FOSS Lab Experiment : #11a

Perl Scripting

MUZAMMIL T
Roll No. : 38
S4 CSE

28 March 2020

Contents

Date : 07/02/2020

1 String Pattern Matching

1.1 AIM

To create a text file and answer the following queries :
a) Search for the pattern 'apple' in the file and display the number of occurences.
b) Count the number of words that ends with 'e'
c) Count the number of words that starts with 'ap'
d) Search for words containing 'a' or 's'
e) Search for words containing zero or more occurrence of 'e'
f) Search for words containing one or more occurrence of 'e'
g) Search for words containing the letters 'l' and 'm', with any number of characters in between.

1.2 Perl Script

Perl Script

```
open(FH, "text1.txt") or die "Couldn't open the file.";

my $total_count = 0;
my $e_oc_count = 0;
my $lm_count = 0;
@a_s = ();
@e_1 = ();
@l_m = ();
@all = ();

sub gui
{
        $string = $_[0];
        print("---------------------------\n");
        print("$string\n---------------------------\n\n");
}

while(my $file = <FH>)  {
        foreach my $str (split /\s+/, $file) {
            $total_count++;
            push(@all, $str);
            if($str =~ /^apple$/) {
                $apple_count++;
            }
```

Perl Script Continues ...

```
                if($str =~ /e$/) {
                        $e_count++;
                }
                if($str =~ /^ap/) {
                        $ap_count++;
                }
                if($str =~ /[as]/) {
                        push(@a_s, $str);
                        $as_count++;
                }
                if($str =~ /e/) {
                        push(@e_1, $str);
                        $e_oc_count++;
                }
                if($str =~ /[l]/) {
                            if($str =~ /[m]/) {
                                    push(@l_m, $str);
                                    $lm_count++;
                            }
                }
    }
}


print("Words with 'apple': $apple_count\n");
print("Words ending with 'e': $e_count\n");
print("Words starting with ap: $ap_count\n");

print("\nWords containing a or s: \n");
$string = join("\t", @a_s);
gui($string);

print("Words containing 0 or more occurences of e: \n");
$string = join("\t", @all);
gui($string);

print("Words containing 1 or more occurences of e: \n");
$string = join("\t", @e_1);
gui($string);

print("Words containing l and m: \n");
$string = join("\t", @l_m);
gui($string);

close;
```

1.3 Sample Output

```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ cat text1.txt
apple
grapes
apples
mental
mango
apple
malnutritions

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ perl perl_1.pl
Words with 'apple': 2
Words ending with 'e': 2
Words starting with ap: 3

Words containing a or s:
--------------------------
apple   grapes  apples  mental  mango   apple   malnutritions
--------------------------

Words containing 0 or more occurences of e:
--------------------------
apple   grapes  apples  mental  mango   apple   malnutritions
--------------------------

Words containing 1 or more occurences of e:
--------------------------
apple   grapes  apples  mental  apple
--------------------------

Words containing l and m:
--------------------------
mental  malnutritions
--------------------------
```

2 Conclusion

The Perl script for performing required problems was made,familiarised with the scripting and observed the ouput.

FOSS Lab Experiment : #11b

awk Scripting

28 March 2020

Contents

Date : 07/02/2020

1 Date Finder

1.1 AIM

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..

1.2 awk Script

awk Script

```
BEGIN{
        FS="-"
        print "Enter Date in dd-mm-yyyy format: "
        getline < "/dev/tty"

#           checking for validity of day and months
#           In february check for leap year case (28 & 29 days)
#           Also Check for month with 31 days and 30 days
#           finally Check for month validity

        if((($3%4!=0) && ($2==2) && ($1>28)) ||
           (($3%4==0) && ($2==2) && ($1>29)) ||
           ((($2==1)||($2==3)||($2==5)||($2==7)||
                ($2==8)||($2==10)||($2==12)) && ($1>31)) ||
           ((($2==4)||($2==6)||($2==9)||($2==11)) && ($1>30)) ||
           $2 > 12)

            print"Date is Invalid\n"

        else
        {
#               mapping month with month number
                split("January February March April May June July August
                 ↪  September October November December", month, " ")
                print "Day: " $1
                print "Month: " month[$2]
                print "Year: " $3
        }

}
```

1.3 Sample Output

```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_1.awk
Enter Date in dd-mm-yyyy format:
29-2-2020
Day: 29
Month: February
Year: 2020
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_1.awk
Enter Date in dd-mm-yyyy format:
29-2-2019
Date is Invalid

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_1.awk
Enter Date in dd-mm-yyyy format:
31-6-2020
Date is Invalid

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_1.awk
Enter Date in dd-mm-yyyy format:
30-6-2020
Day: 30
Month: June
Year: 2020
```

2 Removing Duplication

2.1 AIM

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

2.2 awk Script

awk Script

```
1 !exist[$0]++ {print $0}
```

2.3 Sample Output

```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ cat text1.txt
apple
grapes
apple
mango
mango
apple


muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_2.awk text1.txt
apple
grapes
mango
```

3 Books Sold

3.1 AIM

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
electronics 198
computers 64

3.2 awk Script

awk Script

```
{
        book[$1]+=$2;
#       counting individual book count
        total_books+=$2;
#       counting total no. of books
}

END {
    for (i in book){
        print i "\t=\t" book[i];
#       printing individual book count
    }
        print "----------------------------"
#       printing total book count
    print "Total books \t=\t" total_books
}
```

3.3 Sample Output

```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ cat file.txt
electrical 34
mechanical 67
electrical 80
computers 43
mechanical 65
electronics 198
computers 64
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_3.awk file.txt
electronics     =       198
electrical      =       114
computers       =       107
mechanical      =       132
--------------------------
Total books     =       551
```

4 Gross Salary

4.1 AIM

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.

4.2 awk Script

awk Script

```
BEGIN{
    print "Enter the Basic Salary: ";
    getline < "/dev/tty";

    if($0<10000){
        DA = 45/100 * $0;
        HRA = 15/100 * $0;
        print "Your DA is 45% and HRA is 15%"
    }
    else{
        DA = 50/100 * $0;
        HRA = 20/100 * $0;
        print "Your DA is 50% and HRA is 20%"
    }

    gross_sal = $0 + DA + HRA;
#   calculating gross salary = basic + DA + HRA
    print "Gross Salary = " gross_sal
}
```

4.3 Sample Output

```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_4.awk
Enter the Basic Salary:
15000
Your DA is 50% and HRA is 20%
Gross Salary = 25500
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/test$ awk -f awk_4.awk
Enter the Basic Salary:
5000
Your DA is 45% and HRA is 15%
Gross Salary = 8000
```

5 Conclusion

The awk script for performing given problems was made,familiarised with the scripting and observed the ouput.