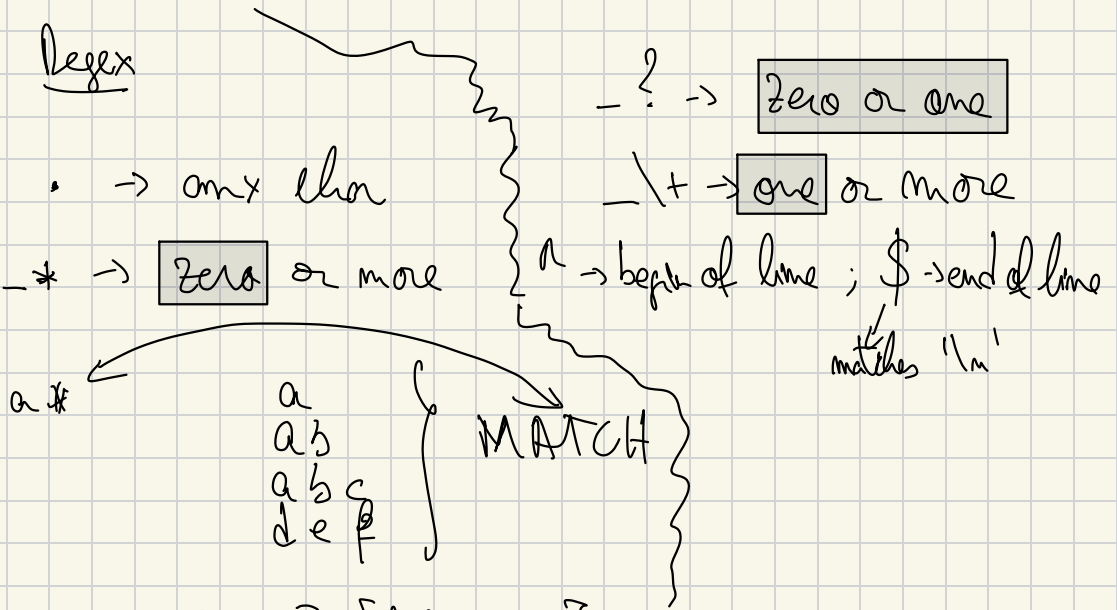
cs.ubbcluj.ro/~rares 27.02.2024

Cmd

bash #$ _command_[args][file1][file2]

[ ] -> optional argument



- grep [-opts] [file1 ....]
  - ↳ filtering the lines and interpreting the regex.
- grep '.' file.txt
- grep 'a' file.txt -> all lines with at least one 'a' will be printed

- grep 'a.*' file.txt
- grep 'a..*' file.txt

a	✓	✓	✓	✓
ab	✓	✓	✓	✓
abc	✓	✓	✓	✓
bcd	✓			
123	✓			
bad	✓	✓	✓	✓

\. → negating the special meaning

grep 'a+' file.txt

grep 'a*' file.txt

1.txt

abc def

abc 123

abc abc 123

123 123 abc

Display all lines with numbers

grep '^.$' 1.txt → displays all lines containing only one character.

grep '^a$' 1.txt → contains only 'a' on the line

grep '^a*$' 1.txt → lines which contain only 'a' or do not contain anything.

grep '^a+$' 1.txt → lines which contain only 'a'

grep '^$' 1.txt → all empty lines.

---

grep -c '^$' 1.txt → will print the number of empty lines in the file.

(-c ↙ count)

grep -c '^.+$' 1.txt → counts all non-empty lines

grep -cv '^$' 1.txt

(-v ↙ invert)

all lines which do not match the regex.

- \< → begin word
- \> → end word.

- (...)
  regex

\(123\)\+

min → a, b ← max.

{a, b}

where a, b ∈ ℕ

- \(123\){1,3}
  ↳ between 1 and 3 times

123
123123
123123123

- \(12\){5}
  ↳ exactly 5 times
  12121212{ ....

- \(12\){,3}
  ↳ maximum of 3 times

  12
  1212
  121212

- \(12\){2,}
  ↳ minimum of 2 times

  1212
  12121 ...

grep -o
↳ only
will print only the part which was matched
-i
↳ case insensitive

grep -c '^a' 1.txt → counts all lines starting with LOWERCASE 'a'.

grep -ci '^a' 1.txt
-ic

grep -civ '^a' 1.txt → counts all lines which
-ivc do not match
-icv
...

grep -e '^a' 1.txt

grep -i -e '^a' -e 'b$' 1.txt
-v
standalone only.

grep -vie '...' ✓

-eiv ✗

-iev ✗

After '-e' we need expresion.

grep -E
↓
interpreted expression in extended mode.

Basic Regex (BRE)

\(, \+ , \{ , \}

Extended Regex (ERE)

(, + , { , }

BRE < ERE < PCRE
(per compatibile)

egrep alias for grep -E.

---

sed 's/ regex1 / regex2 / opts ' [1.txt ...]

s = substitute

control + C → terminate process
control + D → end of input however
control + S → screen lock... !!!!
control + Q → maybe one God

sed 's/ nota=2 / nota=10 /g' 1.txt

g = globally (will replace all occurrences on the line, not only the first one)

```
sed 's/^a/b/' 1.txt
```

```
sed 's/^a.*b.*c$/abc/g' 1.txt
```

Obs: here, 'g' is doing nothing bcs. we already match the whole line with "...$

```
sed 's/\(123\)\{2,3\}/\1 test/g' 1.txt
```

refers to the first GROUP in the regex!

appending.

```
sed -e 's/(123)abc(234).*/\2test\1abc/' 1.txt
```

-e: extended mode

```
234 test 123 abc
```

Obs: for "sed" we can also change the delimiter '/' with anything as long as it's the same everywhere

and not in the regexs.