

**The Tenth International Conference on Advances in Databases, Knowledge, and Data Applications**

**Mai 20 - 24, 2018 - Nice/France**

# **Powerful Unix-Tools - sort & uniq & comm & join**

**Andreas Schmidt**

**Department of Informatics and  
Business Information Systems  
University of Applied Sciences Karlsruhe  
Germany**

**Institute for Automation and Applied Informatics  
Karlsruhe Institute of Technologie  
Germany**

## sort

- Sort lines of text files
- Write sorted concatenation of all FILE(s) to standard output.
- With no FILE, or when FILE is -, read standard input.
- sorting alphabetic, numeric, ascending, descending, case (in)sensitive
- column(s)/bytes to be sorted can be specified
- Random sort option (-R)
- Remove of identical lines (-u)
- Examples:
  - sort file city.csv starting with the second column (field delimiter: ,)  
`sort -k2 -t',' city.csv`
  - merge content of file1.txt and file2.txt and sort the result  
`sort file1.txt file2.txt`

## sort - examples

- sort file by country code, and as a second criteria population (numeric, descending)

```
sort -t, -k2,2 -k4,4nr city.csv
```

field separator: ,

numeric (-n), descending (-r)

second sort criteria from column 4 to column 4

first sort criteria from column 2 to column 2

## sort - examples

- Sort by the second and third character of the first column  
`sort -t, -k1.2,1.2 city.csv`
- Generate a line of unique random numbers between 1 and 10  
`seq 1 10 | sort -R | tr '\n' ' '`
- Lottery-forecast (6 from 49) - defective from time to time ;-)  
`seq 1 49 | sort -R | head -n6`
- Test if a file is sorted  
`seq 1 10 | sort -R | sort -c`

## uniq (1)

- report or omit repeated lines
- Filter adjacent matching lines from INPUT
- Range of comparison can be specified (first n chars, skip first m chars)
- options:
  - -c: count number of occurrences
  - -d: only print duplicate lines
  - -u: only print unique line
  - -i: ignore case
  - -w<num>: compare not more than <num> characters per line

## uniq - example

- file1.txt

Barcelona  
Bern  
Chamonix  
Karlsruhe  
Pisa  
Porto  
Rio

- file2.txt

Andorra  
Barcelona  
Berlin  
Pisa  
Porto

- Intersection:

```
$ cat file*.txt | sort | uniq -d  
Barcelona  
Pisa  
Porto
```

- Counting:

```
cat file*.txt | sort | uniq -c  
  1 Andorra  
  2 Barcelona  
  1 Berlin  
  1 Bern  
  1 Chamonix  
  1 Karlsruhe  
  2 Pisa  
  2 Porto  
  1 Rio
```

## Compare Operator

- comm - compare two sorted files line by line

Barcelona  
Bern  
Chamonix  
Karlsruhe  
Pisa  
Porto  
Rio

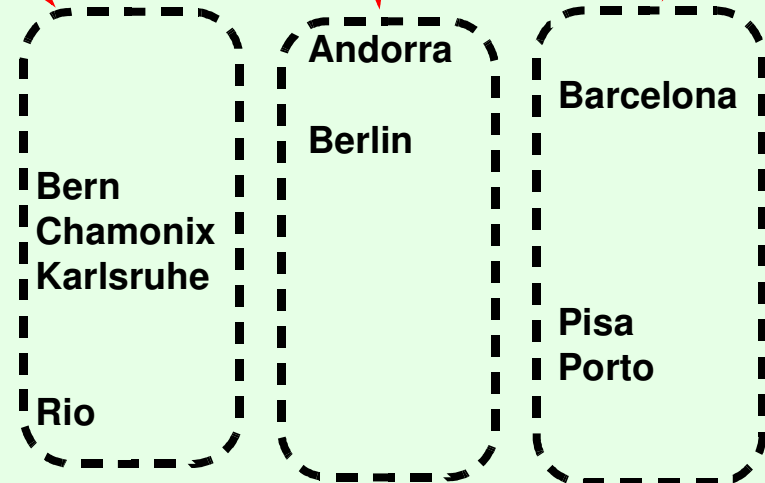
Andorra  
Barcelona  
Berlin  
Pisa  
Porto

comm

only in file1

only in file2

in file1  
and file2



- Options:

- -1: suppress column 1
- -2: suppress column 2

- -3: suppress column 3
- --total: output a summary



## Join operations

- join - join lines of two files on a common field
- Fields to compare must be sorted (alphabetic, not numeric)
- Output fields can be specified
- Example:

```
sort -k2 -t, city.csv | join -t, -12 -22 - country.csv \  
-o1.1,2.1,1.3,1.4
```

# Join Operation

- city.csv

```
Aachen, D, "Nordrhein Westfalen", 247113, NULL, NULL
Aalborg, DK, Denmark, 113865, 10, 57
Aarau, CH, AG, NULL, NULL, NULL
Aarhus, DK, Denmark, 194345, 10.1, 56.1
Aarri, WAN, Nigeria, 111000, NULL, NULL
...
```

- country.csv

```
...
Germany, D, Berlin, Berlin, 356910, 83536115
Djibouti, DJI, Djibouti, Djibouti, 22000, 42764
Denmark, DK, Copenhagen, Denmark, 43070, 524963
Algeria, DZ, Algiers, Algeria, 2381740, 2918303
Spain, E, Madrid, Madrid, 504750, 39181114
...
```

```
sort -k2 -t, city.csv | join -t, -12 -22 - country.csv \
-o1.1,2.1,1.3,1.4
```

```
Aachen, Germany, "Nordrhein Westfalen", 247113
Aalborg, Denmark, Denmark, 113865
Aarau, Switzerland, AG, NULL
Aarhus, Denmark, Denmark, 194345
Aarri, Nigeria, Nigeria, 111000
Aba, Nigeria, Nigeria, 264000
Abakan, Russia, "Rep. of Khakassiya", 161000
```