Innopolis University

SYSTEM AND NETWORKING ENGINEERING

Essential Skills

LABORATORY REPORT 2

Regular Expression

Student Name	Student ID
Sergey Grebennikov	47611

Lecturer:

Stanislav Litvinov

Submission Date : January 22, 2019

Contents

1 Introduction

A regular expression allows saving a lot of time working with large amounts of text. It searches for patterns in a text. The patterns are created by users. Regular expressions are built-in into many programming tools and programming languages. Therefore, the knowledge of the technique of working with the regular expressions is simply necessary for specialists in the field of information security.

Lab work consists of four assignments:

1. URI

2. Telephone numbers

3. VISA card

4. bash

2 Main Part

2.1 URI

1. Create a **Unix** regular expression to show all lines that contain URI's in a **html**-file (*hrefs . . .*)

   (a) Use **grep –E**

   (b) Use **wget** to retrieve a page

   (c) Look at pages with many URI's, like **www.csszengarden.com**

   Solution:

   Using **wget** I downloaded a page **index.html** from **www.csszengarden.com** and applied the following command at a command prompt:

```
grep -E -o "(href|src|codebase|cite|profile|action|longdesc|usemap|classid|data|
formaction|poster)=\"[^\"]*\"\ " index.html
```

   The command **grep** with attribute *-E* interpret PATTERN (concluded in the quotes) as an extended regular expression.The option *-o* allows to print only the matched parts of a matching line.

2.2 Telephone numbers

1. Create a *regexp* to filter Dutch telephone land numbers correctly

   (a) With and without national code, like *31205257414* or *0205257514*

   (b) With *+* or *00* like *+31205257414* or *0031205257514*

   (c) With various spaces like *+31 20 525 7514* or *00 31 525 75 14* or *(020) 525 75 14*

   Solution:

   Using **wget** I created a file **file** with phone numbers and other strings. Then I applied the following command at a command prompt:

```
grep -E "([\+]?|00)3120[0-9]{7}|020[0-9]{7}|
[+(]+[0-9]{2,3}[)]?([ ]|[0-9]?)*|(([0-9]{2,4}(\ )?)?)" file
```

2.3 VISA card

1. Create a regexp to match VISA card correctly

   (a) Consider the case where we separate the number with "-" or " "

   (b) Only VISA card no other card numbers

   Solution:

   Using **wget** I created a file **visa** with the VISA-card numbers and other strings. Then I applied the following command at a command prompt:

```
grep -E "4[0-9]{3}[\ |\-]?([0-9]{4}[\ |\-]){2}[0-9]{4}" visa
```

2.4 bash

1. Create a regular expression using **grep** that removes all lines containing only comments from a **bash** script file (with your example)

   Solution:

   I had a bash script file **comments.sh** that contains comments:

```
#!/bin/bash
dt=0
t=0
s=0
d=0
e=0
count=0
   #####fdgdfgfdg
        #sddf
#this is comment only
for i in {1000..10000}
do ####dsgdfgdhf
        dt=$(($i/10000))
        t=$(($i%10000/1000)) #;lsdjf;lkj
        s=$(($i%1000/100))
        d=$(($i%100/10))
        e=$(($i%10))
        sum=$(($dt + $t + $s + $d + $e))
        sum_1=$(($sum/10))
        sum_2=$(($sum%10))
        sum=$(($sum_1 + $sum_2))
        if [[ $sum -eq 7 ]] ;then
                count=$(($count + 1))
        fi
done #jsf
echo $count
#kj;[asdf]
```

   Then I created a regular expression using **grep** that removes all lines containing only comments from it:

```
grep -Ev "^[ ]*#" comments.sh
```

   This command deletes all lines containing only comments:

```
dt=0
t=0
s=0
d=0
e=0
count=0
for i in {1000..10000}
do ####dsgdfgdhf
        dt=$(($i/10000))
        t=$(($i%10000/1000)) #;lsdjf;lkj
        s=$(($i%1000/100))
        d=$(($i%100/10))
        e=$(($i%10))
        sum=$(($dt + $t + $s + $d + $e))
        sum_1=$(($sum/10))
        sum_2=$(($sum%10))
        sum=$(($sum_1 + $sum_2))
        if [[ $sum -eq 7 ]] ;then
                count=$(($count + 1))
        fi
done #jsf
echo $count
```

3 Conclusion

The **Regular Expression** is a powerful technology of searching engines. Knowledge about it and how it works is useful for the administrators of information systems so that simplifies the life of them. Also, it saves the time while working with a big amount of data. This is really necessary for my further work and study.

4 References

[1] RegExpr: Learn, Build and Test http://regexr.com/.

[2] Regular Expression 101 https://regex101.com/.

[3] Practice regular expression http://www.ccl.net/cgi-bin/ccl/regexp/test_re.pl

[4] Tutorial regular expression http://gnosis.cx/publish/programming/regular_expressions.html

5 Appendices

5.1 URI

```
grep -E -o "(href|src|codebase|cite|profile|action|
longdesc|usemap|classid|data|formaction|poster)=\"[^\"]*\"\ " index.html
```

5.2 Telephone numbers

```
grep -E "([\+]?|00)3120[0-9]{7}|020[0-9]{7}|[+(]+[0-9]
{2,3}[)]?([ ]|[0-9]?)*|(([0-9]{2,4}(\ )?)?)" phone
```

5.3 VISA card

```
grep -E "4[0-9]{3}[\ |\-]?([0-9]{4}[\ |\-]){2}[0-9]{4}" visa
```

5.4 bash

```
grep -Ev "^[ ]*#" comments.sh
```