in a file using single command ? (OR)

🡪 grep ‘unix\|this’ grep.txt

🡪grep -E ‘unix|this’ grep.txt

🡪egrep ‘unix|this’ grep.txt

1. How to find multiple different search strings in a single file ? (AND)

Ans: **$grep ‘chakra\|GMAIL\|unix’ filename.txt**

\*\*grep -E ‘resources’|

🡪grep -E ‘pattern1.\*pattern2’ filename

= grep -E ‘Dev.\*Tech’ file.txt

\*\* The following example will grep all the lines that contain both “Dev” and “Tech” in it (in the same order).

🡪 grep -E 'Manager.\*Sales|Sales.\*Manager' employee.txt

\*\* The following example will grep all the lines that contain both “Manager” and “Sales”

in it (in any order).

🡪Grep AND using Multiple grep command

grep -E 'pattern1' filename | grep -E 'pattern2'

example:

\*\* The following example will grep all the lines that contain both “Manager” and “Sales” in the

same line.

$ grep Manager employee.txt | grep Sales

1. All other commonly used GREP commands in LINUX used for day to day operations

\***for NOT use -v**

🡪grep -v ‘pattern’ filename.txt

You can also combine NOT with other operator to get some powerful combinations.

For example, the following will display either Manager or Developer (bot ignore Sales).

**$ egrep 'Manager|Developer' employee.txt | grep -v Sales**

**AWK UNIK**

**WHAT CAN WE DO WITH AWK ?**

**1. AWK Operations:**  
(a) Scans a file line by line  
(b) Splits each input line into fields  
(c) Compares input line/fields to pattern  
(d) Performs action(s) on matched lines

**2. Useful For:**  
(a) Transform data files  
(b) Produce formatted reports

**3. Programming Constructs:**  
(a) Format output lines  
(b) Arithmetic and string operations  
(c) Conditionals and loops

Syntax:

**awk options 'selection \_criteria {action }' input-file > output-file**

**Options:**

-f program-file : Reads the AWK program source from the file

program-file, instead of from the

first command line argument.

-F fs : Use fs for the input field separator

**1. Default behavior of Awk :**By default Awk prints every line of data from the specified file.

$ awk '{print}' employee.txt

**2. Print the lines which matches with the given pattern.**

$ awk '/manager/ {print}' employee.txt

In the above example, the awk command prints all the line which matches with the ‘manager’.

**3. Spliting a Line Into Fields :**For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

$ awk '{print $1,$4}' employee.txt

In the above example, $1 and $4 represents Name and Salary fields respectively.

1. For instance, to match all entries with the letter ‘o’, the syntax will be

$ awk '/o/ {print $0}' file.txt

### Printing columns that match a specific pattern

### $awk ‘/d/ {print $1 “\t” $4}’ filename.txt

### \*\*here it with search for any row with d and display column 1 and 4 from the filename.txt file. It will separate by “\t” tabs in output

### Counting and Printing Matched Pattern

$awk ‘/usa/{++cnt} END {print “Count= “, cnt}’ filename.txt

\*\*the above command prints number of lines it match for ‘usa’

### Print Lines with More or less than a No. of Characters

$awk ‘length($0)>20’ filename.txt

\*\*will print line that has length more than 20

### Saving output of AWK to a different file

### $awk ‘/usa/ {print $3 “\t” $4}’ filename.txt > newfilename.txt

### \*\*will match usa and copy line 3 and 4 in a new file newfilename.txt

### Printing using OR

$awk ‘/clerk|peon/’ employee.txt

**New Example data awk\_file**

$ cat  awk\_file  
Name,Marks,Max Marks  
Ram,200,1000  
Shyam,500,1000  
Ghyansham,1000  
Abharam,800,1000  
Hari,600,1000  
Ram,400,100

**Example:3 Print the lines which matches the pattern**

I want to print the lines which contains the word “Hari & Ram”

linuxtechi@mail:~$ awk ‘/Hari|Ram/’ awk\_file  
Ram,200,1000  
Hari,600,1000  
Ram,400,1000

**Example:4 How do we find unique values in the first column of name**

linuxtechi@mail:~$ awk -F, ‘{a[$1];}END{for (i in a)print i;}’ awk\_file  
Abharam  
Hari  
Name  
Ghyansham  
Ram  
Shyam

**Example:5  How to find the sum of data entry in a particular column .**

Synatx :  awk -F, ‘$1==”Item1″{x+=$2;}END{print x}’ awk\_file

linuxtechi@mail:~$ awk -F, ‘$1==”Ram”{x+=$2;}END{print x}’ awk\_file  
600

\*\*add all column 2 value if first couumn is Ram, and print the value.

**Example:6  How to find the  total of all numbers in a column**.

For eg we take the 2nd and the 3rd column.

linuxtechi@mail:~$ awk -F”,” ‘{x+=$2}END{print x}’ awk\_file  
3500  
linuxtechi@mail:~$ awk -F”,” ‘{x+=$3}END{print x}’ awk\_file  
5000

**Example:8 How to find the sum of all entries in second column  and append it to the end of the file.**

linuxtechi@mail:~$ awk -F”,” ‘{x+=$2;y+=$3;print}END{print “Total,”x,y}’ awk\_file  
Name,Marks,Max Marks  
Ram,200,1000  
Shyam,500,1000  
Ghyansham,1000  
Abharam,800,1000  
Hari,600,1000  
Ram,400,1000  
Total,3500 5000

**Example:9 How to find the count of entries against every column based on the first column:**

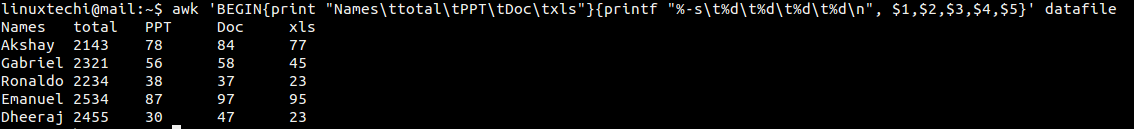
linuxtechi@mail:~$ awk -F, ‘{a[$1]++;}END{for (i in a)print i, a[i];}’ awk\_file  
Abharam 1  
Hari 1  
Name 1  
Ghyansham 1  
Ram 2  
Shyam 1

**Example:10 How to print only the first record of every group:**

linuxtechi@mail:~$ awk -F, ‘!a[$1]++’ awk\_file  
Name,Marks,Max Marks  
Ram,200,1000  
Shyam,500,1000  
Ghyansham,1000  
Abharam,800,1000  
Hari,600,1000

**Example:11  How to populate each column names along with their corresponding data.**

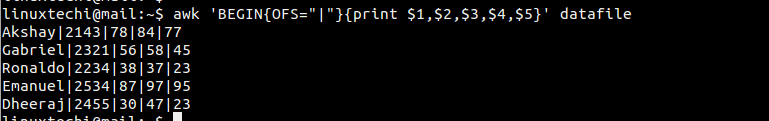
linuxtechi@mail:~$ awk ‘BEGIN{print “Names\ttotal\tPPT\tDoc\txls”}{printf “%-s\t%d\t%d\t%d\t%d\n”, $1,$2,$3,$4,$5}’ datafile



**Example:12 How to change the Field Separator**

As we can see **space**is the **field separator**in the datafile , in the below example we will change field separator  from space to “|”

linuxtechi@mail:~$ awk ‘BEGIN{OFS=”|”}{print $1,$2,$3,$4,$5}’ datafile



**More examples:**

**$ ls -l /dir1 | awk ‘{x+=$5} END {print “total bytes:” x}’**

\*\*this command read details of file in dir1 and add column 5 values (that’s size here) and display it.

**$awk -F, ‘{x+=$2} END {print “total sum:” x}’ filename.txt**

\*\*this command will print sum of column2 from filename.txt

**COUNT**

$awk ‘/sunil/ {count++} END {print count}’ employee.txt

\*\*This will count number of times sunil word is on file employee.txt

Endnotes:

Can read anyfiles and summarize based on what we need.