**SEMINAR 1**

TODO:

-enter University's server

1. **Commands** (options+arguments):

*ls*: directories

-l: details for each entry

-l: dir1: content of dir1

*man ls*: manual for options (?files that start with dot, A: configuration files: contain the config/structure of a system)

*cat /etc/passwd*

*cut -d: -f* 1,5 /etc/passwd (numbers are for fields, to display only the content we need)

PROGRAMS:

*grep*:

eg. Dan /etc/passwd (lines that contain Dan)

adapted: "\<Dan\>" -beginning and end

"^$", "^.$"-one character

".\*$"-empty and non-empty strings

-i: ignore case

-v: invert

-o: prints only the matching part, not the whole line

-E: extended regular expression

-c: count

- E "^[^0-9:]+:" (: where we need to stop)

\*regular expressions:

r+ cant work with an empty list but r\* can

-E -v "." all empty lines

ODD NB OF CHARACTERS:

-E -v "^(..)\*.$" (for even nb we delete the last dot)

USERNAMES with at least two vowels:

-E "^[^:]\*([aeiou])+[^:]\*){2,}:" (any number of characters in the beginning and the end)

vim: create a file

EXTRACTING THE EMAIL:

grep -E -i "\<[^@\*!? ]+@[a-z0-9\_-]+(\.[a-z09\_-]+)+/>" email.txt (+ one or more times)

sed: (search and replace mostly)

-E "s/dan/alice/gi" (if u need / somewhere wlse use #,

g=global ocurrence [otherwise only the first occurence

is changed], i=case insensitive)

| =the output from the command is used as input for the next one

(?is i at the end equal to -i: A: NO)

(?whats -E, A: extended expression)

-E "s/([a-z]/alice/gi"

THE CHANGES ARE MADE ONLY TO THE OUPUT, NOT TO THE FILE

\*to save: > new\_email\_file.txt

-E "y/aeiou/0123456789/" (a is changed to 0.

?but when the list ends, does it backtrack to the beginning, A:No, they need to be of the same lenght)

-E "/dan/d" /etc/passwd | grep dan not showing the line that match the pattern

awk: treats the text as a table (usually tokens are separated by space)

BEGIN {eg. s=0} be4 we process the file: initialize variables (not necessary)

s+=....................

END {print(s)} after whe finish

$0,$1-entire line and specific line

NF AND NR-CURRENT FIELD OR ROW

(?Shortcut for going back)

-F: 'NR % 2==0 {print$1}' (//-pattern, printing odd lines)

(?What is -F, A: field)

-F: '$3 %2 ==0 ...............

-F: '$NF (ending field referral)...........

BEGIN{sum=0}

{sum+=$3}

END{print(sum)}

awk -F:

!!!LOOK UP TEACHING NOTES