

Free & Open Source Software Lab Report

Experiments 11 Perl and Awk Scripting

Arun Jose
S4 CSE
Roll No. 12

Computer Science and Engineering
College of Engineering Trivandrum
February 2020

Contents

1 Perl Scripting	2
1.1 Aim	2
1.2 Source Code	2
1.3 Sample	4
1.4 Result	4
2 Awk Scripting - Part 1	5
2.1 Aim	5
2.2 Source Code	5
2.3 Sample	5
2.4 Result	5
3 Awk Scripting - Part 2	6
3.1 Aim	6
3.2 Source Code	6
3.3 Sample	6
3.4 Result	6
4 Awk Scripting - Part 3	7
4.1 Aim	7
4.2 Source Code	7
4.3 Sample	8
4.4 Result	8
5 Awk Scripting - Part 4	9
5.1 Aim	9
5.2 Source Code	9
5.3 Sample	9
5.4 Result	10

1 Perl Scripting

1.1 Aim

Create a text file and answer the following queries :

- Search for the pattern 'apple' in the file and display the number of occurrences.
- Count the number of words that ends with 'e'
- Count the number of words that starts with 'ap'
- Search for words containing 'a' or 's'
- Search for words containing zero or more occurrence of 'e'
- Search for words containing one or more occurrence of 'e'
- Search for words containing the letters 'l' and 'm', with any number of characters in between

1.2 Source Code

```
open my($file), '<', 'text.txt' or die "Error";
$count_e = 0;
$count_ap = 0;
$count_as = 0;

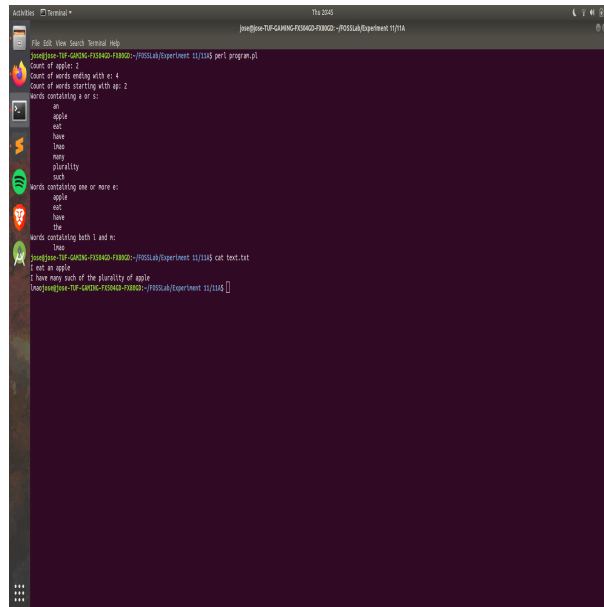
while ( my $line = <$file> )
{
    chomp($line);
    foreach $word (split(' ', $line))
    {
        if($word eq "apple")
        {
            $freq{$word}++;
        }
        if($word =~ /e$/)
        {
            $count_e++;
        }
        if($word =~ /^ap/)
        {
            $count_ap++;
        }
        if((index($word, "a") != -1) or (index($word, "s") != -1))
        {
            $words_as{$word}++;
        }
        if(index($word, "e") != -1)
        {

```

```
        $words_e{$word}++;
    }
    if((index($word, "l") != -1) and (index($word, "m") != -1))
    {
        $words_lm{$word}++;
    }
}

print "Count of apple: $freq{apple}\n";
print "Count of words ending with e: $count_e\n";
print "Count of words starting with ap: $count_ap\n";
print "Words containing a or s:\n";
foreach $word (sort keys %words_as)
{
    print "\t$word\n";
}
print "Words containing one or more e:\n";
foreach $word (sort keys %words_e)
{
    print "\t$word\n";
}
print "Words containing both l and m:\n";
foreach $word (sort keys %words_lm)
{
    print "\t$word\n";
}
```

1.3 Sample



```

jose@jose-TUF-GAMING-PC:~/Documents/Projects/Experiment 11/1104$ perl program.pl
Count of apple: 2
Count of words ending with e: 4
Count of words starting with ap: 2
Words containing a or s:
apple
eat
have
love
many
plurality
such
Words containing one or more e:
apple
eat
have
the
love
Words containing both l and n:
love
jose@jose-TUF-GAMING-PC:~/Documents/Projects/Experiment 11/1104$ cat text.txt
I eat an apple
I have many such of the plurality of apple
jose@jose-TUF-GAMING-PC:~/Documents/Projects/Experiment 11/1104$
```

1.4 Result

The script for simple word processing was made and the output was verified. The script was run on Ubuntu 18.04.3 LTS.

2 Awk Scripting - Part 1

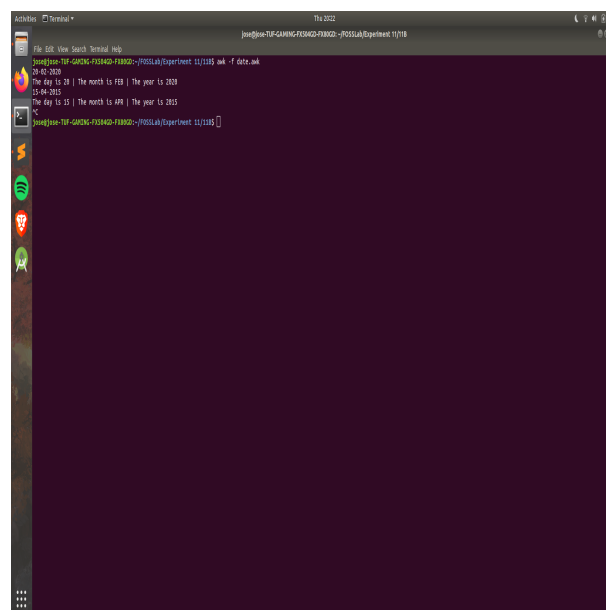
2.1 Aim

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.

2.2 Source Code

```
{
    split("JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC", month, " ")
    date=$1
    split(date, dat, "-")
    dat[2] += 0;
    print "The day is " dat[1] " | The month is " month[dat[2]] " |
        The year is " dat[3]
}
```

2.3 Sample



2.4 Result

The script for Awk processing was made and the output was verified. The shell script was run on Ubuntu 18.04.3 LTS.

3 Awk Scripting - Part 2

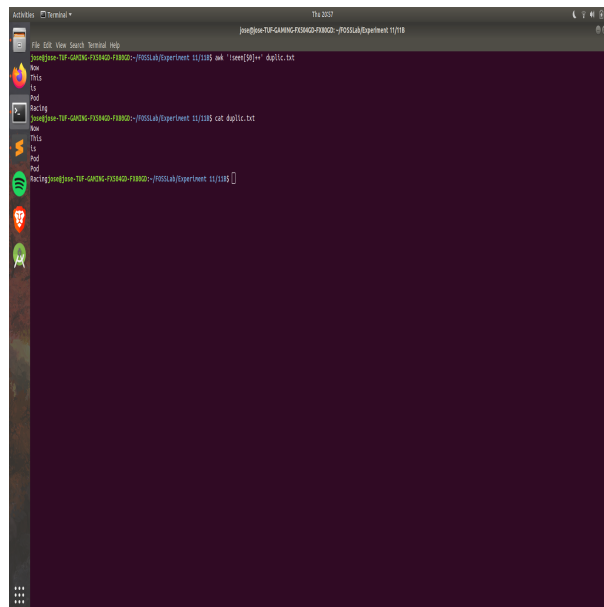
3.1 Aim

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

3.2 Source Code

```
awk '!seen[$0]++' dupes.txt
```

3.3 Sample



3.4 Result

The script for Awk processing was made and the output was verified. The shell script was run on Ubuntu 18.04.3 LTS.

4 Awk Scripting - Part 3

4.1 Aim

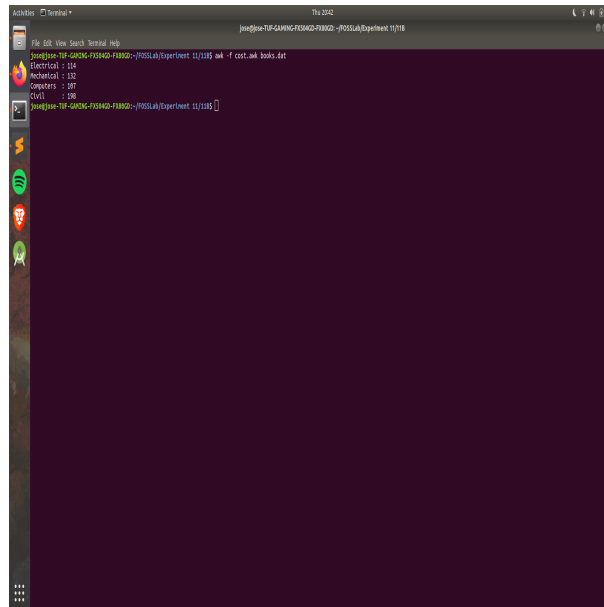
Write an awk script to find out total number of books sold in each discipline as well as total book sold based on the given table

- Electrical 34
- Mechanical 67
- Electrical 80
- Computers 43
- Mechanical 65
- Civil 198
- Computers 64

4.2 Source Code

```
{
    if (match($1, /^electrical$/)) {
        eee += $2;
    } else if (match($1, /^mechanical$/)) {
        mec += $2;
    } else if (match($1, /^computers$/)) {
        cse += $2;
    } else if (match($1, /^civil$/)) {
        civ += $2;
    }
}
END {
    print "Electrical : " eee
    print "Mechanical : " mec
    print "Computers   : " cse
    print "Civil       : " civ
}
```


4.3 Sample



4.4 Result

The script for Awk processing was made and the output was verified. The shell script was run on Ubuntu 18.04.3 LTS.

5.4 Result

The script for Awk processing was made and the output was verified. The shell script was run on Ubuntu 18.04.3 LTS.