Experiment 8

30th April 2019

Aim: To be able to write awk scripts.

---

1. Write a awk script that accepts date argument in the form of mm-dd-yy and displays it in the following format. The script should check the validity of the argument and in the case of error, display a suitable message.

Algorithm:

1.Start
2.Read argument
3.If invalid or null argument , print error
4.Store day, year, month separately
5.Replace numeric month to alphabetical month
6.Print “The date is “ day “Month is “ month “Year is “ year
7.Stop

The code is as follows:

awk_1.awk

```
#!/usr/bin/awk -f


BEGIN{

FS="-"

print "Enter the date:"

getline < "/dev/tty"

flag=1;

if((($3%4!=0) && ($2==2) && ($1>28)) || (($3%4==0) && ($2==2) && ($1>29)) || $2 > 12)

    flag=0;

if(flag==0)

        print"Invalid date."

else

{

        if($2==1)

                temp="January";

        else if($2==2)

                temp="February";
```

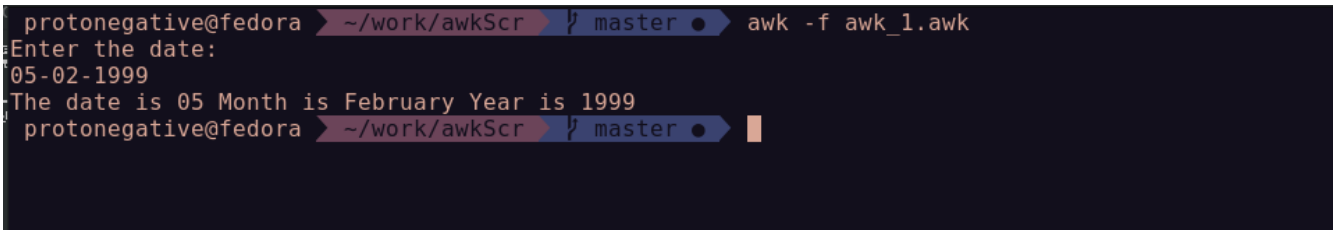
awk_1.awk

```
                temp="February";

        else if($2==3)

                temp="March";

        else if($2==4)

                temp="April";

        else if($2==5)

                temp="May";

        else if($2==6)

                temp="June";

        else if($2==7)

                temp="July";

        else if($2==8)

                temp="August";

        else if($2==9)

                temp="September";

        else if($2==10)

                temp="October";
```

awk_1.awk

```
                temp="October";

        else if($2==11)

                temp="November";

        else

                temp="December";

print "The date is " $1 " Month is " temp " Year is " $3

}

}
```

The output is as follows:

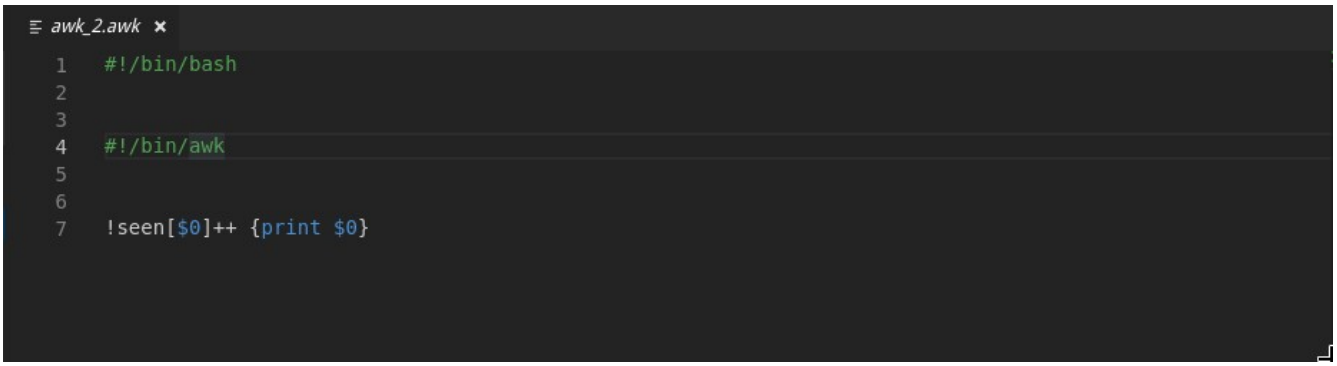


2. Write an awk script to delete duplicate line from a text file. The order of the original lines must remain unchanged

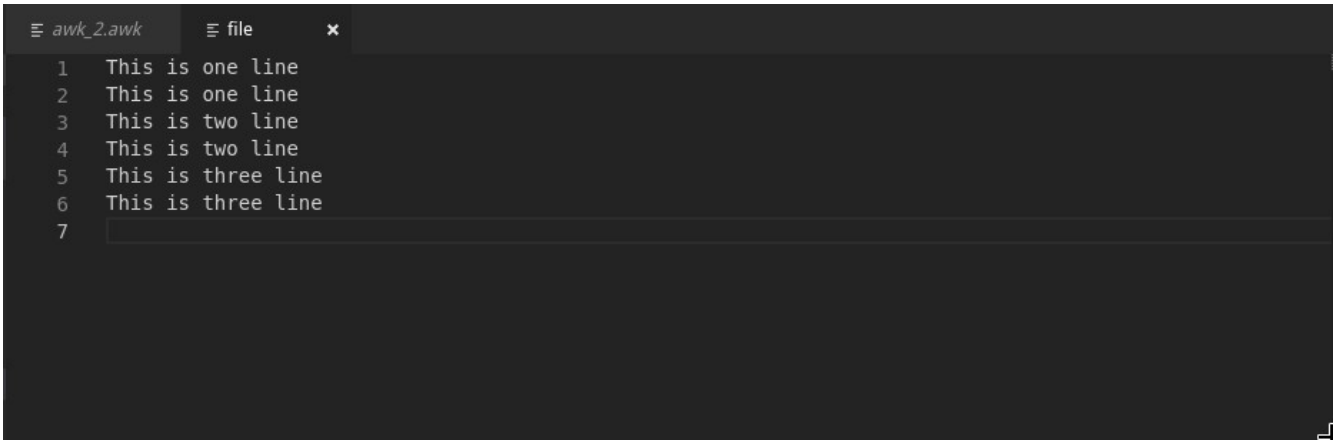
Algorithm:

1.Start
2.Read Argument
3.If argument is invalid or null, print error
4.Print file after deleting duplicate lines
5.Stop

The code is as follows:



The file given as input is:



The output is as follows:

3. Write an awk script to find out total number of books sold in each discipline as well as total book sold based on the given table:

electrical 34

mechanical 67

electrical 80

computers 43

mechanical 65

civil 198

computers 64

Algorithm:

1.Start

2.Read argument

3.If argument is invalid or null, print error

4.Print total number of books sold in each discipline

5.Print total number of books

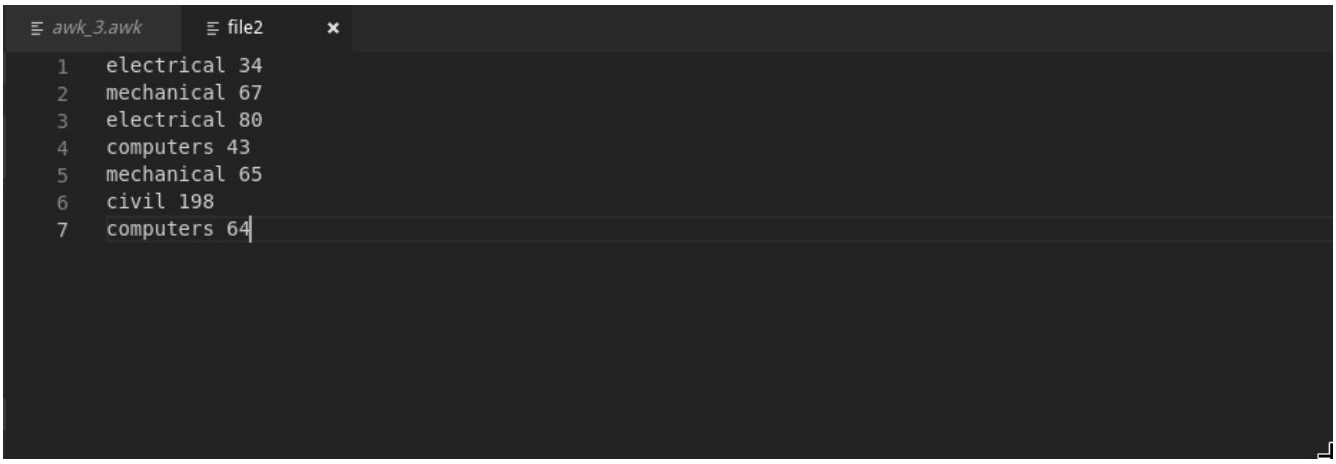
6.Stop

The code is as follows:

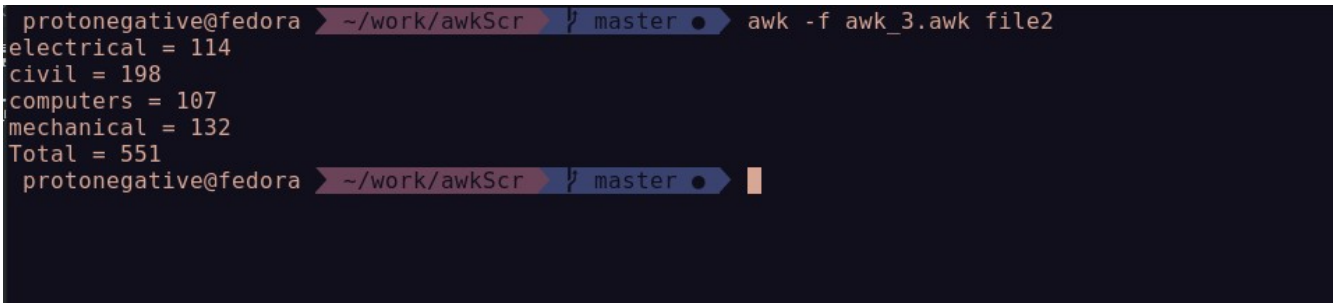
awk_3.awk

```
#!/usr/bin/awk -f


{

  arr[$1]+=$2;

  total+=$2;

}


END{

    for (i in arr){

        print i " = " arr[i];

    }

    print "Total = " total

}
```

The file given as input is:



The output is as follows:



4. Write an awk script to compute gross salary of an employee accordingly to rule given below : If basic salary < 10000 then DA = 45% of the basic and HRA =15% of basic If basic salary >= 10000 then DA =50% of the basic and HRA =20% of basic.

Algorithm:

1.Start

2.Read Salary

3.If (salary < 10000) then

4.DA = 45% of salary

5.HRA = 15% of salary

6.Else

7.DA = 45% of salary

8.HRA = 50% of salary

9.End if

10.Gross salary = DA+HRA+salary

11.Print gross salary

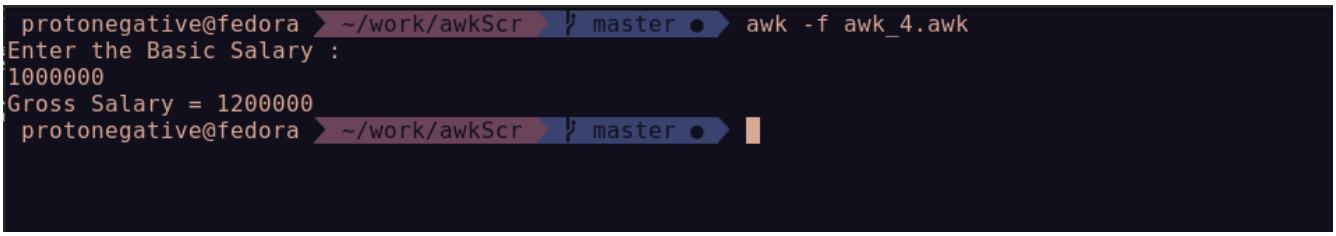
12.Stop

The code is as follows:

awk_4.awk

```
#!/usr/bin/awk -f


BEGIN{

    print "Enter the Basic Salary :";

    getline < "/dev/tty";

    if($0<10000){

        d = 45/100 * 0;

        game = 15/100 * $ 0;

    }


    else{

        d = 50/100 * 0;

        game = 20/100 * $ 0;

    }


    gsal = $ 0 + da + game;

    print "Gross Salary = " gsal

}
```

The output is as follows:



Result: Learned to write awk scripts.

---

eof
