Parth Shroff
S14
Roll No.63

Assignment-8

AIM: To implement AWK in Unix.

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 performs the associated actions.

AWK Operations:

(a) Scans a file line by line
(b) Splits each input line into fields
(c) Compares input line/fields to pattern
(d) Performs actions on matched lines

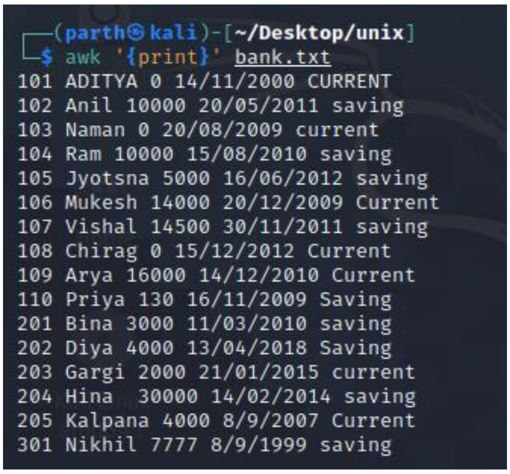
Useful For:

a) Transform data files
(b) Produce formatted reports

Programming Constructs:

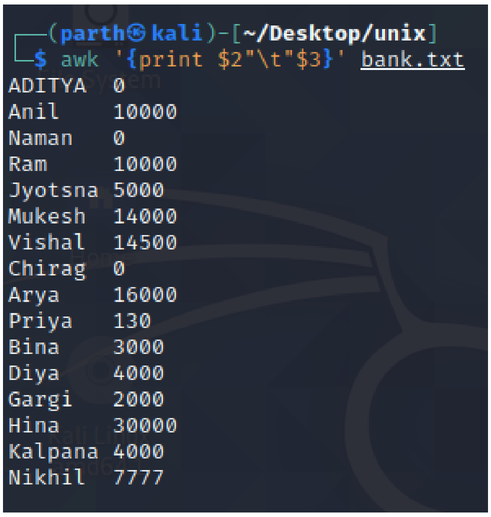
(a) Format output lines
(b) Arithmetic and string operations
(c) Conditionals and loops

1.Display all the records



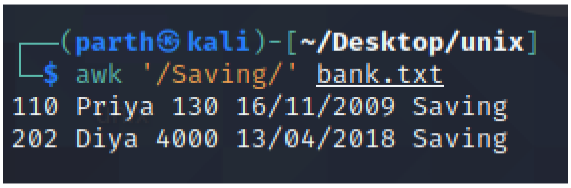
```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '{print}' bank.txt
101 ADITYA 0 14/11/2000 CURRENT
102 Anil 10000 20/05/2011 saving
103 Naman 0 20/08/2009 current
104 Ram 10000 15/08/2010 saving
105 Jyotsna 5000 16/06/2012 saving
106 Mukesh 14000 20/12/2009 Current
107 Vishal 14500 30/11/2011 saving
108 Chirag 0 15/12/2012 Current
109 Arya 16000 14/12/2010 Current
110 Priya 130 16/11/2009 Saving
201 Bina 3000 11/03/2010 saving
202 Diya 4000 13/04/2018 Saving
203 Gargi 2000 21/01/2015 current
204 Hina  30000 14/02/2014 saving
205 Kalpana 4000 8/9/2007 Current
301 Nikhil 7777 8/9/1999 saving
```

2.Display Name and Balance for all records



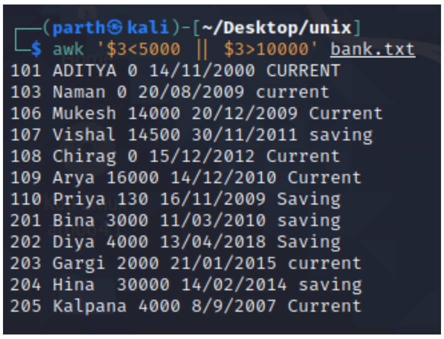
```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '{print $2"\t"$3}' bank.txt
ADITYA  0
Anil    10000
Naman   0
Ram     10000
Jyotsna 5000
Mukesh  14000
Vishal  14500
Chirag  0
Arya    16000
Priya   130
Bina    3000
Diya    4000
Gargi   2000
Hina    30000
Kalpana 4000
Nikhil  7777
```

3.Display only Saving account



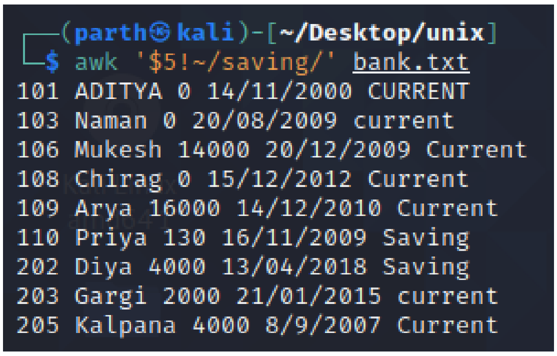
```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '/Saving/' bank.txt
110 Priya 130 16/11/2009 Saving
202 Diya 4000 13/04/2018 Saving
```

4. Print records having balance less than 5000 or more than 10000



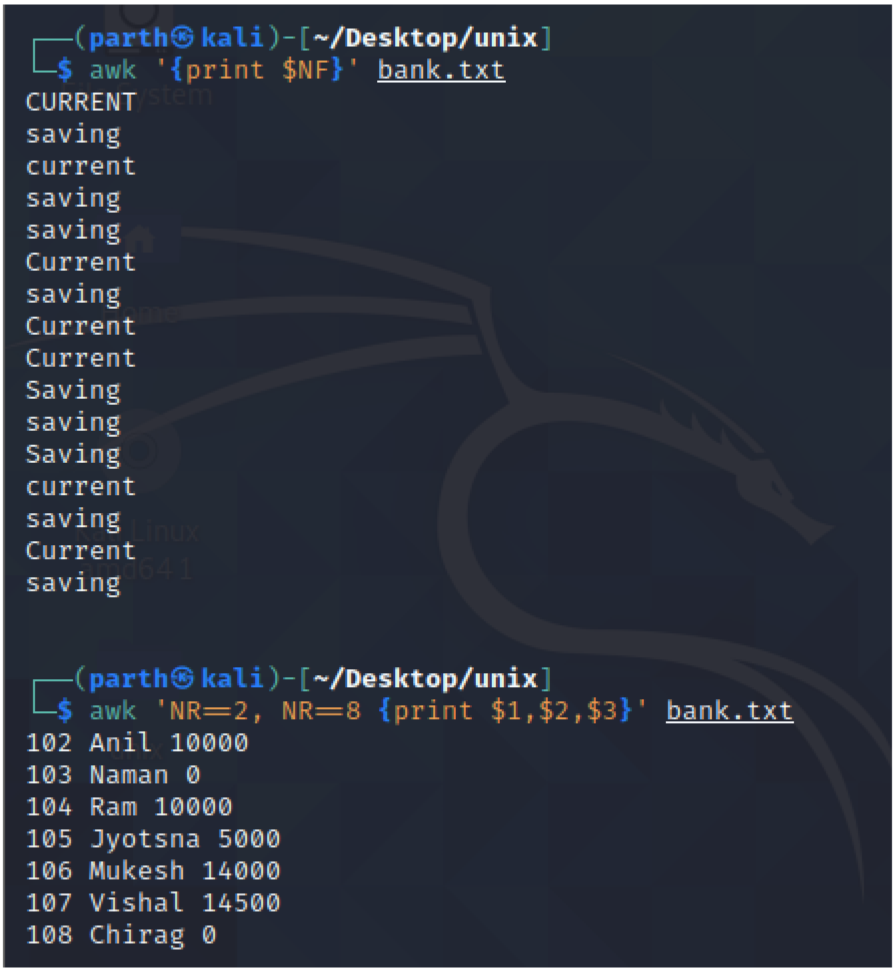
```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '$3<5000 || $3>10000' bank.txt
101 ADITYA 0 14/11/2000 CURRENT
103 Naman 0 20/08/2009 current
106 Mukesh 14000 20/12/2009 Current
107 Vishal 14500 30/11/2011 saving
108 Chirag 0 15/12/2012 Current
109 Arya 16000 14/12/2010 Current
110 Priya 130 16/11/2009 Saving
201 Bina 3000 11/03/2010 saving
202 Diya 4000 13/04/2018 Saving
203 Gargi 2000 21/01/2015 current
204 Hina  30000 14/02/2014 saving
205 Kalpana 4000 8/9/2007 Current
```

5. Print records whose account type is not saving



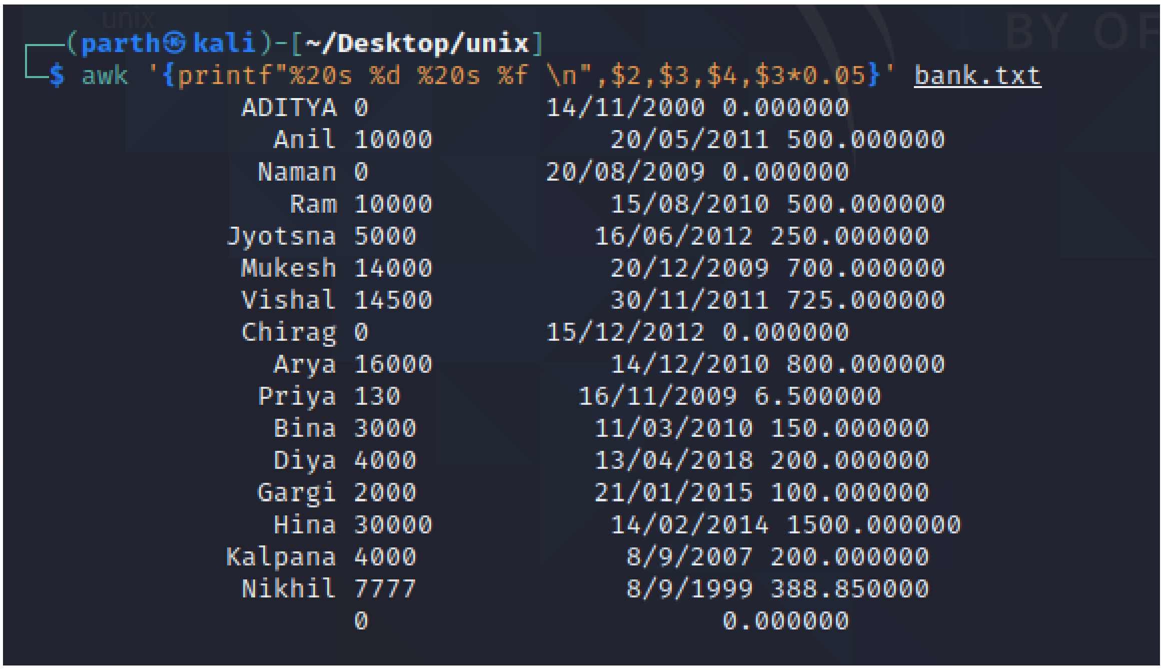
```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '$5!~/saving/' bank.txt
101 ADITYA 0 14/11/2000 CURRENT
103 Naman 0 20/08/2009 current
106 Mukesh 14000 20/12/2009 Current
108 Chirag 0 15/12/2012 Current
109 Arya 16000 14/12/2010 Current
110 Priya 130 16/11/2009 Saving
202 Diya 4000 13/04/2018 Saving
203 Gargi 2000 21/01/2015 current
205 Kalpana 4000 8/9/2007 Current
```

6. Demonstrate use of NF and NR variables.



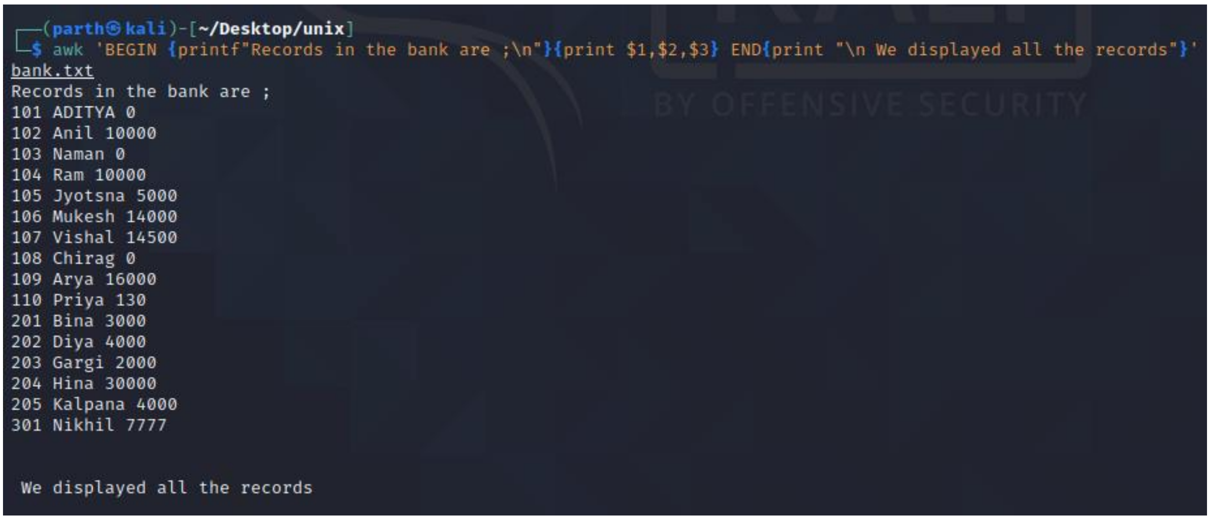
```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '{print $NF}' bank.txt
CURRENT
saving
current
saving
saving
Current
saving
Current
Current
Saving
saving
Saving
current
saving
Current
saving


┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk 'NR==2, NR==8 {print $1,$2,$3}' bank.txt
102 Anil 10000
103 Naman 0
104 Ram 10000
105 Jyotsna 5000
106 Mukesh 14000
107 Vishal 14500
108 Chirag 0
```

7. Calculate and print interest for each record as 0.05 % of balance.



```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk '{printf"%20s %d %20s %f \n",$2,$3,$4,$3*0.05}' bank.txt
              ADITYA 0           14/11/2000 0.000000
                Anil 10000           20/05/2011 500.000000
               Naman 0           20/08/2009 0.000000
                 Ram 10000           15/08/2010 500.000000
             Jyotsna 5000           16/06/2012 250.000000
              Mukesh 14000           20/12/2009 700.000000
              Vishal 14500           30/11/2011 725.000000
              Chirag 0           15/12/2012 0.000000
                Arya 16000           14/12/2010 800.000000
               Priya 130           16/11/2009 6.500000
                Bina 3000           11/03/2010 150.000000
                Diya 4000           13/04/2018 200.000000
               Gargi 2000           21/01/2015 100.000000
                Hina 30000           14/02/2014 1500.000000
             Kalpana 4000             8/9/2007 200.000000
              Nikhil 7777             8/9/1999 388.850000
                     0                      0.000000
```

8. Demonstrate use of END and BEGIN with awk



```
┌──(parth㉿kali)-[~/Desktop/unix]
└─$ awk 'BEGIN {printf"Records in the bank are ;\n"}{print $1,$2,$3} END{print "\n We displayed all the records"}'
bank.txt
Records in the bank are ;
101 ADITYA 0
102 Anil 10000
103 Naman 0
104 Ram 10000
105 Jyotsna 5000
106 Mukesh 14000
107 Vishal 14500
108 Chirag 0
109 Arya 16000
110 Priya 130
201 Bina 3000
202 Diya 4000
203 Gargi 2000
204 Hina 30000
205 Kalpana 4000
301 Nikhil 7777

 We displayed all the records
```

CONCLUSION:

Hence, we understood the concept of AWK in unix.