



DEPARTAMENTO
DE SISTEMAS
INFORMÁTICOS



Regular expression, filters, find and other commands

Enrique Arias

Universidad de Castilla–La Mancha

Contents

- Main objectives
- File visualization commands: cat, more, less
- Sort, cut, paste and join commands
- Regular expressions
- Finding files by grep and find

Contents

- Main objectives
- File visualization commands: cat, more, less
- Sort, cut, paste and join commands
- Regular expressions
- Finding files by grep and find

Main objectives

- To know commands cat, more, less
- To know other commands on files: sort, cut, paste and join.
- To be able to express a pattern by using regular expressions
- To Use find and grep to look for files

Contents

- Main objectives
- File visualization commands: cat, more, less
- Sort, cut, paste and join commands
- Regular expressions
- Finding files by grep and find

File visualization commands: cat, more, less

■ cat

- has three related functions with regard to text files: displaying them, combining copies of them and creating new ones.
- `cat [options] [filenames] [-] [filenames]`
- Examples
 - `cat file1`
 - `cat file1 > file2`
 - `cat > file` (File creation. then press ENTER and finish writing pressing Ctrl+D)
 - `cat file1 file2 file3 > file4` (concatenation)

File visualization commands: cat, more, less

■ more and less

- more and less are utilities for reading a very large text file in small sections at a time
- more allows to visualize files screen to screen. With the space key you pass screen to screen. Pressing q the execution is abandoned.
- less is a modern version of more

Contents

- Main objectives
- File visualization commands: cat, more, less
- Sort, cut, paste and join commands
- Regular expressions
- Finding files by grep and find

Sort, cut, paste and join commands

■ sort

- sort sorts the contents of a text file, line by line
- sort [OPTION]... [FILE]...
- Options
 - -c Check for sorted input; do not sort.
 - -m Merge already sorted files; do not sort.
 - -d Consider only blanks and alphanumeric characters.
 - -f Fold lower case to upper case characters.
 - -o FILE Write result to FILE instead of standard output.
 - -r Reverse the result of comparisons.
 - -t SEP Use SEP instead of non-blank to blank transition.
 - -k FIELD Sorts according to FIELD

Sort, cut, paste and join commands

■ sort

□ Examples

- Sort file in reverse order and sends output to file1

```
sort -r file -o file1
```

- Sort file according to field number 2 and considering that field separator is :

```
sort -t: -k2 file
```

Sort, cut, paste and join commands

■ cut

- cutting out the sections from each line of files and writing the result to standard output
- cut OPTION... [FILE]...
- Options
 - -b(byte): To extract the specific bytes, you need to follow -b option with the list of byte numbers separated by comma or a range of numbers separated by hyphen(-)
 - -c (column): To cut by character use the -c option. This can be a list of numbers separated comma or a range of numbers separated by hyphen(-)
 - -f (field): List of the fields number specified must be separated by comma. Ranges are not described with -f option. cut uses tab as a default field delimiter but can also work with other delimiter by using -d option.
 - -d, --delimiter=DELIM: use character DELIM instead of a tab for the field delimiter.

Sort, cut, paste and join commands

■ cut

□ Examples (create a file and experience)

- `cut -b 1,2,3 file.txt`
- `cut -b 1-3 file.txt`
- `cut -b 1- file.txt`
- `cut -b -3 file.txt`
- `cut -c 1,2,3 file.txt`
- `cut -c 1-3 file.txt`
- `cut -f 1 file.txt`
- `cut -d " " -f 1 file.txt`

Sort, cut, paste and join commands

■ paste

- It is used to join files horizontally by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output.
- `paste [OPTION]... [FILES]...`
- Options
 - `-d (delimiter)`: Paste command uses the tab delimiter by default for merging the files. The delimiter can be changed to any other character by using the `-d` option.

Sort, cut, paste and join commands

■ paste

□ Examples

```
Kike@Enriq-Arias-UB:~$ cat Country.txt
```

```
Spain
```

```
Italy
```

```
United Kingdom
```

```
Kike@Enriq-Arias-UB:~$ cat Capital.txt
```

```
Madrid
```

```
Rome
```

```
London
```

```
Kike@Enriq-Arias-UB:~$ cat Population.txt
```

```
5
```

```
3
```

```
8
```



paste

```
Kike@Enriq-Arias-UB:~$ paste Country.txt Capital.txt Population.txt
```

```
Spain Madrid 5
```

```
Italy Rome 3
```

```
United Kingdom London 8
```

```
Kike@Enriq-Arias-UB:~$
```

Sort, cut, paste and join commands

■ join

- It is a command line utility for joining lines of two files on a common field.
- To join two files using the join command files must have identical join fields.
- The default join field is the first field delimited by blanks.
- Join expects that files will be sorted before joining
- join [OPTION]... FILE1 FILE2
- Options
 - -t Char Use CHAR as input and output field separator
 - -n m N represents the number of file (order) and m the field of this file use for joining

Sort, cut, paste and join commands

■ join

□ Examples

```
Kike@Enriq-Arias-UB:~$ cat File1.txt
Spain:Madrid
Italy:Rome
United Kingdom:London
```

```
Kike@Enriq-Arias-UB:~$ cat File2.txt
Madrid: 5
Rome:3
London:8
```



join

```
Kike@Enriq-Arias-UB:~$ join -t: -1 2 -2 1 File1.txt File2.txt
Madrid:Spain: 5
Rome:Italy:3
London:United Kingdom:8
```


Exercise 1

- Create file Data.txt (name:DNI:City)

```
Pepe:81234567:Albacete  
Maria:23456781:Valencia  
Juan:12345678:Madrid  
Ana:34567812:Barcelona
```

- Create file Salary.txt (DNI: salary)

```
34567812:1500  
81234567:1000  
23456781:1200
```

- Join both files according the common field and sends the output to DataP.txt file

Exercise 1

- Use cut on DataP.txt file to show char columns 2, 5 and 6. Also the fields 1 and 3.
- Now use paste to merge Data.txt and Salary.txt files using : as delimiter.

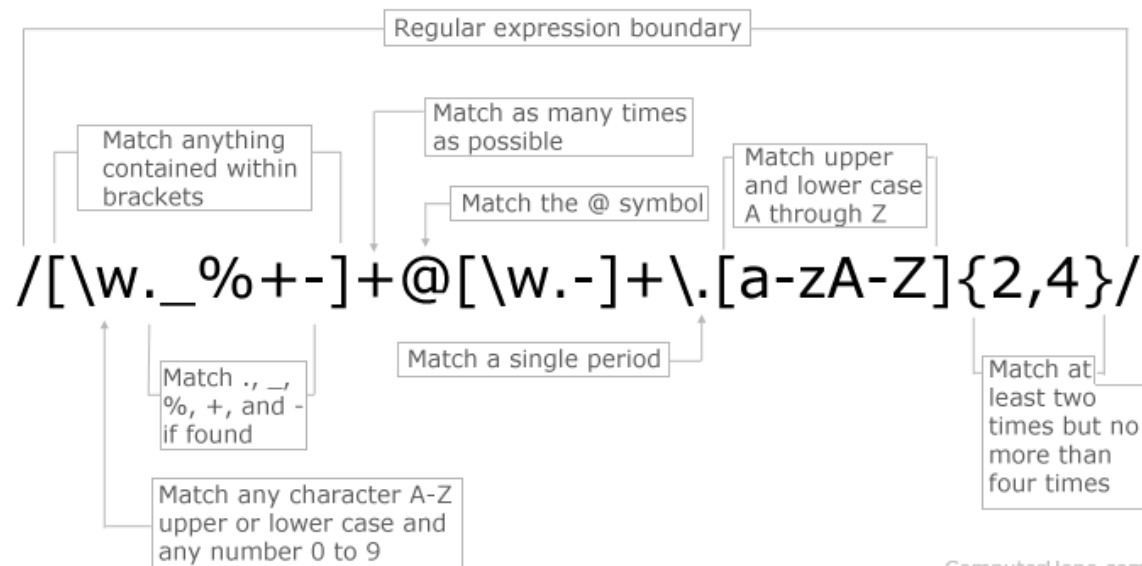
Contents

- Main objectives
- File visualization commands: cat, more, less
- Sort, cut, paste and join commands
- Regular expressions
- Finding files by grep and find

Regular expressions

- “Regex” are special strings representing a pattern to be matched in a search operation
- Applications: Java, Perl, grep, sed, vi

Regular Expression E-mail Matching Example



ComputerHope.com

Regular expressions

- **Metacharacters:** special characters (or sequences of characters) that are used to represent something else

Metacharacter	Meaning
.	Matches any single carácter.
*	Zero or more of the preceding carácter.
[...]	The characters listed inside the brackets are part of a matching-character set.
[^...]	Characters inside the brackets are a NON-matching set. Any character not inside the brackets is a matching character.
^...	Start of string or line.
...\$	End of string, or end of line.
\\	Literal backslash
\^	Literal caret
\\$	Literal dollar
\.	Literal period
*	Literal asterisk
\[Literal open bracket
\]	Literal close bracket
\(expression\)	Expression will match as a group

Regular expressions

■ Examples

- Whatever expression

`.*`

- A expression which begins by lowercase.

`^[a-z].*`

- A expression which begins by a character different from a number.

`^[^0-9].*`

- The expression begins by lowercase and ends in a no lowercase

`^[a-z].*[^a-z]$`

Contents

- Main objectives
- File visualization commands: cat, more, less
- Sort, cut, paste and join commands
- Regular expressions
- Finding files by grep and find

Finding files by grep and find

■ grep

- searches a file for a particular pattern of characters, and displays all lines that contain that pattern
- `grep [options] pattern [files]` (pattern → Regex)
- Options
 - `-c` : This prints only a count of the lines that match a pattern
 - `-i` : Ignores, case for matching
 - `-n` : Display the matched lines and their line numbers.
 - `-v` : This prints out all the lines that do not matches the pattern

Exercise 2

- Create a file with the following contents

```
Ab.qqqqB  
PA.C  
P.C  
,Ad.qqqB  
Ab.qqqqBl  
AbqqqqB
```

- Write a regular expression that recognizes a line of text that begins with A or P, followed by any character, followed by a period and ending with B or C.
- Use this regex in combination with grep and verify that only the two first lines of text accomplish the regex. Could you explain why?

Finding files by grep and find

■ find

- It can search the entire filesystem to find files and directories according to the search criteria you specify.
- find (starting directory) (matching criteria and actions)
- Criteria
 - -name name: The filename is name
 - -type c: Specifies file type: f=plain text, d=directory
 - -user usr: The file's owner is usr
 - -group grp: The file's group owner is grp
 - -size n: File is n blocks big (a block is 512 bytes)
 - -atime n: File was accessed n days ago
 - -mtime n: File was modified n days ago
 - -perm p: The file's access mode is p (where p is an integer)

Finding files by grep and find

■ find

□ Criteria

- Use + (plus) and - (minus) modifiers with the atime, mtime, and size criteria to increase their usefulness

`-mtime +7`

Matches files modified more than seven days ago

`-atime -2`

Matches files accessed less than two days ago

`-size +100`

Matches files larger than 100 blocks (50KB)

Finding files by grep and find

■ find

□ Actions

- -print: Display pathnames of matching files.

- -exec cmd: Execute command cmd on a file.

- Exec commands must end with \; and may use {} as a placeholder for each file that the find command locates

-exec ls -l {} \;

Finding files by grep and find

■ find

□ Examples

- To find and report all C language source code files starting at the current directory

```
find . -name \*.c -print
```

- To report all files starting in the directories /mydir1 and /mydir2 larger than 2,000 blocks (about 1,000KB) and that have not been accessed in over 30 days

```
find /mydir1 /mydir2 -size +2000 -atime +30 -print
```

- To show a long listing starting in /mydir of files not modified in over 20 days or not accessed in over 40 days

```
find /mydir \(-mtime +20 -o -atime +40\) -exec ls -l {} \;
```

Exercise 3

- Make a list of the files that occupy more space, show the first five. The command `head -n` can be helpful.
- Show the files in the current directory that have a certain size



DEPARTAMENTO
DE SISTEMAS
INFORMÁTICOS



Regular expression, filters, find and other commands

Enrique Arias

Universidad de Castilla–La Mancha