

# Experiment 8

30<sup>th</sup> 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 x

```
1  #!/usr/bin/awk -f
2
3
4  BEGIN{
5
6  FS="-"
7
8  print "Enter the date:"
9
10 getline < "/dev/tty"
11
12 flag=1;
13
14 if(((($3%4!=0) && ($2==2) && ($1>28)) || (($3%4==0) && ($2==2) && ($1>29)) || $2 > 12)
15
16     flag=0;
17
18 if(flag==0)
19
20     print "Invalid date."
21
22 else
23
24 {
25
26     if($2==1)
27
28         temp="January";
29
30     else if($2==2)
31
32         temp="February";
33
```

≡ awk\_1.awk x

```
32         temp="February";
33
34     else if($2==3)
35
36         temp="March";
37
38     else if($2==4)
39
40         temp="April";
41
42     else if($2==5)
43
44         temp="May";
45
46     else if($2==6)
47
48         temp="June";
49
50     else if($2==7)
51
52         temp="July";
53
54     else if($2==8)
55
56         temp="August";
57
58     else if($2==9)
59
60         temp="September";
61
62     else if($2==10)
63
64         temp="October";
```

≡ awk\_1.awk x

```
64         temp="October";
65
66     else if($2==11)
67
68         temp="November";
69
70     else
71
72         temp="December";
73
74     print "The date is " $1 " Month is " temp " Year is " $3
75
76 }
77
78 }
79
80
```

The output is as follows:

```
protonegative@fedora > ~/work/awkScr > master ● awk -f awk_1.awk
Enter the date:
05-02-1999
The date is 05 Month is February Year is 1999
protonegative@fedora > ~/work/awkScr > master ●
```

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:

```
awk_2.awk x
1  #!/bin/bash
2
3
4  #!/bin/awk
5
6
7  !seen[$0]++ {print $0}
```

The file given as input is:

```
awk_2.awk  file  x
1  This is one line
2  This is one line
3  This is two line
4  This is two line
5  This is three line
6  This is three line
7
```

The output is as follows:

```
protonegative@fedora > ~/work/awkScr > master ● awk -f awk_2.awk file
This is one line
This is two line
This is three line
protonegative@fedora > ~/work/awkScr > master ●
```

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 x
1  #!/usr/bin/awk -f
2
3
4  {
5
6      arr[$1]+=$2;
7
8      total+=$2;
9
10 }
11
12
13 END{
14
15     for (i in arr){
16
17         print i " = " arr[i];
18
19     }
20
21     print "Total = " total
22
23 }
```

The file given as input is:

```
awk_3.awk  file2  x
1  electrical 34
2  mechanical 67
3  electrical 80
4  computers 43
5  mechanical 65
6  civil 198
7  computers 64
```

The output is as follows:

```
protonegative@fedora ~/work/awkScr master ● awk -f awk_3.awk file2
electrical = 114
civil = 198
computers = 107
mechanical = 132
Total = 551
protonegative@fedora ~/work/awkScr master ●
```

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 x
1  #!/usr/bin/awk -f
2
3
4  BEGIN{
5
6      print "Enter the Basic Salary :";
7
8      getline < "/dev/tty";
9
10     if($0<10000){
11
12         d = 45/100 * $ 0;
13
14         game = 15/100 * $ 0;
15
16     }
17
18
19     else{
20
21         d = 50/100 * $ 0;
22
23         game = 20/100 * $ 0;
24
25     }
26
27
28     gsal = $ 0 + da + game;
29
30     print "Gross Salary = " gsal
31
32 }
33
34
35
```



The output is as follows:

```
protonegative@fedora > ~/work/awkScr > master ● awk -f awk_4.awk
Enter the Basic Salary :
1000000
Gross Salary = 1200000
protonegative@fedora > ~/work/awkScr > master ●
```

**Result:** Learned to write awk scripts.

---

eof

---