Session – 5 : Regex
16-08-2019, 08:00 – 09:00 Hrs, RJN 302

Copy the file RollList.csv from the moodle page to try out the following commands. The interpretation of the regular expression is given in plain English to help you gain understanding.

While grep uses a limited capability of pattern matching, egrep uses an extended list of patterns.

[1] Search for a string

```
Lines that contain the string “MM” list all students from Meta.
egrep  'MM' RollList.csv

Lines that contain the string “MM19” will list all first years
from Meta.
egrep 'MM19' RollList.csv

Lines that contain the string “ME19” will list all first years
from Mech.
egrep 'ME19' RollList.csv

Lines that containt the sting “,A” will list all students whose
name starts with an A.
egrep ',A' RollList.csv
```

[2] First use of caret symbol to mark beginning of line

The caret symbol indicates that the string match has to be done at the beginning of the line. BTW, a dollar does the same for the end of the line.

List of first year students from Mech.

```
egrep '^ME19' RollList.csv
```

Since the command egrep picks up only those lines that match the pattern, we can combine the output of egrep command with other commands to further process the output.

```
Count the number of first year students from Mech.
egrep '^ME19' RollList.csv | wc -l

Count the number of second year students from Mech.
egrep '^ME18' RollList.csv | wc -l

Count the number of first year students from Meta.
egrep '^MM19' RollList.csv | wc -l
```

[3] Use of dot to match any single character

The dot symbol indicates that pattern can match at that position with any single character.

List of first year students who are from either Meta or Mech:
```
egrep '^M.19' RollList.csv
```

List of first year students from Meta whose name starts with A.
```
egrep 'MM19B...,A' RollList.csv
```

List of students from Mech whose roll numbers are within first 100.
```
egrep '^ME..B0..' RollList.csv
```

List of students from Mech whose roll numbers are above 100 and below 200.
```
egrep '^ME..B1..' RollList.csv
```

List of all kumars from Mech.
```
egrep '^ME.+Kumar' RollList.csv
```

[4] Use of square brackets to provide options

Each pair of square brackets matches one character. The various options that be used for matching can be given inside the bracket. Ranges can also be given using a dash. Ranges work for three sets namely small alphabets, capital alphabets and numbers.

List of students who have a character “a” in their name, either capital or small. The letter can occur anywhere in the name.
```
egrep '[Aa]' RollList.csv
```

List of students whose name contains a vowel. We use the fact that names come after a comma in this file.
```
egrep '[aAeEiIoOuU]' RollList.csv
```

List of all students who have the characters “h” and “a” - either capital or small occuring in their name side by side. We are counting on the fact that roll numbers will not have this pattern.
```
egrep '[hH][aA]' RollList.csv
```

List of students who have the letter “a” occuring exactly twice and side by side in their name.
```
egrep '[Aa]{2}' RollList.csv
```

List of first year students – from either from Mech or Meta – whose roll numbers end with a digit between 0 and 4. This is one way to divide the class into two halves.
```
egrep '^M.19B..[0-4]' RollList.csv
```

List of students whose names start with letters in the range a to m, either small or capital.

```
egrep ',[a-mA-M]' RollList.csv
```

[5] Use of \b for word boundaries

Special characters with a backslash indicate pattern matching with boundaries. The characters “\b” indicate word boundary.

List of students who have a variant of “jai” as a part of their name.

```
egrep '[jJ]a[iy]' RollList.csv
```

List of student who have a variant of “jai” as one of their names.

```
egrep '\b[jJ]a[iy]\b' RollList.csv
```

List of students who have a variant of “raj” as a part of their name.

```
egrep '[rR]aj' RollList.csv
```

List of students who have a variant of “raj” as one of their names.

```
egrep '\b[rR]aj\b' RollList.csv
```

[6] Second use of caret symbol to negate matching with a character

If the caret symbol occurs within a pair of square brackets, it is to negate matching.

List of students who are not from first year.

```
egrep '1[^9]B' RollList.csv
```

List of students who do not have the letter “p” or “P” in their name. The plus character after the square brackets indicates at least one or more matching. Use the command without the plus and see the difference.

```
egrep '\b[^pP]+\b' RollList.csv
```

Students whose names sound like we are giving respect while pronouncing with their initial. That is, they have a G in the end as their initial. Since the name is the second and last field in each line, we can use the dollar symbol to signal that the matching should be done at the end of the line.

```
egrep '\b[^ ]+\b G$' RollList.csv
```

[7] Use of brackets and pipe for options

While square brackets match exactly one character with options given inside, parantheses and a pipe match strings of different lengths for pattern matching.

List of first year students who are from either Mech or Meta.

```
egrep 'M(M|E)19' RollList.csv
```

List of students from Mech who are from either first or second year and whose name starts with the character A.

```
egrep 'ME1(8|9)B...,A' RollList.csv
```

```
List of students whose name has a raj or a kumar.
egrep '(Raj|Kumar)' RollList.csv
```

One can also given multiple options using the pipe symbol.

List of students who have a variant of “sri” in their name.

```
egrep '(Shri|shri|Sri|Sri|Sree|sree|Shree|shree)' RollList.csv
```

[8] Use of word boundaries:

```
List of all students
egrep '.*,.*\b.*$' RollList.csv
```

Use of caret symbol within square brackets to indicate negation of matching for non blank character. This will have same output as above.

```
egrep '.*,[^ ]+\b.*$' RollList.csv
```

```
List of students with two part names
egrep '.*,[^ ]+ [^ ]+$' RollList.csv
```

List of students with three part names

```
egrep '.*,[^ ]+ [^ ]+ [^ ]+$' RollList.csv
```

List of students with single initial in the end. We use the flower brackets with an integer within to state exactly how many times the pattern given in the preceding pair of square bracket should match. By asking for a non blank character that is occuring exactly once at the end of the string, we achieve this pattern.

```
egrep '.*,[^ ]+ [^ ]+ [^ ]{1}$' RollList.csv
```

List of students with single initial in the middle of their name. We use same logic as above but swap the positions of second and last part of the names.

```
egrep '.*,[^ ]+ [^ ]{1} [^ ]+$' RollList.csv
```

[9] Check for alphabetic or digit or alphanumeric strings from strings

Regular expression matching also has capability to use the following categories of strings – alphabetic, numeric and alphanumeric.

Match all alphabetical strings in the file. All names get matched this way.

```
egrep '[[:alpha:]]' RollList.csv
```

Match all numeric strings in the file. Only part of the roll numbers can be matched this way.

```
egrep '[[:digit:]]' RollList.csv
```

Match all alphanumeric strings in the file. You will see that roll numbers get matched as they are purely alphanumeric.

```
egrep '[[:alnum:]]' RollList.csv
```

[10] Check for email addresses in a text file

Try and interpret the following pattens and see the output to check your understanding. Look at the contents of the file elist given below to try out.

```
egrep '\b[a-zA-Z]+@[a-zA-Z]+\.[a-zA-Z0-9]+\b' elist.txt

egrep '\b[a-zA-Z]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+\b' elist.txt

egrep '\b[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+\b' elist.txt

egrep '\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}\b'
elist.txt

egrep '^\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}\b$'
elist.txt
```

Contents of file elist.txt for your practice:

```
This file contains sometimes with some emails.
bogus@bogus
name@domain.com
gphani@iitm.ac.in
ThisIsMyEmailAddress
email.address
name@domain.com@domain.com
Here is one email: phanikumar@gmail.com
vip@123.com
mm2090@iitm.ac.in
Here is a complex one: gphani-B27_6C.new@gmail.com
```

Homework:

[1] List processes that are being run by root or logged-in user

[2] List files that have write access to others

[3] List used space for only mounted hard disk partitions

[4] Find out which files in the /etc directory use the name of your machine

Tips: Output redirection using 2>~/errorfile.txt will make the errors go to a file called errofile.txt in your home directory.