Table of Content

Why awk command is used in Unix or Linux

Awk command in Unix / Linux is a powerful command for processing text. Administrator & developer often need to look for the pattern in the files and then process that text as per there requirement. Awk command comes quite handy for these types of task.

- It is very helpful for processing table i.e (rows and column) type of data in the files.
- awk is quite powerful scripting tools and we can also do arithmetic operations on it.

In this lab, we will learn awk syntax , records, fields, line Separator and it will helpful you understand the overall picture of the awk command.

Task0: Basic Syntax - WarmUp

In this task we will work with file **mark.txt** that contains:

```
1) Amit      Physics    80
2) Rahul     Maths      90
3) Shyam     Biology    87
4) Kedar     English    85
5) Hari      English    89
```

- Write a command to display the content of the file without any modification (similar to command cat mark.txt).

- Print the 3rd and 4th column of mark.txt, th result should be :

```
Physics   80
Maths     90
Biology   87
English   85
English   89
```

- Print the name of students and his mark, where the names of students contain the letter a.

```
Rahul   90
Shyam   87
Kedar    85
Hari     89
```

- Print the columns in reverse order, such that output look like this :

```
1) 80 Physics Amit
2) 90 Maths Rahul
3) 87 Biology Shyam
4) 85 English Kedar
5) 89 English Hari
```

- Counting the number of students that have ‘English’ mark. The output should be saved in the cnt variable and printed by the awk as follow :

“Count = 2”

- Print the lines where the student mark is above 87 .

Task1 : Built-in Variables (20 Point , 5 for each)

- Write program (awk script) that print the name of variables in this pattern

```
ARGV[0] = awk
ARGV[1] = v1
ARGV[2] = v2
ARGV[3] = v3
```

Use **oscar_age_male.csv** file as input for the next sub-tasks in Task1, the file contains this pattern : **Index,Year,Age,Actor Name,Movie Name.**

- Write awk script that print the content of the file in this way:

```
<Actor Name > | <Age> | < Year>
```

In this part you must use **OFS combined with FS.**

- Print the name of actors and their movies that have won an Oscar since 1970.

```
------------------------------
Actor Name:  <Actor Name>
Movie Name:  <Move Name>
------------------------------
Actor Name:  <Actor Name>
Movie Name:  <Move Name>
```

Example :

```
---------------------------
Actor Name:  Eddie Redmayne
Movie Name:  The Theory of Everything
---------------------------
Actor Name:  Daniel Day-Lewis
Movie Name:  Lincoln
```

- Display the Movies names between index 50 to 70.

Task2 : Built-in Functions and More (30 Points)

In this task we will work with **StudentsPerformance.csv** files that include several information about students. You need to check the format and the fields names in the file before starting doing this task.

- Display the average score of math, reading and writing scores for students that have **standard lunch**. (5 points)

- Count the number of students whose parents have ‘bachelor's degree . (5 points)

- Display in the beginning a message of “Success Student List”, then print all success students details (the entire line) . Student is written in the list if he got 80 in each one of the scores. In the end write the number of them. (See format below). **You must use BEGIN and END of awk.** (10 points)

```
========
“Success Student List”
========
<student details >
<student details >
<student details >

The number of students : <number>
```

Task3 : Regular Expression (30 Points, 10 for each)

In this task we will work with a **haiku.csv** file that includes poems from Japanese culture. You need to check the format and the fields names in the file before starting doing this task. In this task you will have to use regular expression.

- Display the poetry that includes the patterns “**spring**” in this format :

  <1st poem> - - - <column two of poem> - - - <3rd column of poetry>

  Example :
  I barely even - - - got any sleep last night and - - - spring break is over

- Count the number of poems that include the word fun or fan in the file.

  The number of poem that have fun or fan is : < number of poems>

- Display all poems that in its first part (first column in poem part) include the world **happy.** Also print “- - -” as separator between poem parts as above.

  Example :

  so happy to have - - - finally handed all my - - - work in for my course