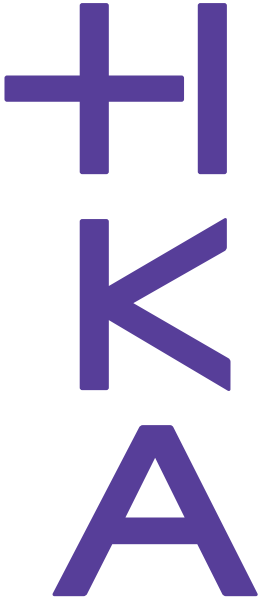
Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022)

24 - 26 October, 2022

Tutorial

Knowledge Discovery and Information Retrieval using the Shell

Andreas Schmidt

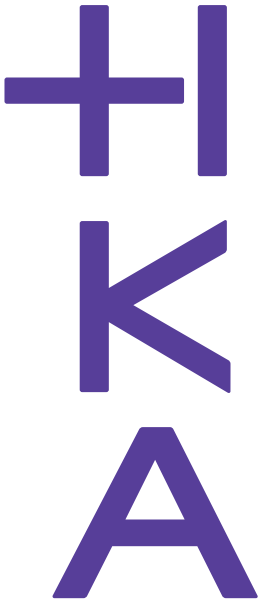
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Germany

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

Resources available

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



https://www.smiffy.de/KDIR-2022/ [1]

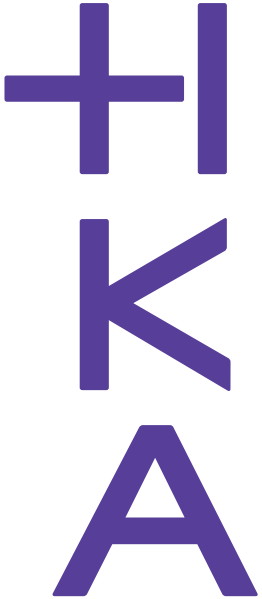
- Slideset
- 3 Exercises
- Command refcard
- Many examples
- Example datasets
- Further resources

1. all materials copyright 2017, 2018, 2019,2020, 2021, 2022 by andreas schmidt

Outlook

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



- Introduction
- Functionality Overview
- Filter & Pipes Architecure
- Command Overview Part I
- Exercise I: Start solving a criminal case using the shell
- Command overview Part II
- Exercise II: Solve the criminal case from Exercise I
- sed & awk
- Summary & Outlook

Coreutils

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- The GNU Core Utilities are the basic file, shell and text manipulation utilities of the GNU operating system. These are the core utilities which are expected to exist on every operating system. [1]

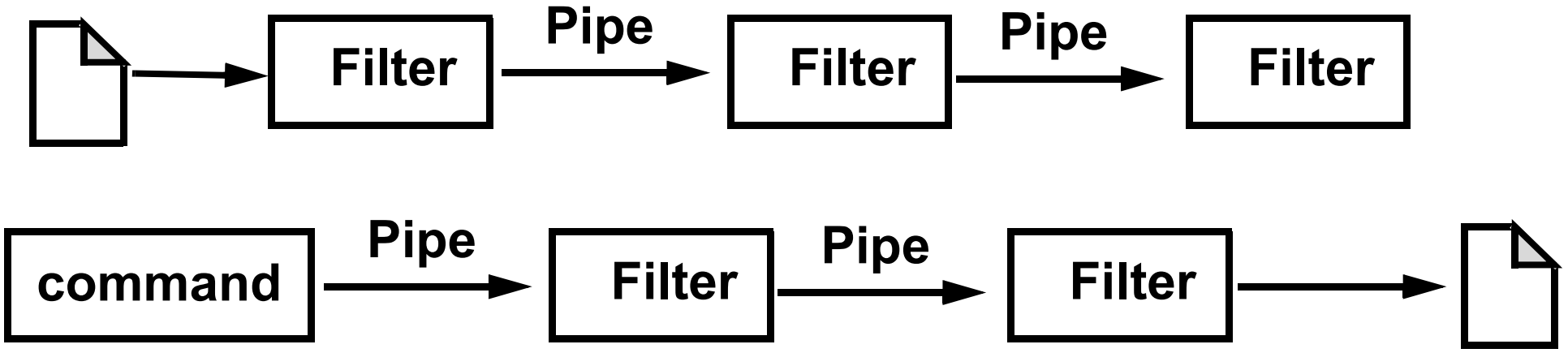
- These utilities are typically used in a Filter & Pipes Architectural pattern, where the output of the first utility program acts as input for the next utility program (and so on ...)

[1] https://www.gnu.org/software/coreutils/

Filter and Pipes Architecture

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

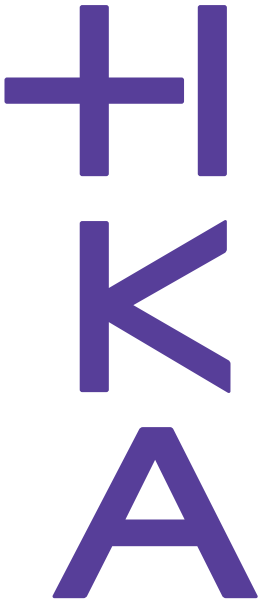
- Architectural Pattern: Filter and Pipes (Douglas McIlroy, 1973)
- Data exchange between processes
- Loose coupling
- POSIX Standard
- Filter represent data-sources and data-sinks



Why Should I use these Tools (Coreutils)?

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



With R and Python, there exist great tools that can perform the same job (and much more)

- It's already on your computer and nothing needs to be installed[1]
- You don't need to learn a programming language
- You don't need an editor, compiler or interpreter
- Low main memory footprint
- You got first results after 20 sec.
- Intuitive iterative development cycle (add filter by filter) ... like lego blocks
- It makes fun !!!!

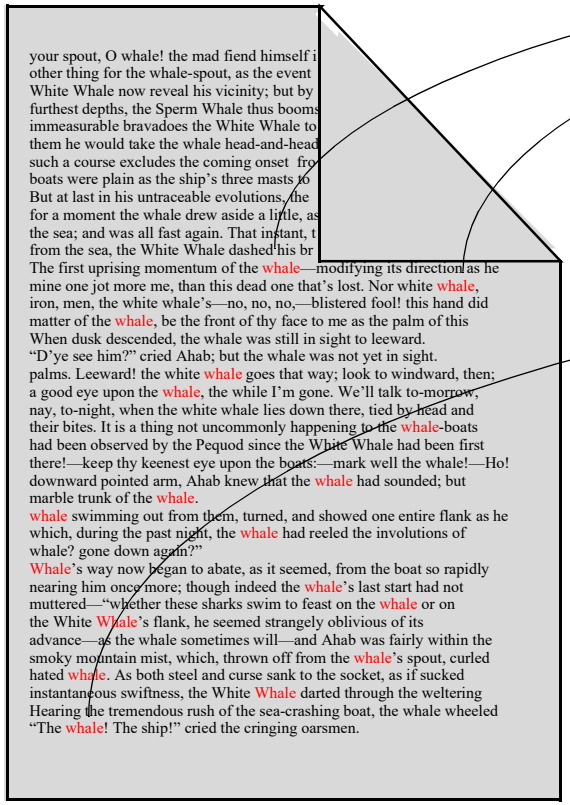
1. If you have a Linux or Mac-computer. Windows users have to install *cygwin* or the *Windows Subsytem for Linux (WSL)* to use theese tools.

Overview Functionality of the Core Utils

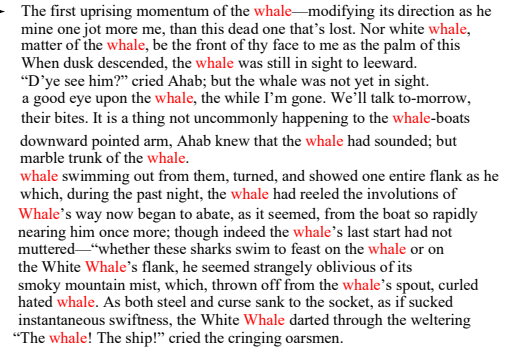
Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- **Selection/extraction** from unstructured File(s) - search for strings/regexs



tools: **grep**, sed, awk



possible output:

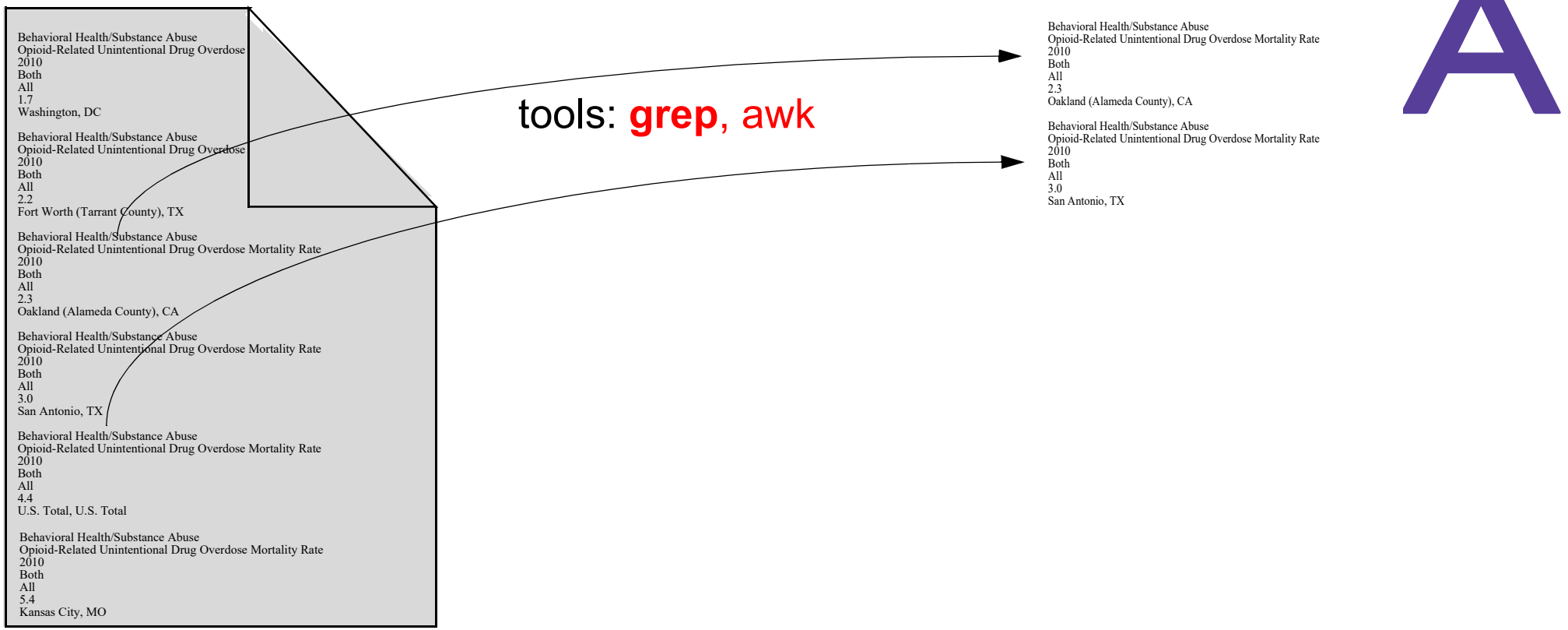
- lines (not) containing pattern
- only pattern (typically with regexps)
- filename(s) (not) containing pattern
- filename(s) & line (not) containing pattern
- additional line numbers
- with additional context lines

Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- **Selection/extraction** from structured File(s)

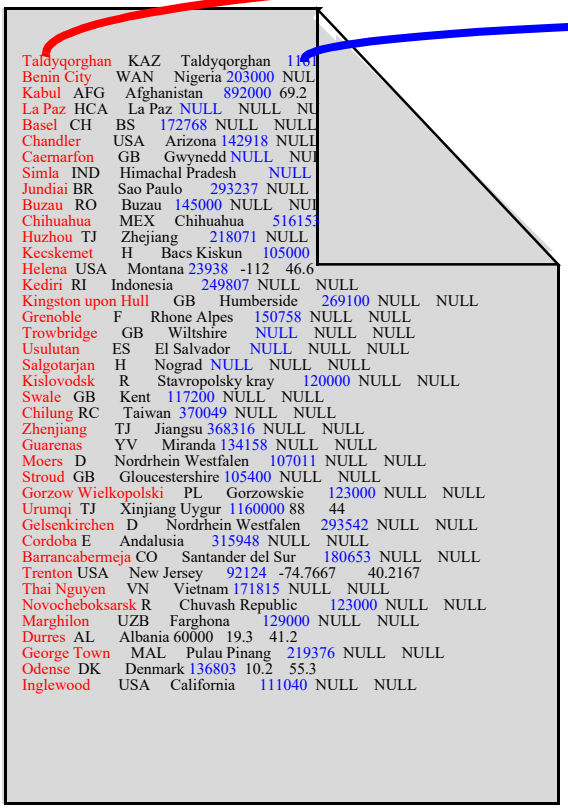


Overview Functionality of the Core Utils

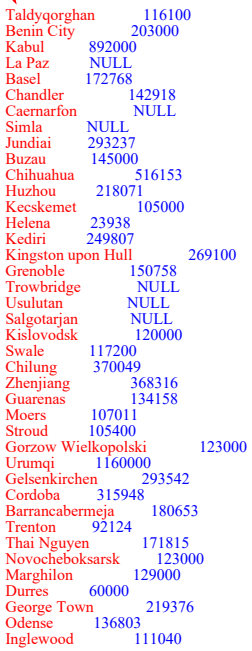
Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- **Projection** from structured Files (extract columns)



tools: cut, awk



Specification of ...

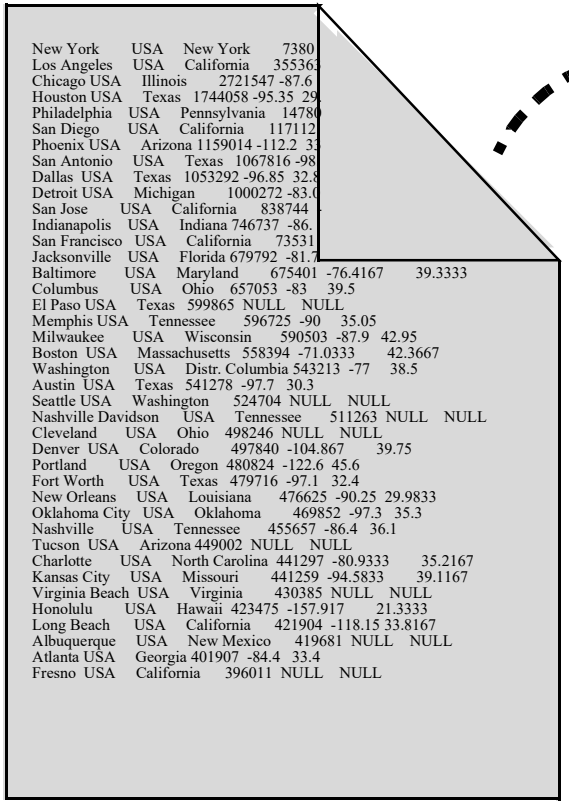
- Column separator
- desired column numbers

Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- **Sorting** of files



tools: sort

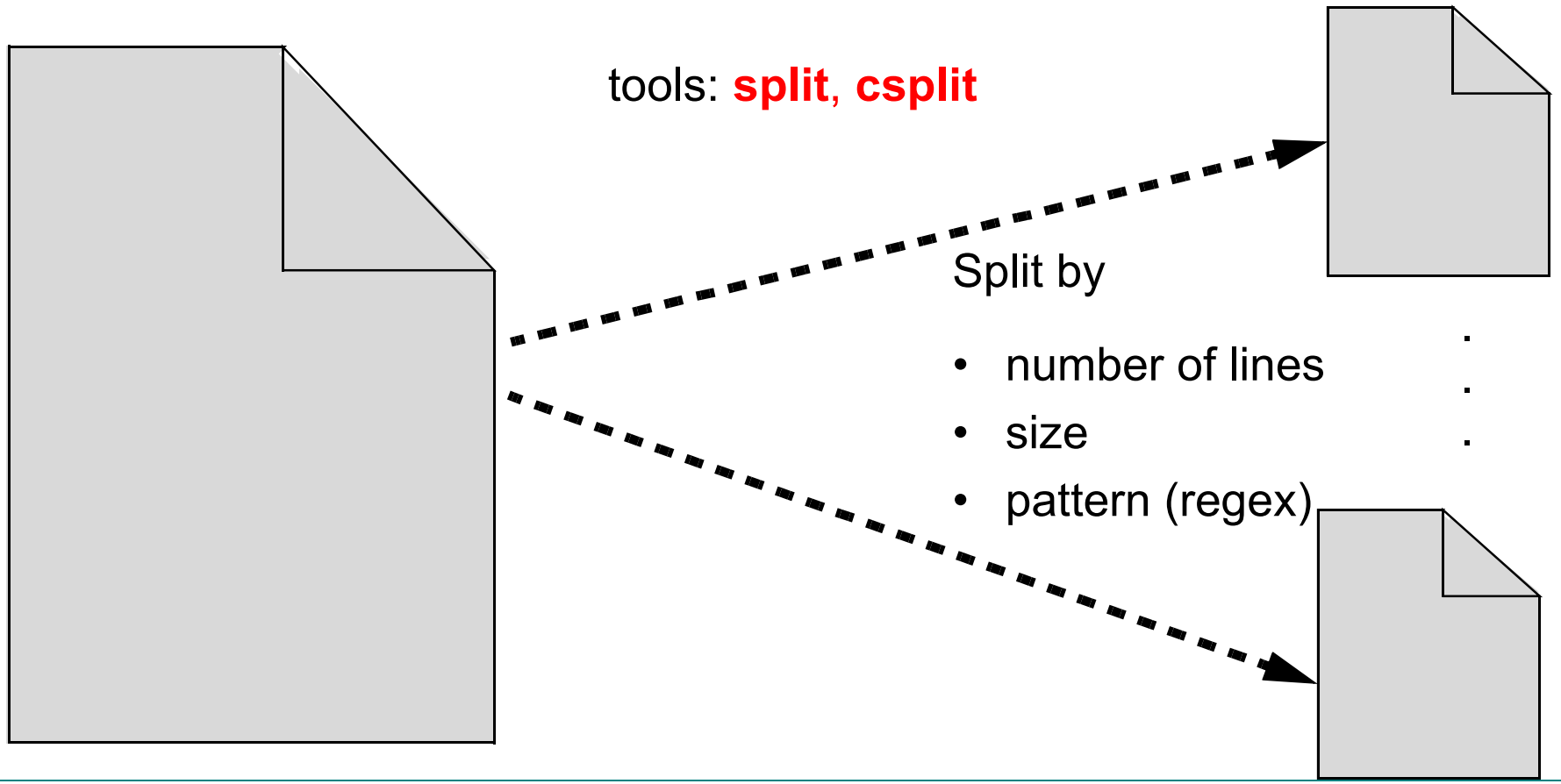
- line-by-line sorting
- Multiple critera sort
- alphabetic, numeric, random sort
- ascending/descending

Phoenix USA Arizona 1159014 -112.2 33.2
Tucson USA Arizona 449002 NULL NULL
Los Angeles USA California 3553638 -118 34
San Diego USA California 1171121 -117.167 32.7333
San Jose USA California 838744 -121.933 37.3667
San Francisco USA California 735315 -122.383 37.6167
Long Beach USA California 421904 -118.15 33.8167
Fresno USA California 396011 NULL NULL
Denver USA Colorado 497840 -104.867 39.75
Washington USA Distr. Columbia 543213 -77 38.5
Jacksonville USA Florida 679792 -81.7 30.5
Atlanta USA Georgia 401907 -84.4 33.4
Honolulu USA Hawaii 423475 -157.917 21.3333
Chicago USA Illinois 2721547 -87.6333 41.8833
Indianapolis USA Indiana 746737 -86.2833 39.7333
New Orleans USA Louisiana 476625 -90.25 29.9833
Baltimore USA Maryland 675401 -76.4167 39.3333
Boston USA Massachusetts 558394 -71.0333 42.3667
Detroit USA Michigan 1000272 -83.0167 42.4167
Kansas City USA Missouri 441259 -94.5833 39.1167
Albuquerque USA New Mexico 419681 NULL NULL
New York USA New York 7380906 -74 40.4
Charlotte USA North Carolina 441297 -80.9333 35.2167
Columbus USA Ohio 657053 -83 39.5
Cleveland USA Ohio 498246 NULL NULL
Oklahoma City USA Oklahoma 469852 -97.3 35.3
Portland USA Oregon 480824 -122.6 45.6
Philadelphia USA Pennsylvania 1478002 -75.25 39.8833
Memphis USA Tennessee 596725 -90 35.05
Nashville Davidson USA Tennessee 511263 NULL NULL
Nashville USA Tennessee 455657 -86.4 36.1
Houston USA Texas 1744058 -95.35 29.9667
San Antonio USA Texas 1067816 -98.4 29.3
Dallas USA Texas 1053292 -96.85 32.85
El Paso USA Texas 599865 NULL NULL
Austin USA Texas 541278 -97.7 30.3
Fort Worth USA Texas 479716 -97.1 32.4
Virginia Beach USA Virginia 430385 NULL NULL
Seattle USA Washington 524704 NULL NULL
Milwaukee USA Wisconsin 590503 -87.9 42.95

Overview Functionality of the Core Utils

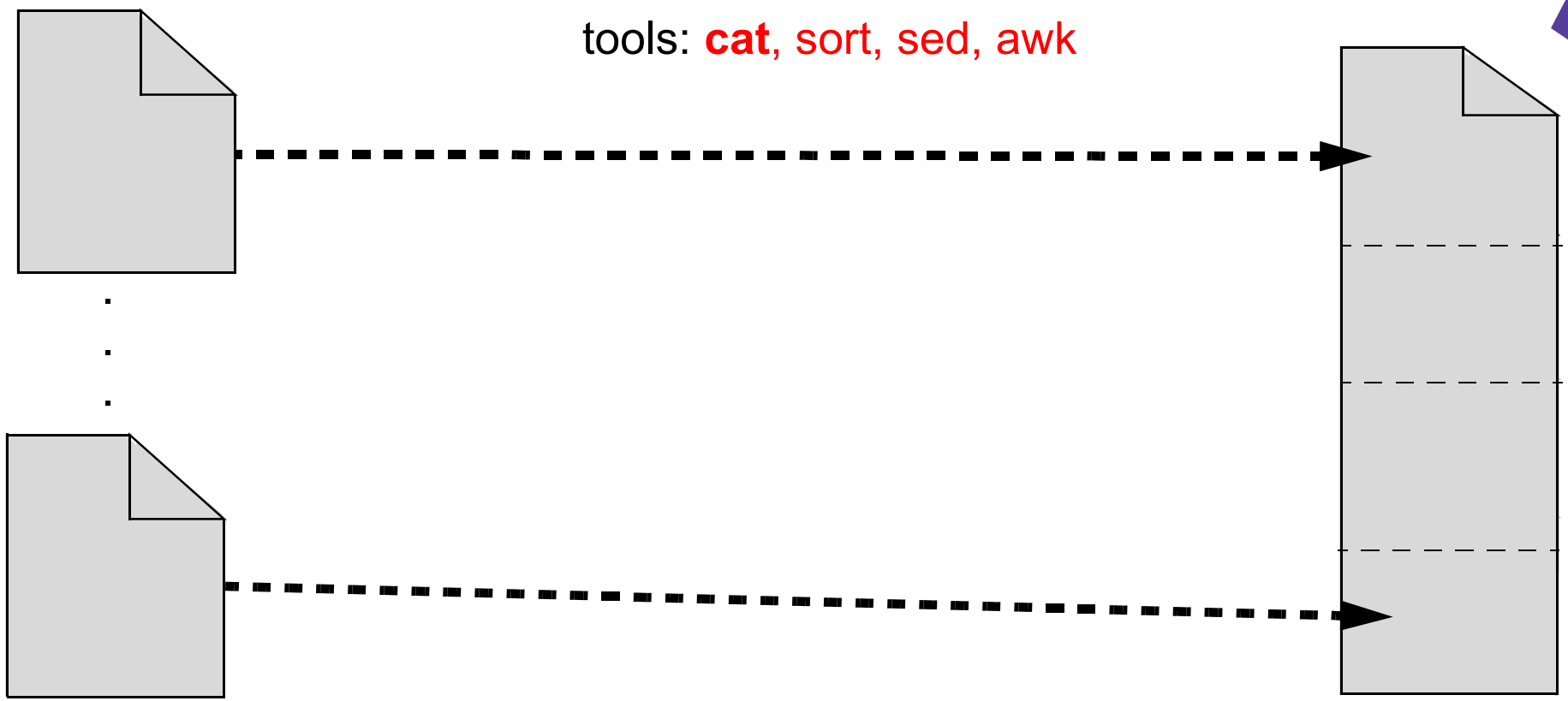
Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- **Splitting Files**



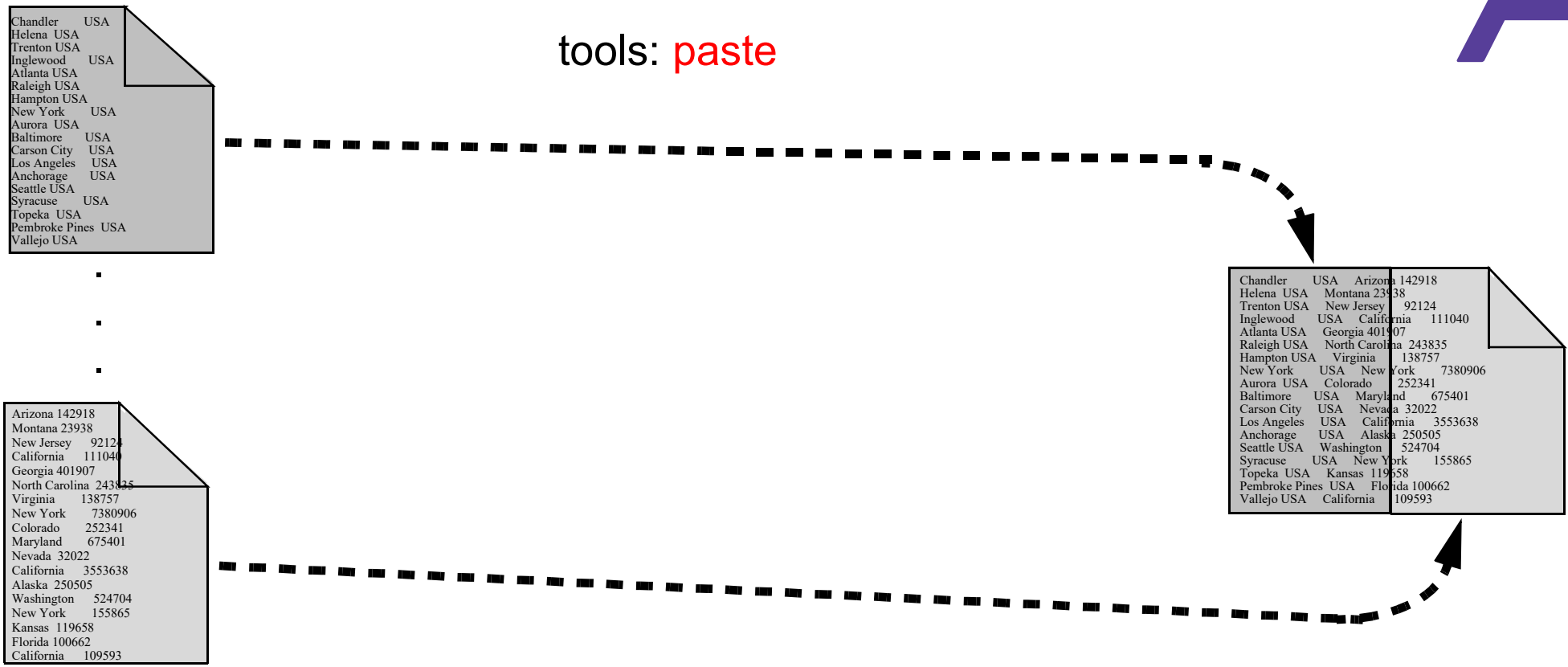
Overview Functionality of the Core Utils

- **Merging Files (1) - line-by-line**



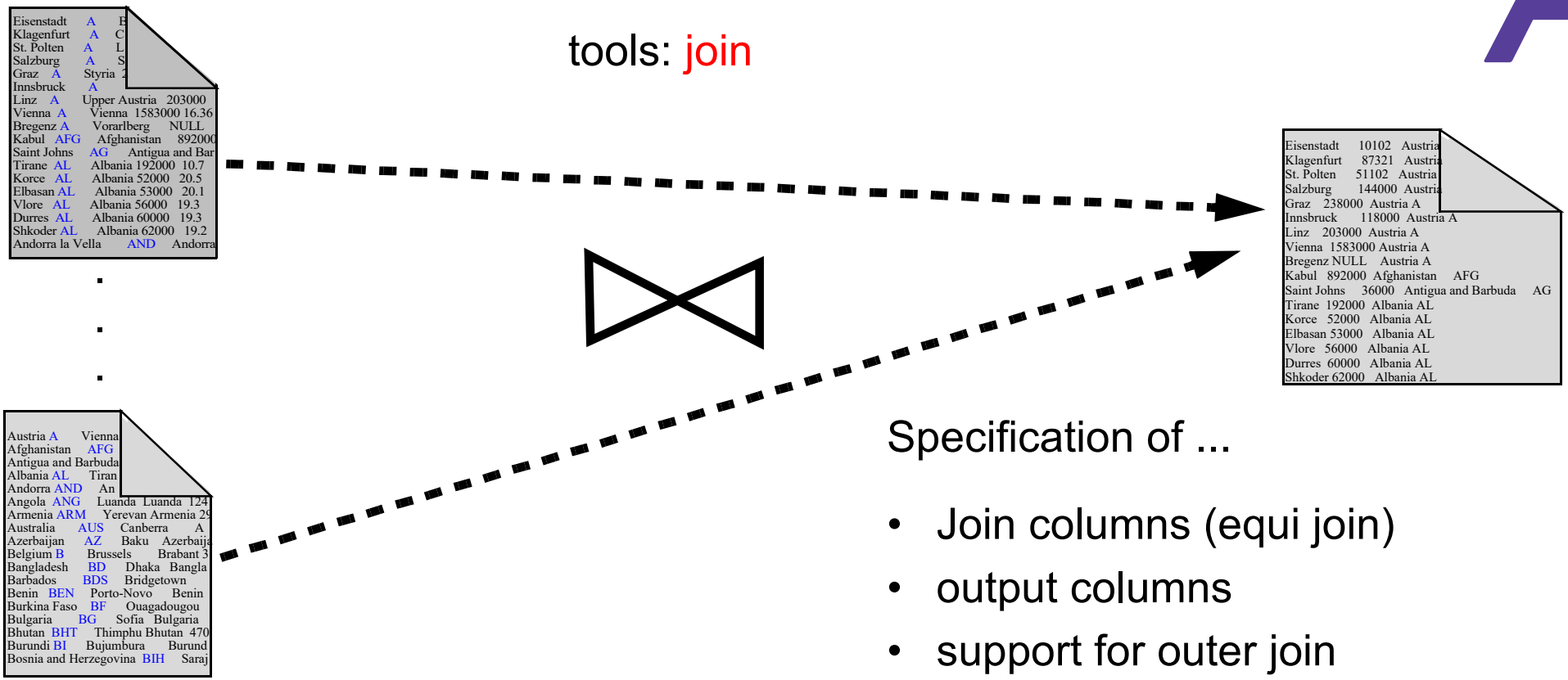
Overview Functionality of the Core Utils

- **Merging Files (2) - column-by-column**



Overview Functionality of the Core Utils

- **Merging Files (3) - by matching column**

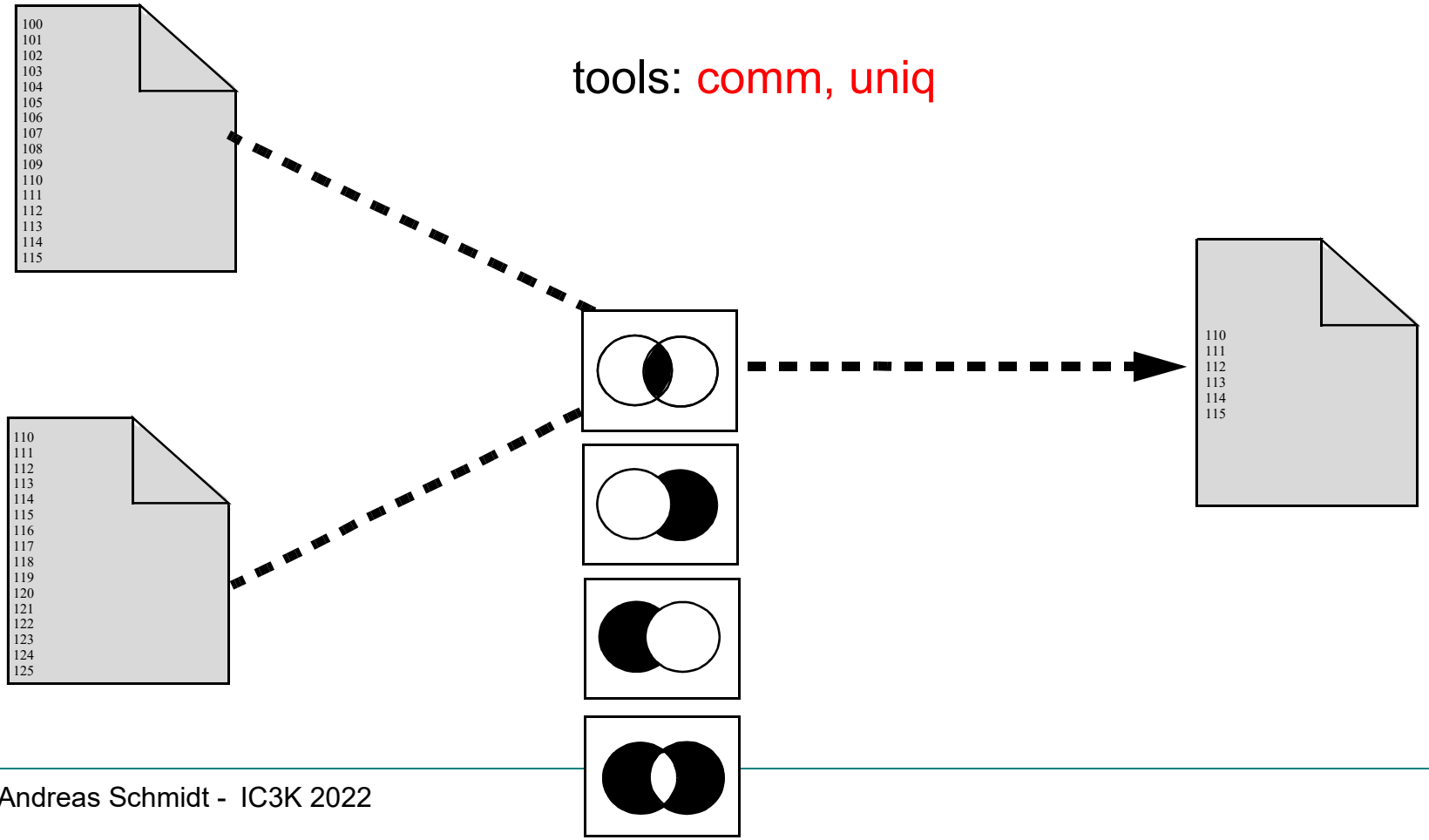


Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- **Set based operations: Intersect, Minus, Duplicate detection/ elimination**

tools: comm, uniq

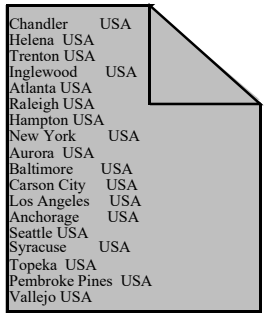


Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- **Script based modification of Files (programatic editing)**

Adding lines



Modifying lines

Deleting lines

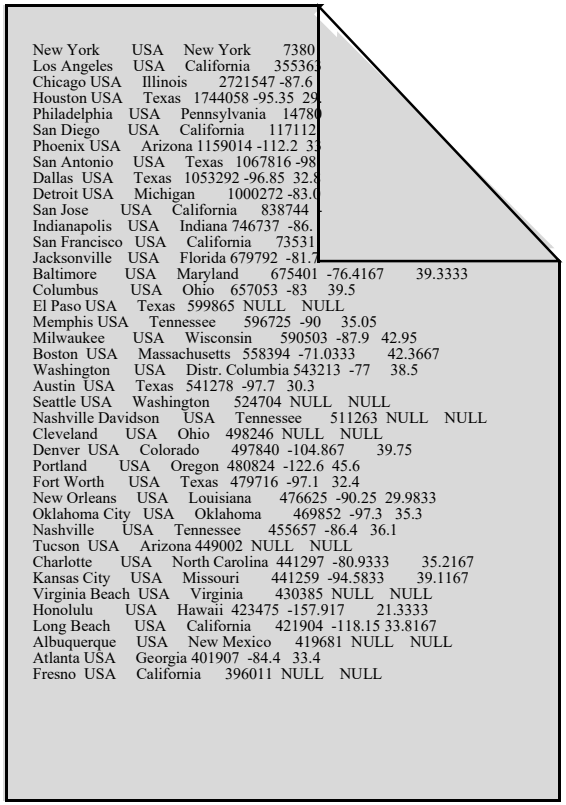
tools: **sed**, awk

Overview Functionality of the Core Utils

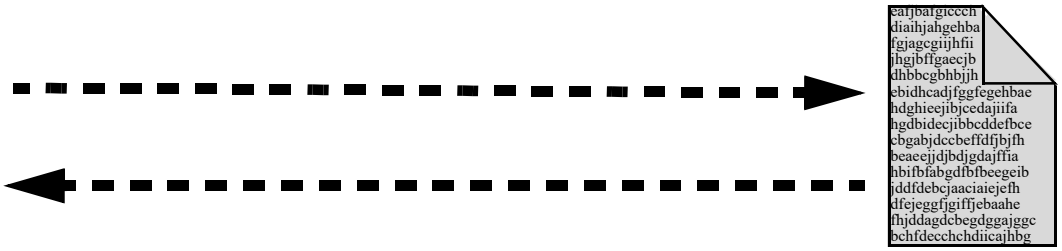
Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- **File compression**



tools: **gzip, gunzip, bzip2, bunzip2**

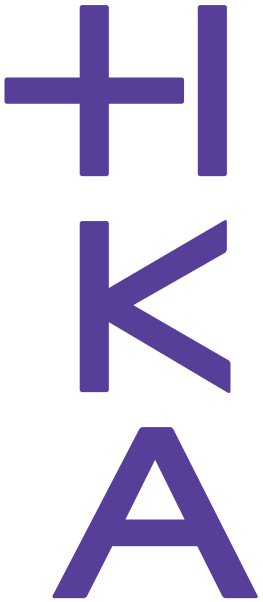


Operations on compressed data:

- search
- list/concatenate
- inspect interactively

Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



- Character based **transformations**
  - Translate single characters or ranges of characters (i.e. A-Z -> a-z)
  - Delete specified characters
  - Squeeze repeated occurences of specified characters

tools: tr, sed

- **Transformations** based on regular expressions
  - example (date transformation):

```
31.9.2019 -> 2019-9-31
```

```
s#\b([0-9]{1,2})\.([0-9]{1,2})\.([0-9]{4})\b#\3-\2-\1#
```

matching pattern

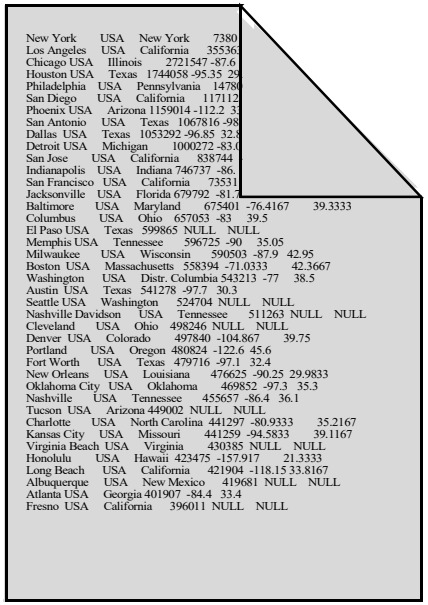
replace-ment

Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- **File Inspection**

Navigate/search in file



Show content

View first/last *n* lines

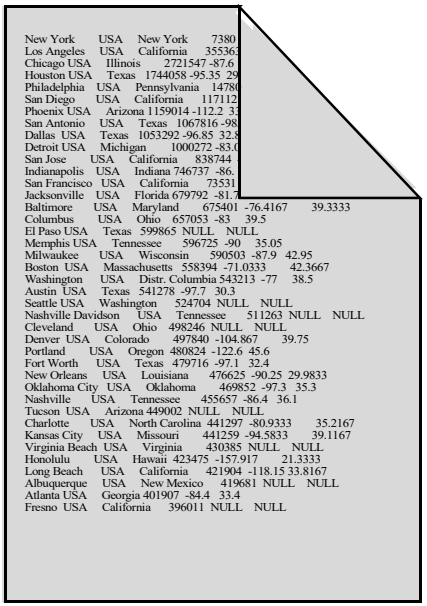
tools: **less**, **head**, **tail**, cat, sed

Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- **Counting** lines, words, bytes



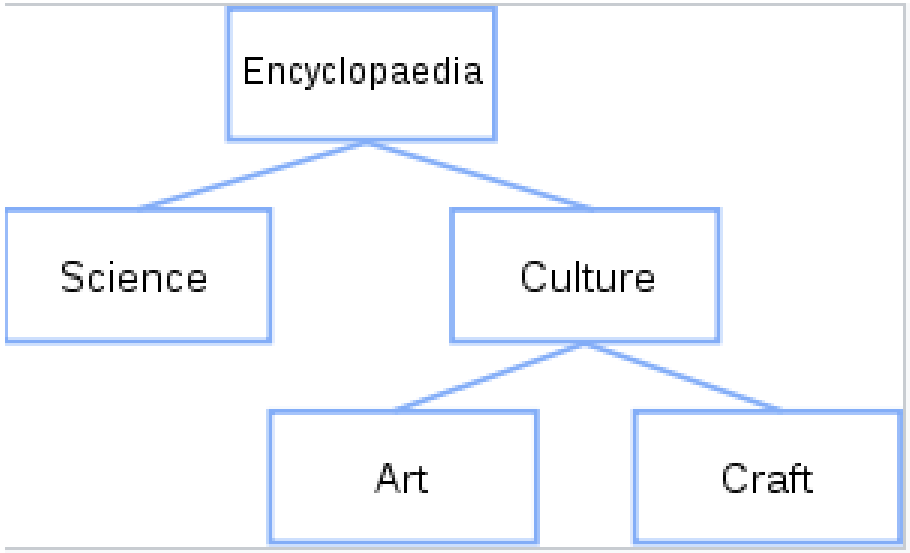
***n* lines**
***k* words**
***i* bytes**

tools: wc

Overview Functionality of the Core Utils

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- **Search directory tree**



tools: **find, ls**

Search by

- name pattern
- type
- access date
- user/group
- ...
- a mix of above

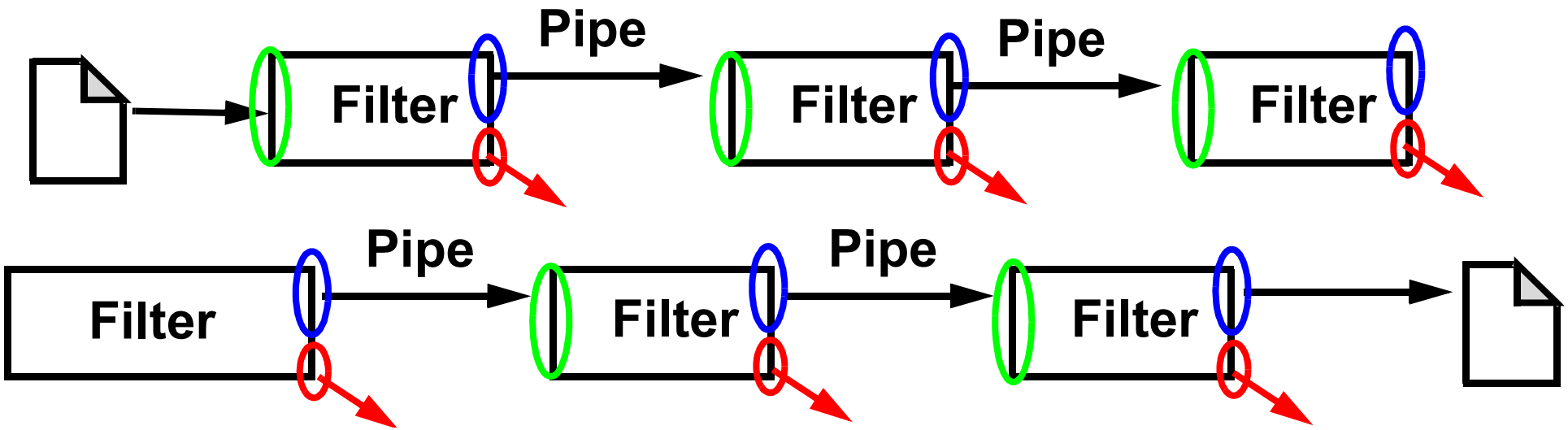
an additional action can be performed on the found files (default: print filename)

Filter and Pipes (Combining commands)

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

Communication between filters via channels

- Standard Input (STDIN)
- Standard Ouput (STDOUT)
- Standard Error (STDERR)



Communication Channels/Redirection

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- In-/Output Redirection
  - | : Pipe operator: Connect STDOUT of a command with STDIN of the next command
  - > : Redirect Standard Output (into file)
  - < : Redirect Standard Input (from file)
  - 2> : Redirect Standard Error (into file)
  - >> : Redirect Standard Output (append into file)

- Example[1]:

```
cut -d, -f1 city.csv | sort | uniq -c | \
    sort -nr | awk  '$1>1' > result.txt
```

1. https://www.smiffy.de/KDIR-2022/

Retrieving the names of cities which have „name siblings“

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

```
cut -d, -f1 city.csv|sort|uniq -c|sort -nr|awk '$1>1' > res.txt
```

```
Binjai
Hsinchu
Zhuhai
Jinxi
Reynosa
Livonia
"Hpa an"
Paterson
Kaifeng
Orlando
Brescia
Tepic
...
```

```
Aachen
Aalborg
Aarau
Aarhus
Aarri
Aba
Abakan
Abancay
Abeokuta
Aberdeen
Aberystwyth
...
```

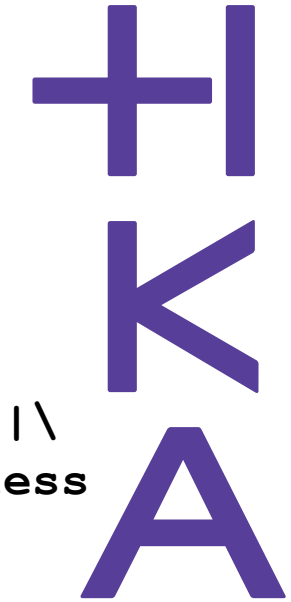
```
...
1 Leiyang
1 Lekoa
1 Lelystad
1 Lengshuijiang
1 Leninsk
3 Leon
1 Leshan
1 Leszno
1 Leticia
1 Leverkusen
...
```

```
3 Trujillo
3 Springfield
3 Merida
3 Leon
3 Kingston
3 Cordoba
3 Alexandria
3 "La Paz"
2 York
2 Yichun
...
1 Zurich
```

```
3 Trujillo
3 Springfield
3 Merida
3 Leon
3 Kingston
3 Cordoba
3 Alexandria
3 "La Paz"
2 York
2 Yichun
```

Another example: Word count

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



```
grep '[A-Za-z]+' -Eo moby-dick.txt | \
                    tr 'A-Z' 'a-z' | \
                                sort | uniq -c | sort -nr |\
                                                      less
```

```
The
Project
Gutenberg
EBook
of
Moby
Dick
or
The
Whale
...
```

```
the
project
gutenberg
ebook
of
moby
dick
or
the
whale
...
```

```
a
a
a
a
a
a
a
a
a
...
```

```
4805 a
   2 aback
   2 abaft
   3 abandon
   7 abandoned
   1 abandonedly
   2 abandonment
   2 abased
```

```
14715 the
 6742 of
 6517 and
 4805 a
 4707 to
 4241 in
 3100 that
 2536 it
 2532 his
```

Some Best Practice

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Incremental development of a complex pipe command (filter by filter)
- Typically, the last command is a *less* or *head* command, so that you can see, what's the result, but not get overrun by the large number of results
- While development: If the input is large, start with a *head -n<x>* command, to reduce the data lines to process or simply extract an sample using awk(*)
- Create intermediate result file, so that the entire process chain does not have to be repeatedly run through.

```
(*)
# 0.01 % extract
awk 'rand() < 0.0001 {print $0}' very-big-file.csv
```

What happens next ...

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Some general comments about the commands
- Short presentation of the most important commands
- for each command ...
  - mention most important features
  - possible preconditions
  - typically show one or two examples to clarify usage
  - link to further examples

- 2 (3) Exercises: Based on a series of police files, you have to solve a criminal case - thrilling !!!!! ;-)
  - Resources (see https://www.smiffy.de/KDIR-2022/):
    - Slideset
    - Command Refcard
    - Command examples

General comment

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Most of the commands accept the input from file or from STDIN. If no (or not enough) input files are given, it is expected that the input comes from STDIN (some commands like *join*, *comm* expect a „-“ character as parameter, if the input comes from STDIN)

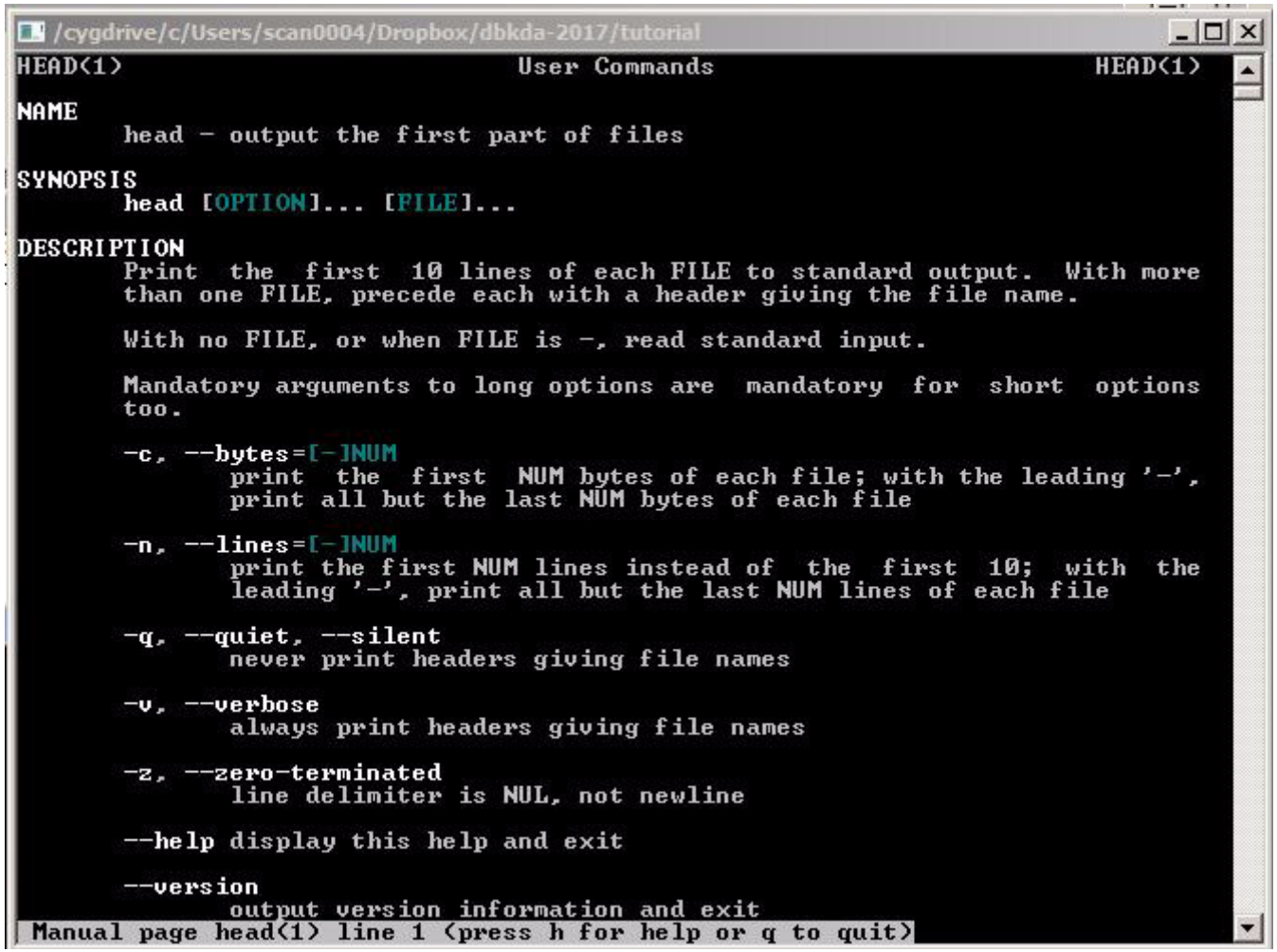
```
head -n4 my-file.txt
cat -n my-file.txt | head -n4
```

- Most of the commands have a lot of options which couldn’t be explained in detail. To get an overview of the possibilities of a command, simple type

```
man command
```

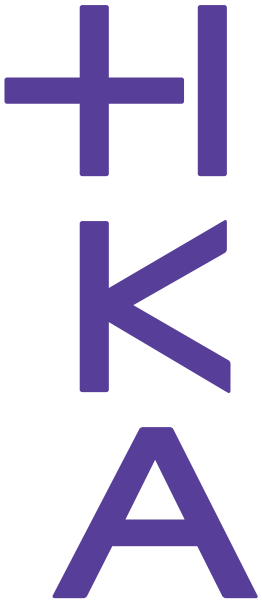
- Example:

```
man head
```


/cygdrive/c/Users/scan0004/Dropbox/dbkda-2017/tutorial
HEAD(1) User Commands HEAD(1)
NAME
head - output the first part of files
SYNOPSIS
head [OPTION]... [FILE]...
DESCRIPTION
Print the first 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name.
With no FILE, or when FILE is -, read standard input.
Mandatory arguments to long options are mandatory for short options too.
-c, --bytes=[-]NUM
print the first NUM bytes of each file; with the leading '-', print all but the last NUM bytes of each file
-n, --lines=[-]NUM
print the first NUM lines instead of the first 10; with the leading '-', print all but the last NUM lines of each file
-q, --quiet, --silent
never print headers giving file names
-v, --verbose
always print headers giving file names
-z, --zero-terminated
line delimiter is NUL, not newline
--help display this help and exit
--version
output version information and exit
Manual page head(1) line 1 (press h for help or q to quit)

cat command

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



- Print content of file to STDOUT

```
cat HelloWorld.java
```

- Concatenate files and writes them via redirection (>) to a file

```
cat german_cities.csv french_cities.csv > cities.csv
cat *_cities.csv > cities.csv
```

- Add line numbers to each line in file(s)

```
cat -n city.csv
```

- Create a file with input from STDIN:

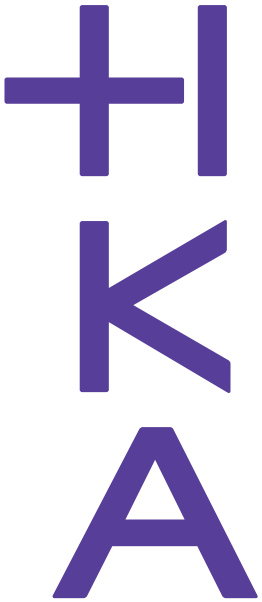
```
cat > grep-search-words.txt
Obama
Climate
CTRL-D
```

- More example:

```
https://www.smiffy.de/KDIR-2022/command-examples/cat,%20head,%20tail,%20less,%20wc
```

head/tail/wc command

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



- **head: view first *n* lines or skip last *n* lines of a file.**
  - View first 5 lines from file:

    ```
    head -n5 city.csv
    ```
  - Print all but the last 20 lines:

    ```
    head -n -20 city.csv
    ```

    to remove trailing line(s)

- **tail: view last *n* lines or start from line *n***
  - View last 10 lines of a file

    ```
    tail -n 10 city.csv
    ```

- **wc:** Count the number of lines, words and bytes

  ```
  wc city.csv
  ```

less command

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Page by page scrolling of a file or STDIN (also with search capability)
- Examples:

```
less city.csv
ls -l | less

man head      # inspection of man-pages with less !!
```

- Commands:
  - q : quit less
  - > : Goto end of file
  - < : Goto begin of file
  - f: Scroll forward one page
  - b: scroll backwards on page
  - *e, ret*,↓ : scroll forward one line
  - y,↑: scroll backwards one line
  - *n*d : scroll forward *n* lines (i.e. 20n)
  - *m*b : scroll backwards *m* lines
  - *n*g: Goto line <n>

less commands (2)

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- /*pattern* : Search forward the next line with *pattern*
- ?*pattern* : Search backward the previous line with *pattern*
- n : repeat previous search
- N : repeat previous search in reverse direction
- &*pattern* : Display only lines containing the *pattern* (type &<*ret*> to quit)
- !*command* : executes shell command
- v : invokes standard editor for file (at current position, if supported)

type **`man less`** for complete reference

grep command

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Print lines matching pattern (case sensitive)

```
grep USA city.csv
```

- Print lines containing the regular expression (City starting with ’S’, ending with ’g’)

```
grep -E 'S[a-z]+g,' city.csv
```

- Print only lines, not containing the String NULL

```
grep -v NULL city.csv
```

- Print lines which contain the pattern ’Agassi’

```
grep Agassi bbcsport/tennis/*.txt
```

when multiple files are queried,
the filename is part of the
result (<filename>:<line matching pattern)>

Search

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Print name of files which contain the pattern 'Agassi'

```
grep -l Agassi bbcsport/tennis/*.txt
```

- Print only matching part (i.e. 'Salzburg' instead of whole line)

```
grep -E -o 'S[a-z]+g' city.csv
```

- Look for lines containing words from a file (OR-Semantic)

```
grep -f grep-search-words.txt -E newsCorpora.csv
```

  - file: grep-search-words.txt

```
Obama
Climate
```

More example:

https://www.smiffy.de/KDIR-2022/command-examples/grep

File operations

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Print selected parts of lines from each file to standard output.

```
cut -d',' -f1,4 city.csv
```

Column separator

Output columns 1 and 4

- Output bytes 10 to 20 from each line

```
cut -b10-20 data.fixed
```

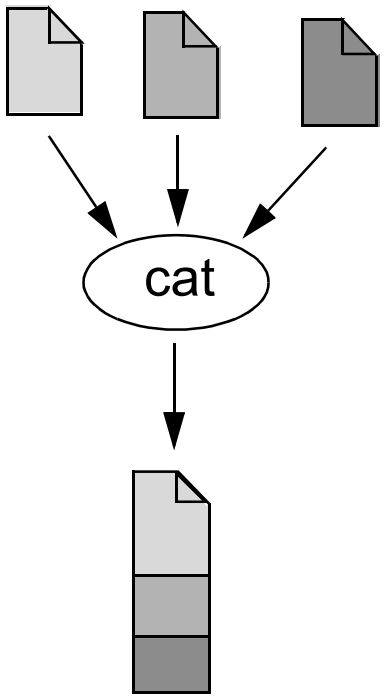
- Output bytes 1-5 and starting from position 20 to the end of line:

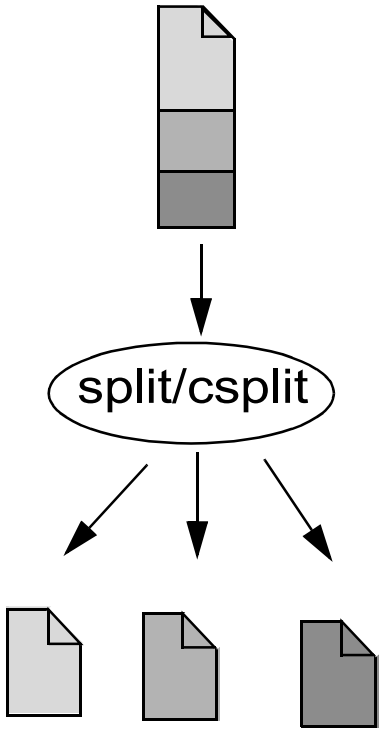
```
cut -b1-5,20- data.fixed
```

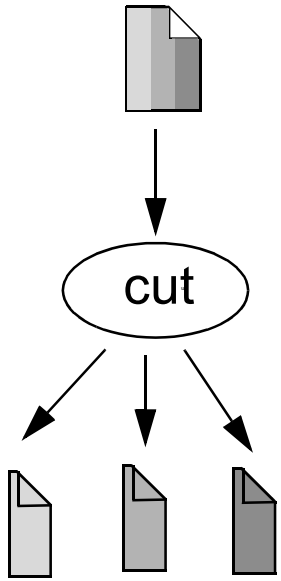
Summary of Fundamental File Operations

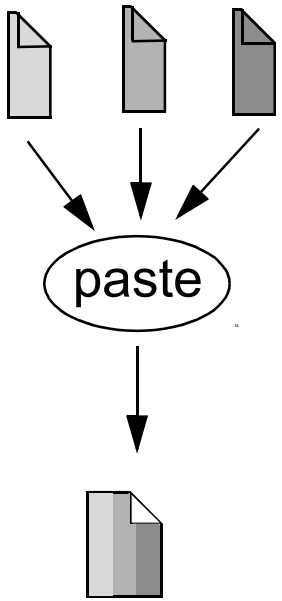
Hochschule Karlsruhe
University of
Applied Sciences

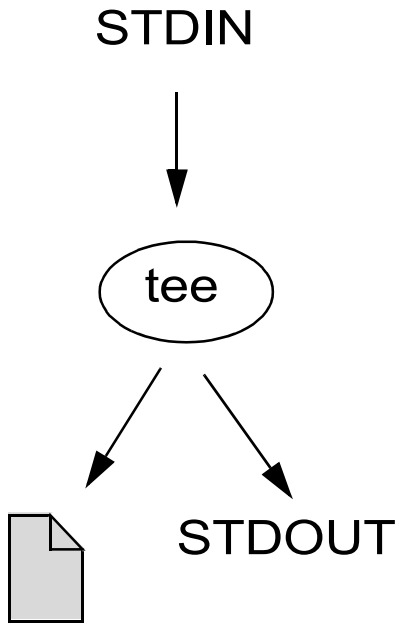
Fakultät für
Informatik und
Wirtschaftsinformatik





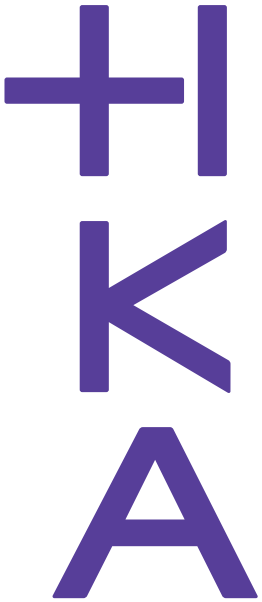






Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



And now you are prepared for a ...

thrilling exercise !!!!!!*

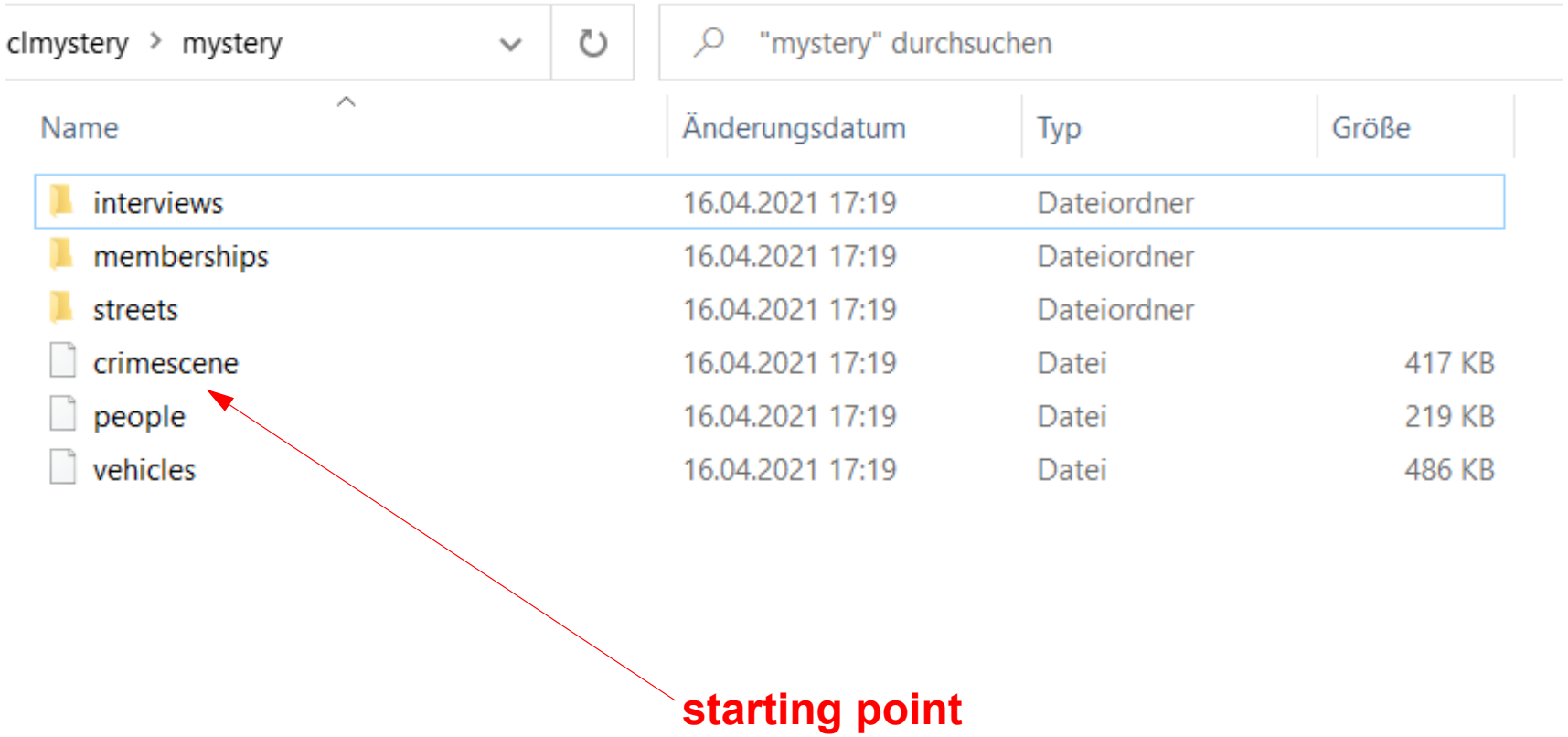
Go to the page www.smiffy.de/KDIR-2022, open the first Exercise and work on the task ...

(*) command line murders by Noah Veltman,
https://github.com/veltman/clmystery

mystery File/Directory Structure

Hochschule Karlsruhe
University of
Applied Sciences

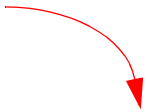
und
informatik



- mystery/people$

```
head mystery/people
***************
To go to the street someone lives on, use the file
for that street name in the 'streets' subdirectory.
To knock on their door and investigate, read the line number
they live on from the file.  If a line looks like gibberish, you're at the wrong
house.
***************

NAME     GENDER  AGE     ADDRESS
Alicia Fuentes   F       48      Walton Street, line 433
Jo-Ting Losev    F       46      Hemenway Street, line 390
...
Annabel Fuglsang M       40     Haley Street, line 176
Diego Michan     M       74     Wyola Place, line 25
```



- File mystery/streets/Haley_Street, lines 174 - 179

```
173 ...
174 pinto simile fuzing pestering neutralized atriums
daunted
175 irradiates liquidates flimflams dispossessed
176 SEE INTERVIEW #871877
177 balmy metamorphosis nervier pilfered
178 proofreaders steeping editorialized solutions
```

- Interviews:

```
$ ls mystery/interviews/interview-*| head -n5
mystery/interviews/interview-000296
mystery/interviews/interview-00448418
...
mystery/interviews/interview-871877
```

```
$ cat mystery/interviews/interview-871877
Mr. Fuglsang is male and has brown hair ...
```

sort

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Sort lines of text files and/or lines from STDIN
- Write sorted concatenation of all FILE(s) to standard output.
- sorting alpabetic, 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 the entries in file alphabetically

    ```
    sort member-list.txt
    ```
  - sort the entries in file (Format: <fist-name> <last-name>) by second column

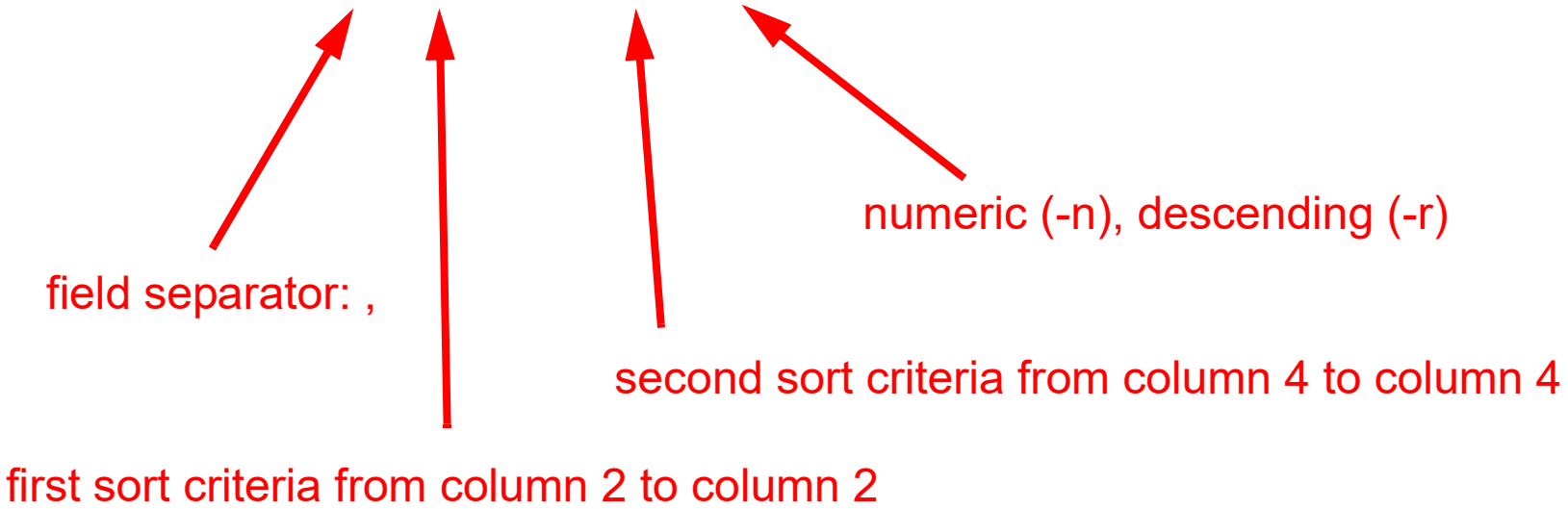
    ```
    sort -t' ' -k2 member-list.txt
    ```

field separator <space>

sort - examples

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

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



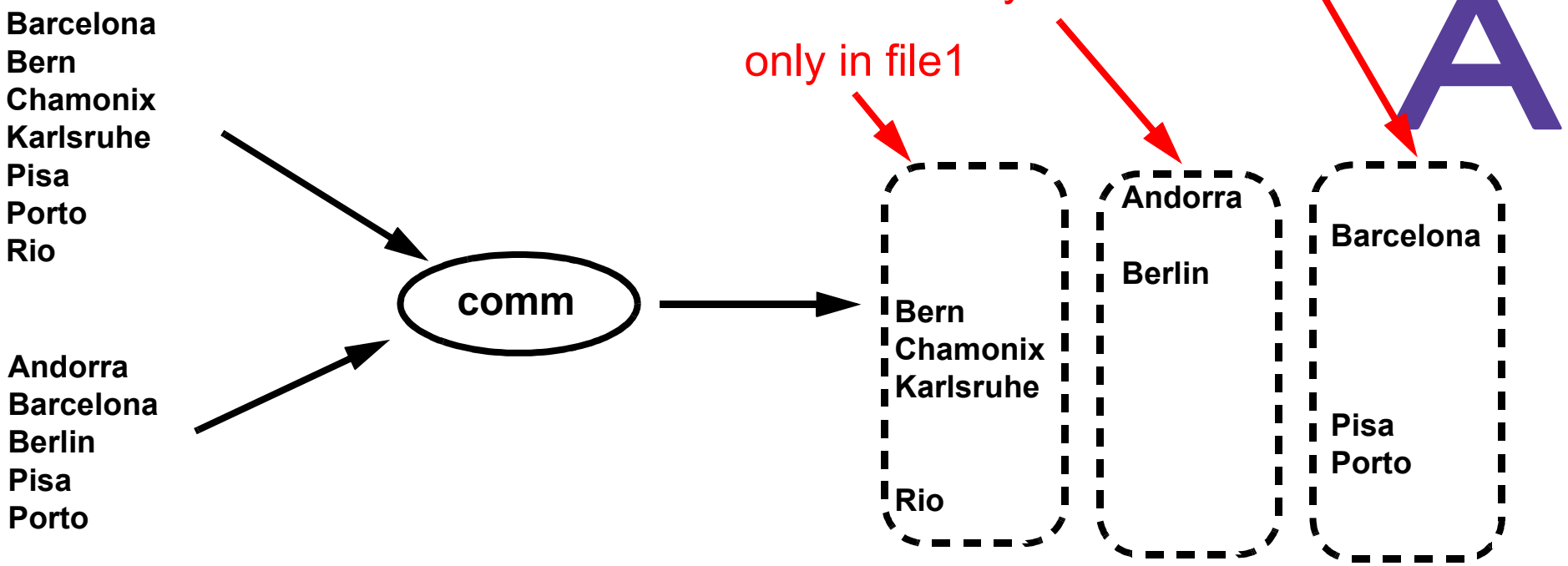
- More example:

```
https://www.smiffy.de/KDIR-2022/command-examples/sort
```

Compare Operator

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- comm - compare two **sorted** files line by line



- Options:
  - -1: supress column 1
  - -2: supress column 2
  - -3: supress column 3
  - --total: output a summary

set semantic with sorted input !!!

uniq

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- report or omit **repeated** lines
- Filter adjacent matching lines from INPUT
- Range of comparision can be specified (first n chars, skip first m chars)
- options:
  - -c: count number of occurences
  - -d: only print duplicate lines
  - -u: only print unique line
  - -i: ignore case

set semantic with sorted input !!!

uniq - example

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- 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
```

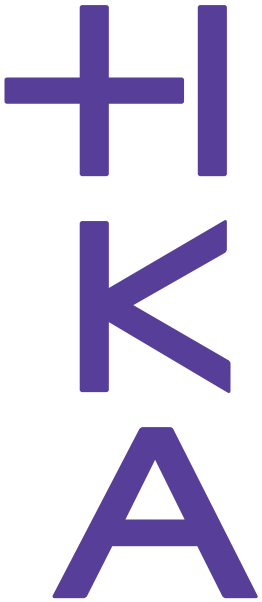
- More example:

`https://www.smiffy.de/KDIR-2022/command-examples/comm,%20uniq`

Exercise Part II

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

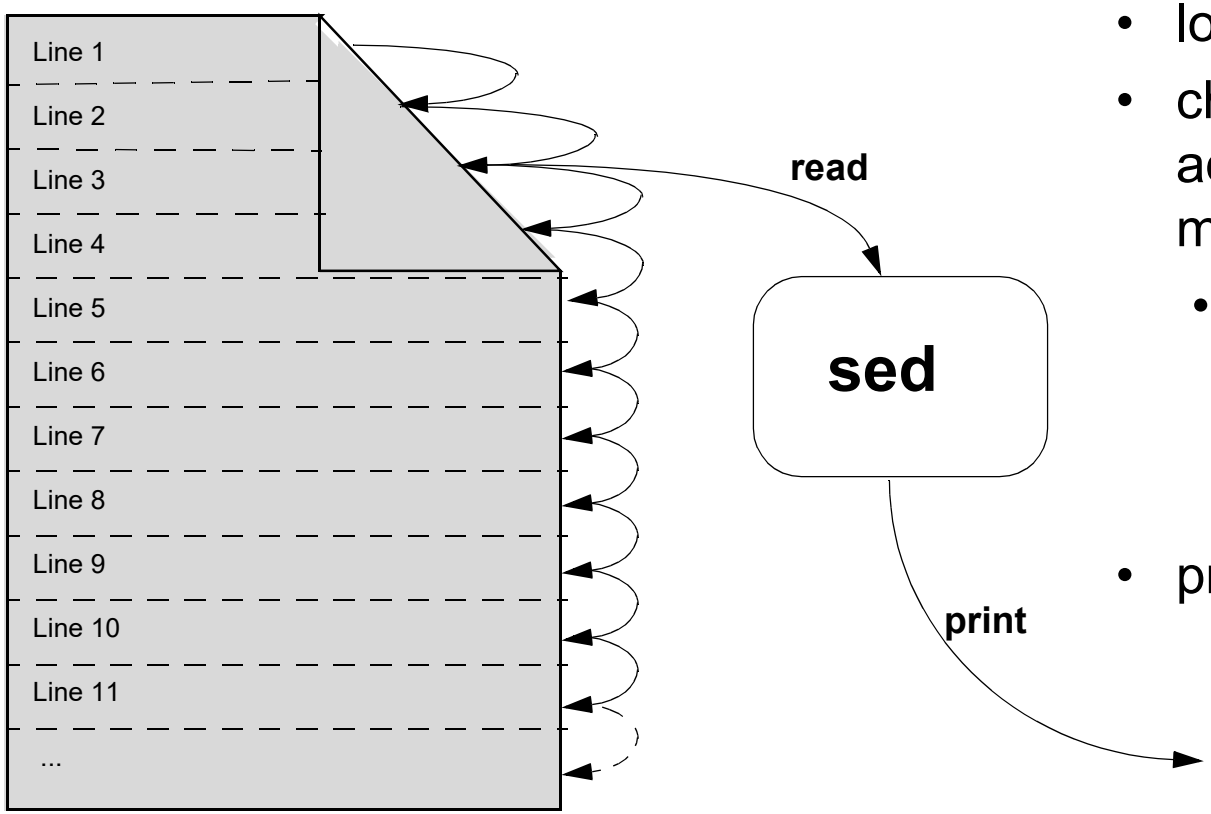


continue solving the case from the first exercise (Exercise II) ...

… still thrilling ;-)

(*) command line murders by Noah Veltman,
https://github.com/veltman/clmystery

sed Principle

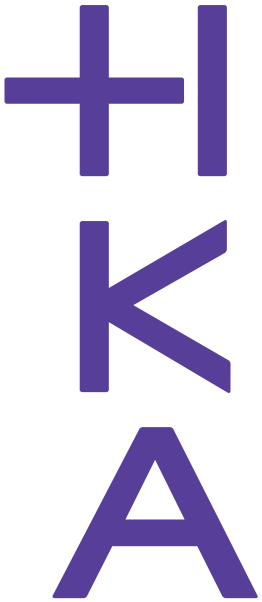


For every line:

- load line in pattern space
- check if optional address/address range matches
  - apply operation (insert, append, delete, substitude, change, print, ...) on pattern space
- print content of pattern space

String Substitution with sed

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



- sed - Stream Editor
- non interactiv, controlled by a script
- line oriented text processing
- A loop executes script commands on each matching (by address) input line
- short scripts are typically given as parameter, longer scripts as files (-f option)
- Possible operations: Insert, Substitude, Delete, Append, Change, Print, Delete
- Commands in script can take an optional *address*, specifying the line(s) to be performed.
- *Address* can be a a single line number or a regular expression
- *Address* can be an interval (start, stop)
- Default behavior: printing each processed line to STDOUT (suppress with: -n)

sed commands

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

s: substitude

- Replace in every line the first occurences of D with GER

```
sed 's/\bD\b/GER/' city.csv > city2.csv
```

- Replace all occurences of NULL in a line with \N (Inplace Substitution)

```
sed -i 's/\bNULL\b/\\N/g' city.cs
```

- Replace „Stuttgart“ with „Stuttgart am Neckar“ (extended regexp)

```
sed -E '/^Stuttgart/ s/^([^,]+)/\1 am Neckar/' city.csv
```

p: print (typically used with default printing behaviour off (-n option))

- print from line 10 to 20 (resp.: 5-10, 23, 56-71)

```
sed -n 10,20p city.csv
sed -n '5,10p;23p;56,71p' city.csv
```

i: insert

- Insert dataset about Karlsruhe at line 2

```
sed '2i Karlsruhe,D,"Baden Wuerttemberg",312005,49.0,6.8'
city.csv
```

d: delete

- delete the city Aachen (inplace)

```
sed -i '/^Aachen/ d' city.csv
```

- delete all empty lines

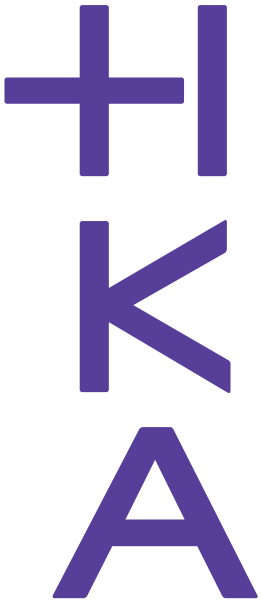
```
sed '/^ *$/d' The-Adventures-of-Tom-Sawyer.txt
```

- delete lines 2-10

```
sed '2,10d' city.csv
```

sed Examples

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



c: change

- Replace entry of Biel

```
sed '/^Biel,/ c Biel,CH,BE,53308,47.8,7.14' city.csv
```

a: append

- Underline each CHAPTER

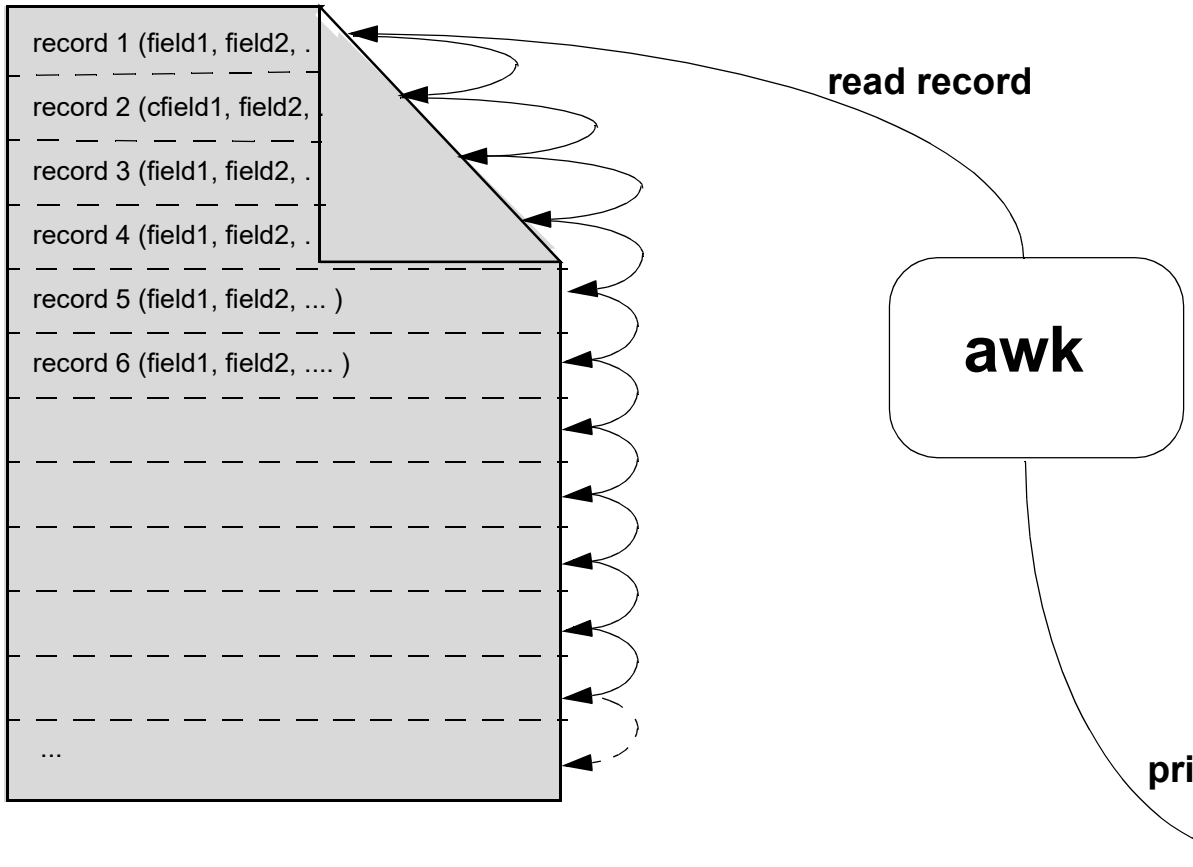
```
sed '/^CHAPTER/ a ------------' The-Adventures-of-Tom-Sawyer.txt
```

r: read file

- insert the content from file city-D.csv starting at line 3

```
sed '3 r city-D.csv' city.csv'
```

awk Principle



Perform start action (BEGIN)

For every record:

- check if condition holds
  - Perform action (can be any arbitrary code)

Perform end action (END)

awk

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- like sed, but with powerful programming language
- filter and report writer
- flexible record definition (i.e. line with columns, record with fields, ...)
- full programming language, suport for associative arrays
- structure: one or multiple *pattern { action }* blocks
- special BEGIN, END pattern match **before** the first record is read and **after** the last record is read
- Access to column values via $1, $2, ... variables ($0: whole record)
- Examples:

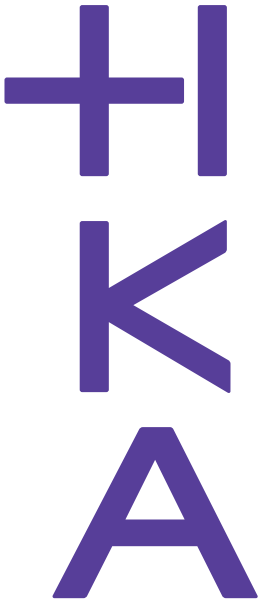
```
awk -F, '$3=="Bayern" && $4 < 1000000 { print $1, $4 }' city.csv
```

pattern

action

awk

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



- predefined variables
  - NF: number of fields
  - NR: number of records
  - FS: field separator (default: " ", same as -F from command line)
  - RS: record separator (default: \n)
  - ORS: output record separator
  - OFS: output field separator
  - FPAT: Field pattern (alternative way to specify a field instead of use of FS)
  - FILENAME: contains the file that is actually read

- More example:

  `https://www.smiffy.de/KDIR-2022/command-examples/awk`

awk example: multi-line input

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

```
Andreas Schmidt
KIT
Germany

Manolo Diaz
IARIA
USA

Fritz Laux
University Reutlingen
Germany
```

Input

```
Andreas Schmidt,KIT,Germany
Manolo Diaz,IARIA,USA
Fritz Laux,University Reutlingen,Ger-
many
```

Output

```
cat adress.txt | awk -f rec2csv.awk
```

```
BEGIN {
    FS="\n"
    RS="\n\n"
    OFS=","
    ORS="\n"
}
{
    $1=$1
    print $0
}
```

set input and output separators

give awk a hint that anything has changed

print whole record

awk

- Calculating average population

```
awk -F, -f average.awk  city.csv
```

```
# script: average.awk
BEGIN { sum = 0
        num = 0
}

$4 != "NULL" {
   sum += $4
   num++
}

END { print "Average population: "sum/num }
```

pattern

optional, because 0 is default value

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

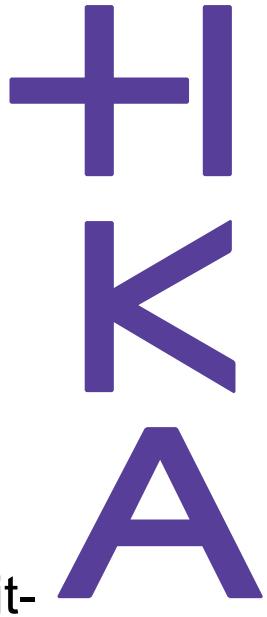
And again its time for crime* ...

As homework: You have solved the case, but there is room for improvement for future cases. So continue with Exercise III ...

(*) command line murders by Noah Veltman,
https://github.com/veltman/clmystery

Commands not Covered (not complete)

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



- **xargs:** build and execute command lines from standard input

```
grep -l Agassi bbcsport/tennis/*.txt | \
    xargs grep -l Federer
```

- **tr:** translate, squeeze, and/or delete characters from standard input, writing to standard output.

```
tr 'A-Z' 'a-z' < moby-dick.txt
```

- **paste:** merge lines of files

```
paste -d',' col1.txt col2.txt col3.txt > col_1-3.txt
```

- **find:** search for files in a directory hierarchy

```
find ./misc -name \*.txt -print
```

- **join:** join lines of two files on a common field

join Example

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- 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,Ber-
lin,356910,83536115
Djibouti,DJI,Djibouti,Dji-
bouti,22000,427642
Denmark,DK,Copenhagen,Denmark,43070,5249632
Algeria,DZ,Algiers,Algeria,2381740,29183032
```

```
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
```

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

Commands not Covered (not complete)

- **split:** split file by number of rows/bytes/records
- **csplit**: split file at given patterns (include/exclude semantics)

Examples:

https://www.smiffy.de/KDIR-2022/command-examples/split

Summary & Outlook

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Summary
  - Powerful filter and pipes architecture
  - Allows easy incremental development
  - Suitable for structured and unstructured data, ETL process
- Outlook
  - Utility make to handle dependencies between files
  - bash control flow elements like conditional execution, loops
  - bash functions
  - Seamless visualization using gnuplot
  - Easily extensible with own filters in any language

Further readings

- http://www.theunixschool.com/p/awk-sed.html
- Dale Dougherty, Arnold Robbinssed & awk, 2nd Edition UNIX Power Tools. O'Reilly, 2nd Edition 1997
- Arnold Robbins. Sed and Awk: Pocket Reference, 2nd Edition Paper-back – June , O'Reilly, 2002
- Ramesh Natarajan. sed and awk 101 hacks. http://www.thegeekstuff.com/sed-awk-101-hacks-ebook/
- gnuplot homepage: http://www.gnuplot.info/