Assignment No. 9

Objective : To study and execute AWK operations on the text file

Theory :

Awk is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

Built-In Variables In Awk

Awk's built-in variables include the field variables—$1, $2, $3, and so on ($0 is the entire line) — that break a line of text into individual words or pieces called fields.

- **NR:** NR command keeps a current count of the number of input records. Remember that records are usually lines. Awk command performs the pattern/action statements once for each record in a file.
- **NF:** NF command keeps a count of the number of fields within the current input record.
- **FS:** FS command contains the field separator character which is used to divide fields on the input line. The default is "white space", meaning space and tab characters. FS can be reassigned to another character (typically in BEGIN) to change the field separator.
- **RS:** RS command stores the current record separator character. Since, by default, an input line is the input record, the default record separator character is a newline.
- **OFS:** OFS command stores the output field separator, which separates the fields when Awk prints them. The default is a blank space. Whenever print has several parameters separated with commas, it will print the value of OFS in between each parameter.
- **ORS:** ORS command stores the output record separator, which separates the output lines when Awk prints them. The default is a newline character. print automatically outputs the contents of ORS at the end of whatever it is given to print.

AWK Operations

awk allows users to perform various operations on an input file or text. Some of the available operations are:

- Scan a file line by line.
- Split the input line/file into fields.
- Compare the input line or fields with the specified pattern(s).
- Perform various actions on the matched lines.
- Format the output lines.
- Perform arithmetic and string operations.

- Use control flow and loops on output.
- Transform the files and data according to a specified structure.
- Generate formatted reports.

1. Print the lines which match the given pattern.

Syntax: $ awk '/pattern_to_be_matched/ {print}' filename.txt

Example: Print all line containing 'toto' in file called 'tst' :

Code:

cat tst

zroezp toto rueziar

roez tutu uezir

roezp toto reziar

zroezp titi rueiar

roezp toto ruezar

Code:

awk '/toto/{print}' tst

zroezp toto rueziar

roezp toto reziar

roezp toto ruezar

2. Splitting a Line Into Fields

Instead of printing the whole file, you can make awk to print only specific columns of the file. Awk treats all words, separated by white space, in a line as a column record by default. It stores the record in a $N variable. Where $1 represents the first word, $2 stores the second word, $3 the fourth, and so on. $0 stores the whole line so the who line is printed, as explained in example 1.

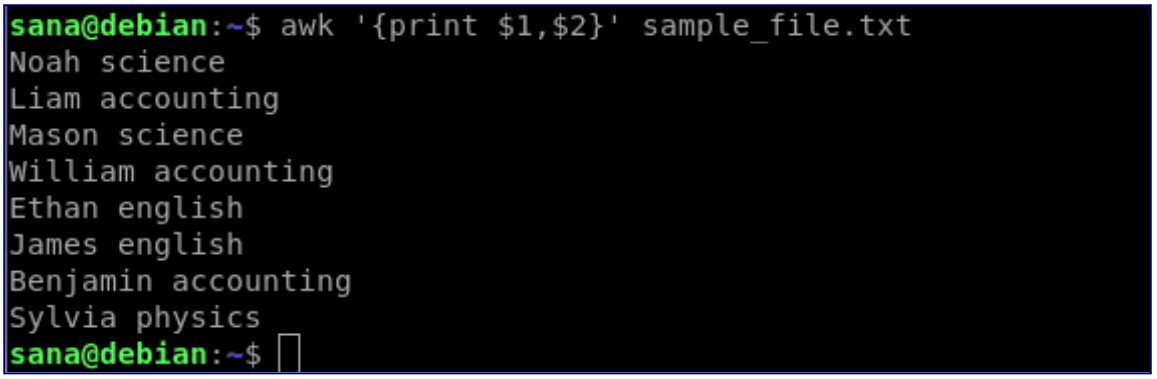
Syntax:

$ awk '{print $N,….}' filename.txt

Example:

The following command will print only the first column(name) and the second column(subject) of sample file:

$ awk '{print $1, $2}' sample_file.txt



3. To find the length of the longest line present in the file

$ awk '{printf "%2d| %s\n",length,$0}' lines.txt
 9| Hi there,
12| How are you?
43| Recently I have to process many text files.
10| I like it.
43| Sometimes files could have very long lines.
37| My task is finding the longest lines.
43| For example, this is a really long... line.
13| I love Linux.
 0|
 3| Bye

$ tr '\t' ' ' <lines.txt | wc -L
This command gives us the actual longest length of the line
$ awk '{ if (length($0) > max) max = length($0) } END { print max }' lines.txt

4. Printing the lines with more than specified characters

$ awk 'length($0) > 10' lines.txt
This command will print lines having more than 10 characters.

Output:
12| How are you?
43| Recently I have to process many text files.
43| Sometimes files could have very long lines.
37| My task is finding the longest lines.
43| For example, this is a really long... line.
13| I love Linux.