كلية العلوم
جامعة عبد المالك السعدي
Université Abdelmalek Essaad

Operating systems

Adil ENAANAI

adil.enaanai@gmail.com

Computer Science Department

Faculty of Science -Tetouan-

Chapter 8

Regular expressions under Linux

What is a regular expression?

Definition

A regular expression is a pattern or textual model that describes the way characters are strung together.

Example

aaaaaaabbbbbccccccccc

Suite of 'a' followed by suite of 'b' followed by suite of 'c

monemail12@gmail.com

Letter sequence followed by a number sequence followed by '@' followed by a letter sequence followed by a number sequence followed by '.' followed by a letter sequence

Syntax

Special characters

The following special characters are generally used in regular expression syntax:

Metacharacters

These characters each have a special meaning in regular expressions. They are one of the strengths of REGEX.

Syntax

Special characters

Metacharacters

These charactershaveeach aspecial meaning in regular expressions.

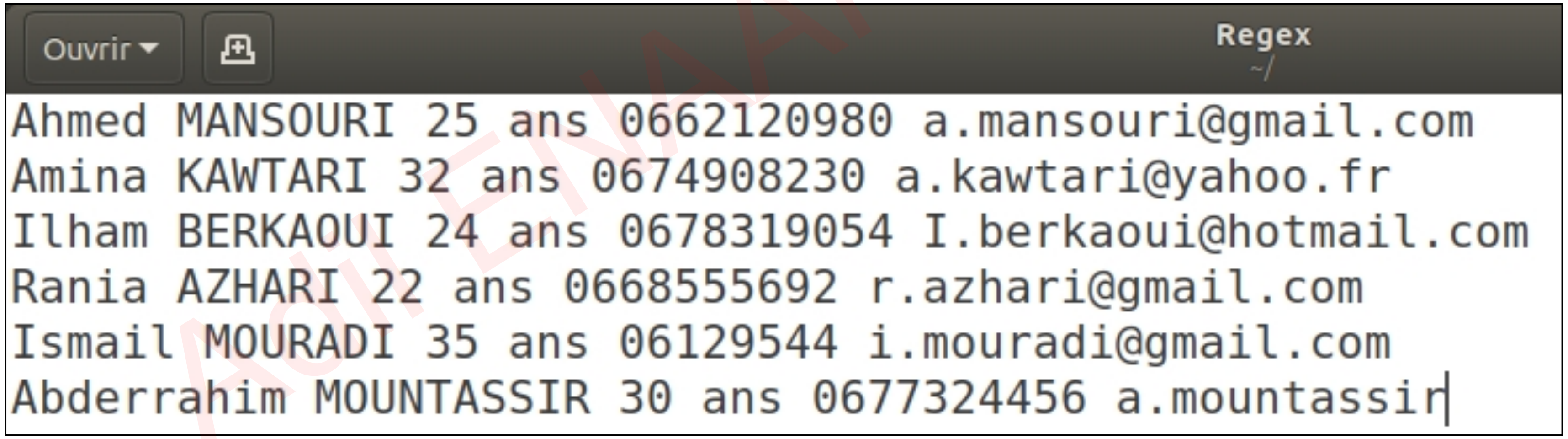
Sign	Meaning	Example
^	marks a chain start	^music' (begins with music)
$	marks the end of the chain	^music$' (begins and ends with music)
|	logic connector or	music | musique' (music or music)
.	all characters except carriage returns (to do this, use the s option)	'.' (applies to a or 8 or $)
\	exhaust character	\?' (meaning that the "?" here counts as a normal character)

Syntax

Special characters

Metacharacters

Example Consider the following work file



```
Ahmed MANSOURI 25 ans 0662120980 a.mansouri@gmail.com
Amina KAWTARI 32 ans 0674908230 a.kawtari@yahoo.fr
Ilham BERKAOUI 24 ans 0678319054 I.berkaoui@hotmail.com
Rania AZHARI 22 ans 0668555692 r.azhari@gmail.com
Ismail MOURADI 35 ans 06129544 i.mouradi@gmail.com
Abderrahim MOUNTASSIR 30 ans 0677324456 a.mountassir
```

All lines beginning with 'A' :

```
grep -E   '^A'   Regex
```

All lines beginning with ' with 'A' or 'I':

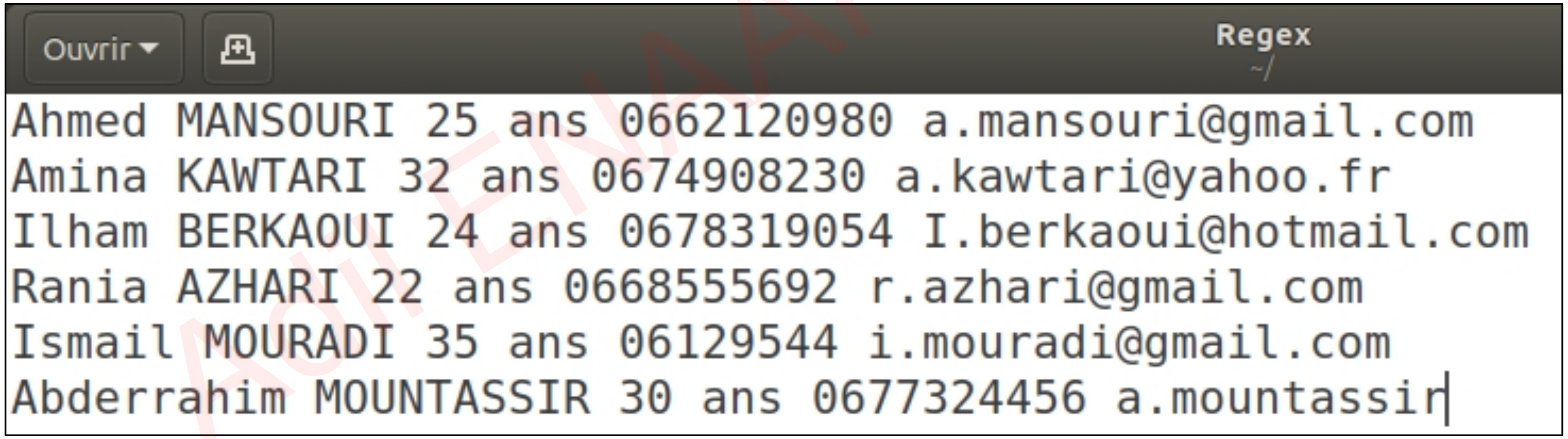
```
grep -E   ^A|^I    Regex
```

Syntax

Special characters

Metacharacters

Example Consider the following work file



```
Ahmed MANSOURI 25 ans 0662120980 a.mansouri@gmail.com
Amina KAWTARI 32 ans 0674908230 a.kawtari@yahoo.fr
Ilham BERKAOUI 24 ans 0678319054 I.berkaoui@hotmail.com
Rania AZHARI 22 ans 0668555692 r.azhari@gmail.com
Ismail MOURADI 35 ans 06129544 i.mouradi@gmail.com
Abderrahim MOUNTASSIR 30 ans 0677324456 a.mountassir
```

All lines ending in 'r':

```
grep -E 'r$' Regex
```

All words ending in 'd':

```
grep -E   'd '   Regex
```

Syntax

Special characters

Quantifiers

Quantifiers are used to specify the number of times a character or sequence of characters may be repeated.

Sign	Meaning	Example
?	0 or 1 time	bue?no' (buno, or bueno)
+	1 or more	bue+no' (bueno, bueno, bueeeeeeno...)
*	0, 1 or more	bue*no' (buno, bueno, bueeeeeeno...)
( )	can be applied repeatedly to several signs	Ay(Ay)*' (Ay, AyAy, AyAyAyAyAyAy...)
{ }	specify number of repetitions	•Ay(Ay){3}' (AyAyAyAy) •Ay(Ay){1-4}' (AyAy, AyAyAy [...] AyAyAyAyAy) •Ay(Ay){3,}' (AyAyAyAy; AyAyAyAyAy; etc)

Syntax

Special characters Classes and intervals

Classes allow you to search between several different characters, giving you alternatives.

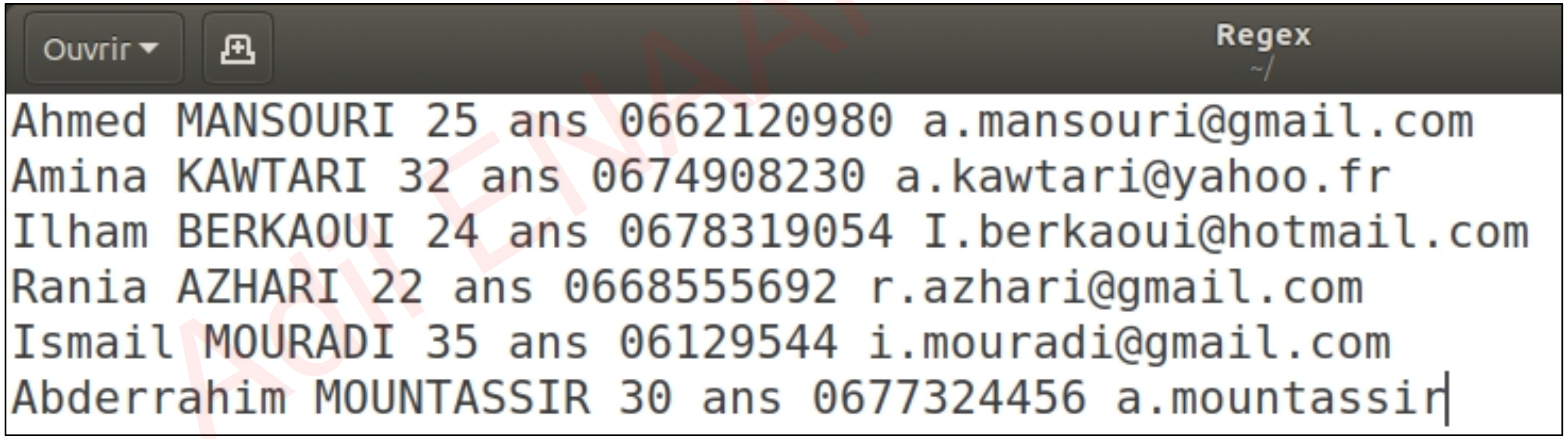
Sign	Meaning	Example
[ ]	character class	gr[oai]s' (big, fat or grey)
[-]	class interval	n°[0-9]' (n°1, n°2, [...] n°9)
[^]	class to exclude	h[^3-9]' (h1 and h2 only)

in a class, the hyphen "-" acts as a delimiter, so if you want to include it as a character, you'll need to place it at the end of the class (or at the beginning). The closing bracket "]" also delimits the end of the class, so you'll need to escape it with a backslash.

Syntax

Special characters

Example Consider the following work file



```
Ahmed MANSOURI 25 ans 0662120980 a.mansouri@gmail.com
Amina KAWTARI 32 ans 0674908230 a.kawtari@yahoo.fr
Ilham BERKAOUI 24 ans 0678319054 I.berkaoui@hotmail.com
Rania AZHARI 22 ans 0668555692 r.azhari@gmail.com
Ismail MOURADI 35 ans 06129544 i.mouradi@gmail.com
Abderrahim MOUNTASSIR 30 ans 0677324456 a.mountassir
```

All phone numbers

grep -E '[0-9]{10}' Customers

All emails

grep -E '[a-zA-Z\._]+[0-9]*@[a-zA-Z]+[0-9]*\.[a-zA-Z\.]+' ClienPt0s8

Syntax

Special characters

Example

All first names

```
grep -E '[A-Z][a-z]+' Customers
```

All names

```
grep -E ' [A-Z]+ ' Customers
```

All ages over 30

```
grep -E '[3-9][0-9] years' Customers
```

Everyone aged 30 and over with a Gmail account

```
grep -E '[3-9][0-9] ans.*gmail.*' Customers
```

*Anyone aged 30 and over **or** with a Gmail account*

```
grep -E '([3-9][0-9] ans| [^ ]*@gmail.*)' Clients
```

Syntax

Special characters

Abbreviated classes

These are shortcuts to the most frequently used long classes

Shortcut	Meaning
\d	Indicates a number. It's exactly the same as typing [0-9].
\D	Indicates what is NOT a number. It's like typing [^0-9]
\w	Indicates an alphanumeric character or an underscore. This corresponds to typing [a-zA-Z0-9_].
\W	Indicates something that is NOT an alphanumeric character or an underscore. Ca is equivalent to typing [^a-zA-Z0-9_].
\t	Indicates tabulation
\n	Indicates a new line
\r	Indicates a carriage return
\s	Indicates white space (corresponds to \t \n \r)
\S	Indicates what is NOT white space.
.	The dot indicates any character! It allows anything!

Syntax

Special characters Named classes

Name of the class	Description
[:alnum:]	alphanumeric characters (equivalent to *[A-Za-z0-9]*)
[:alpha:]	alphabetical characters (*[A-Za-z]*)
[:blank:]	blank characters (space, tab)
[:ctrl:]	control characters (the first of the ASCII code)
[:digit:]	digit (*[0-9]*)
[:graph:]	typeface (which makes a mark on the screen, as it were)
[:print:]	printable character (passes to the printer ... everything except control characters)
[:punct:]	punctuation character
[:space:]	spacing character
[:upper:]	uppercase character
[:xdigit:]	hexadecimal character

Syntax

Special characters Example

All alphabetical characters

```
grep -E '[[:alpha:]]" Clients
```

All numbers

```
grep -E '[[:digit:]]" Customers
```

All non-numeric characters

```
grep -E '[^[:digit:]]" Customer
```

All uppercase characters

```
grep -E '[[:upper:]]" Customers
```

All uppercase characters

```
grep -E '[[:lower:]]" Customers
```

Syntax Exercise

Given the following text, find the first and last names of the people involved

The gentleman's name is **Rayan MIRI**, and he's asleep in the center of the amphitheater.

Meryem CHAKIR prefers the BASH: that's understandable. It's an example that was well understood by **Hicham AMMARI**. But this line does NOT contain a first name and surname.

Syntax Solution

The regular expression is therefore constructed as follows:

a name: **[[:upper:]]{2,}**

a first name: **[[:upper:]][[:lower:]]+**

a first name followed by a last name (separated by a space) :

[[:upper:]][[:lower:]]+ [[:upper:]]{2,}

Now we need to express what can be found before :

(^|[[:space:]])

... and what comes after :

([[:space:][:punct:]]|$)

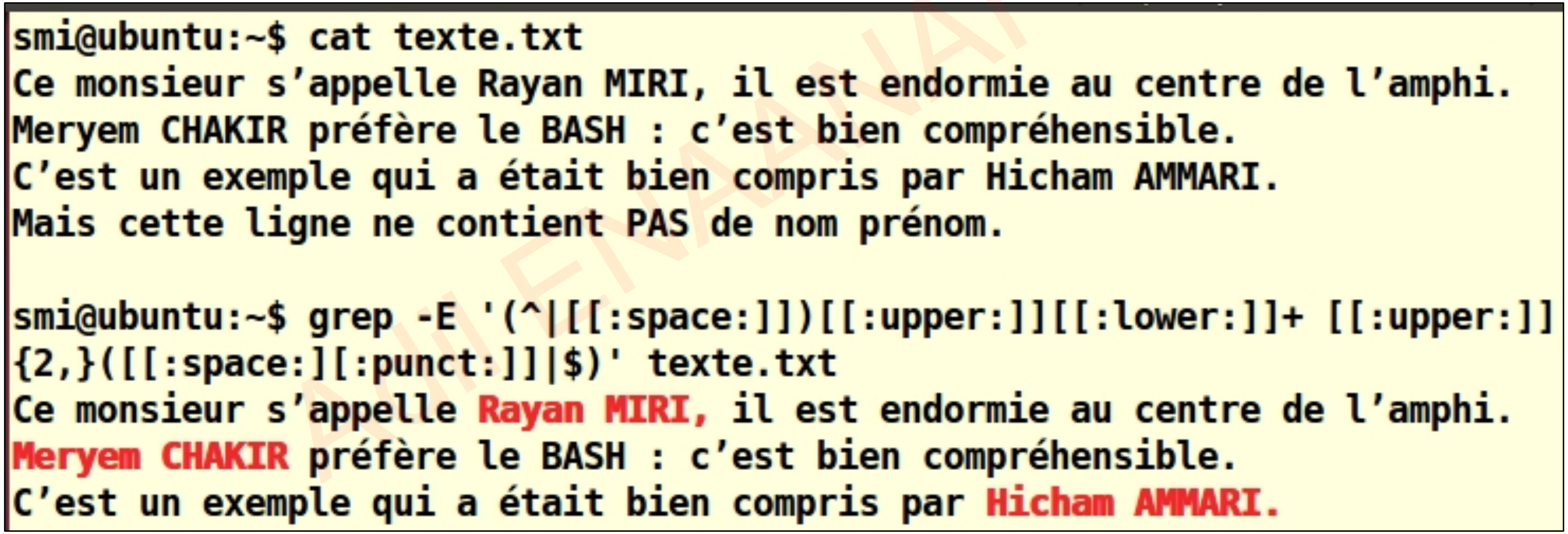
And there you have it: you can put it all together in a single line.

(folded into two lines below for typographical reasons) :

To test

```
grep -E '(^|[[:space:]])[[:upper:]][[:lower:]]+ [[:upper:]]{2,}([[:space:][:punct:]]|$)' text.txt
```

Syntax Solution



```
smi@ubuntu:~$ cat texte.txt
Ce monsieur s’appelle Rayan MIRI, il est endormie au centre de l’amphi.
Meryem CHAKIR préfère le BASH : c’est bien compréhensible.
C’est un exemple qui a était bien compris par Hicham AMMARI.
Mais cette ligne ne contient PAS de nom prénom.

smi@ubuntu:~$ grep -E '(^|[[:space:]])[[:upper:]][[:lower:]]+ [[:upper:]]
{2,}([[:space:][:punct:]]|$)' texte.txt
Ce monsieur s’appelle Rayan MIRI, il est endormie au centre de l’amphi.
Meryem CHAKIR préfère le BASH : c’est bien compréhensible.
C’est un exemple qui a était bien compris par Hicham AMMARI.
```

End of chapter 8