**6. Consider the marks.txt is a file that contains one record per line( comma separate fields) of the student data in the form of studentid, student name, Telugu marks, English marks, Maths Marks, Science marks, Social Marks. Write an awk script to generate result for every students in the form of studentid, studentname, Total Marks and result. Result is PASS if marks is >=30 in TELUGU and English, and if marks>=40 in other subjects. Result is fail otherwise.**

**SOLUTION:**

**STEP1: create marks.txt file**

$cat > marks.txt

1001,name1,99,69,85,56,75

1002,name2,89,69,65,56,55

1003,name3,50,50,50,55,55

1004,name4,69,29,85,56,75

1005,name5,99,69,85,56,11

^d

**STEP2: create marks.awk script file**

$cat > marks.awk

{

total=$3+$4+$5+$6+$7

if($3>=30 && $4>=30 && $5>=40 && $6>=40 && $7>=40)

print $1,$2,total, "Pass";

else

print $1,$2,total, "fail";

}

**STEP3: execute awk program**

$awk -F “,” -f marks.awk marks.txt

16. In this example, the input file bookdetails.txt contains records with fields — item number, Book name, Quantity and Rate per book.

$ cat bookdetails.txt

1 Linux-programming 2 450

2 Advanced-Linux 3 300

3 Computer-Networks 4 400

4 OOAD&UML 3 450

5 Java2 5 200

Now the following Awk script, reads and processes the above bookdetails.txt file, and generates report that displays — rate of each book sold, and total amount for all the books sold.

So far we have seen Awk reads the commands from the command line, but Awk can also read the commands from the file using -f option.

Syntax:

$ awk -f script-filename inputfilename

Now our Awk script for billing calculation for books is given below.

$ cat book-calculation.awk

BEGIN {

total=0;

}

{

itemno=$1;

book=$2;

bookamount=$3\*$4;

total=total+bookamount;

print itemno," ", book,"\t","$"bookamount;

}

END {

print "Total Amount = $"total;

}

In the above script,

* Awk **BEGIN section** initializes the variable total. itemno, total, book, bookamount are userdefined awk variables.
* In the Awk **Action** section, Quantity\*bookprice will be stored in a variable called bookamount. Each bookamount will be added with the total.
* Finally in the Awk **END** section, total variable will have total amount.

Now execute the book-calculation.awk script to generate the report that displays each book rate and total amount as shown below.

$ awk -f book-calculation.awk bookdetails.txt

1 Linux-programming $900

2 Advanced-Linux $900

3 Computer-Networks $1600

4 OOAD&UML $1350

5 Java2 $1000

Total Amount = $5750

25. In this example, create an input file “student-marks.txt” with the following content — Student name, Roll Number, Test1 score, Test2 score and Test3 score.

$ cat student-marks.txt

Jones 2143 78 84 77

Gondrol 2321 56 58 45

RinRao 2122 38 37 65

Edwin 2537 78 67 45

Dayan 2415 30 47 20

Now the following Awk script will calculate and generate the report to show the Average marks of each student, average of Test1, Test2 and Test3 scores.

$cat student.awk

BEGIN {

test1=0;

test2=0;

test3=0;

print "Name\tRollNo\t Average Score";

}

{

total=$3+$4+$5;

test1=test1+$3;

test2=test2+$4;

test3=test3+$5;

print $1"\t"$2"\t",total/3;

}

END{

print "Average of Test1="test1/NR;

print "Average of Test2="test2/NR;

print "Average of Test3="test3/NR;

}

In the above Awk script,

* In the **Awk BEGIN** section all the awk variables are initialized to zero. test1, test2, test3 and total are user-defined awk variables.
* In the **Awk ACTION** section, $3, $4, $5 are Test1, Test2 and Test3 scores respectively. total variable is the addition of 3 test scores for each student. The awk variable test1, test2 and test3 has the total scores of each corresponding test.
* So in the **Awk END** section, dividing each test total by total number of records (i.e student) will give you the average score.  **NR** is an **Awk built-in variable** which gives total number of records in input.

26. Awk Example 3. HTML Report for Student Details

In the above two example, we have seen awk variable which has numbers as its values. This example shows awk script to generate the html report for the students name and their roll number.

$ cat string.awk

BEGIN{

title="AWK";

print "<html>\n<title>"title"</title><body bgcolor=\"#ffffff\">\n<table border=1><th colspan=2 align=centre>Student Details</th>";

}

{

name=$1;

rollno=$2;

print "<tr><td>"name"</td><td>"rollno"</td></tr>";

}

END {

print "</table></body>\n</html>";

}

Use the same student-marks.txt input file that we created in the above example.

$ awk -f string.awk student-marks.txt

<html>

<title>AWK</title><body bgcolor="#ffffff">

<table border=1><th colspan=2 align=centre>Student Details</th>

<tr><td>Jones</td><td>2143</td></tr>

<tr><td>Gondrol</td><td>2321</td></tr>

<tr><td>RinRao</td><td>2122</td></tr>

<tr><td>Edwin</td><td>2537</td></tr>

<tr><td>Dayan</td><td>2415</td></tr>

</table></body>

</html>

We can store the above output, which gives the following html table. In the above script, variable called name and rollno are string variable, because it is used in string context.

|  |  |
| --- | --- |
| **Student Details** | |
| Jones | 2143 |
| Gondrol | 2321 |
| RinRao | 2122 |
| Edwin | 2537 |
| Dayan | 2415 |