LABORATORY 3

3 COMMANDS FOR THE NEXT TEST

Grep =

Sed = find and replace

Awk = processing text file

We are using regular expression. We’ll try to define a rule for the expression we are searching for ( the one we want to operate with ).

Grep -E “ana” file.txt

To search for the word ana in file.txt and it will show us all lines with ana,

ana highlighted.

This is case sensitive!!!

Grep -i to be case insensitive

Regular expression:

MEMORY THE TEACHING NOTES TABLE OF REGULAR EXPRESSIONS!!!!!!!!!!!!!!!!!

^ = beginning of the line

[abc] = a || b || c ( a or b or c )

abc – matches lines that contain exactly abc

[a-z] all letters; [0-9] all digits

$ end of the line

+ - at least one of the previous mentioned char: [0-9]+$ => finishes with a digit.

\*- previous expressions 0 or more times

Sed – to find and replace

Sed has multiple operation types

s/ to\_replace/replace\_with /= mode substitute

if we want to replace every occurance s/to\_replace/replace\_w/g” file.txt

sed -E = to use regular expression

sed -E “s/a/@/gi” file.txt ; gi = global for g and I for case insensitive

Y SED

sed “y/aeiou/@3\!Ow/” this replaces a with @, e with 3, i with ! and so on

“y/ / /” …

If we want a given character that has a regular expressions we need to use \ before it.

For + for example. Or for !. and so on…

D SED

Sed -E “/[0-9]+/d” file.txt = deletes all lines that contain a digit

Sed -E “s/([aeiou][0-9]/\2\1/g” file.txt

This swaps every vowel followed by a digit with that digit

AWK

Format: awk ‘{}’ file.txt – it will execute the body

Body = ‘{}’; we can have multiple bodies

Print – command

NR = number of line

Awk ‘{print NR}’ file.txt => output:

1

2

3

…

NF = number of words ( because the default separator is space, but we can set other separators)

$2 = the 2nd word (field)

$1 = the 1st word ( field )

$0 = the whole line

$NF = the last field

‘NF%2==0{}’ if the condition before the body is true, the body will be executed

The 3rd field to be a number

Awk ‘$3~/[0-9]+/{}’ file.txt

Awk -F: = “:” is the new separator

For executing awk scripts in separate files

//Awk -f scriptfile file.awk

Micro file.awk to edit the body

Then we use

Awk -f awkfile.awk file.txt

WRITE IN A .AWK FILE:

BEGIN {  
 print …

}

{

Print …

}

END {

Print…

}

--

Begin is executed before we start the file

The one in the middle – for every line

End – at the end of the file

At the test:

You will type the command in a file and that will be the one that is graded. We will go through it together.