Was'n das für'n Programm?
Man-page Lesungen für Neugierige

Marie Mann – m.mann@pengutronix.de
Björn Lässig – b.laessig@pengutronix.de

Wer sind wir?

- Administration bei Pengutronix
- Björn seit 2013
- Marie seit 2017

RTFM!

Read The Fine Manpage

MAN-page

Was’n das?

Action

Wer hat ‚man man' getippt?

Action

Wer hat ‚man man' gelesen?

MAN(1) Manual pager utils MAN(1)

NAME

man - an interface to the system reference manuals

SYNOPSIS

man [man options] [[section] page ...] ...

man -k [apropos options] regexp …

DESCRIPTION

man is the system's manual pager.#

[...]

SEE ALSO

apropos(1)

MAN(1) Manual pager utils MAN(1)

NAME

man - an interface to the system reference manuals

SYNOPSIS

man [man options] [[section] page ...] ...

man -k [apropos options] regexp …

DESCRIPTION

man is the system's manual pager.

[...]

SEE ALSO

apropos(1)

MAN(1) Manual pager utils MAN(1)

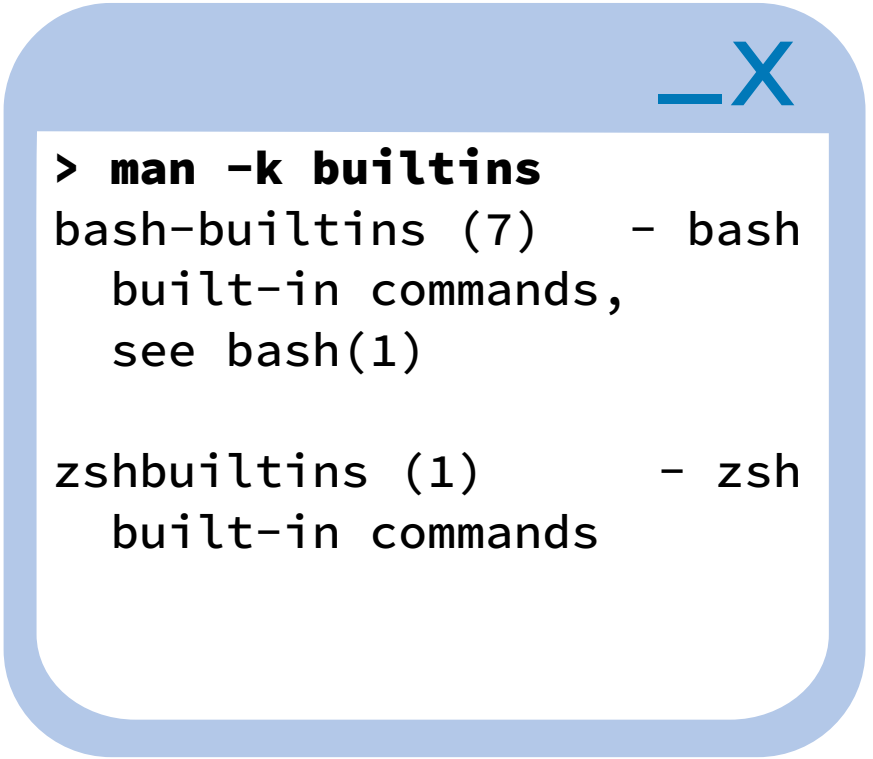
man -k keyword

Search the short descriptions and manual page names for the keyword […].

MAN(1) Manual pager utils MAN(1)

man -k keyword

Search the short descriptions and manual page names for the keyword […].



Agenda

man add-group bis

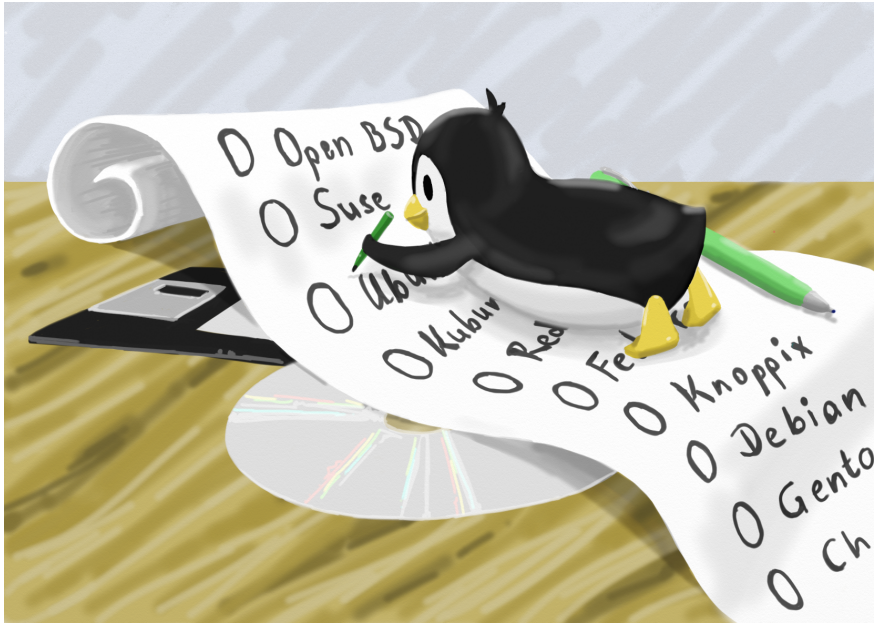
man zmore

Maries Problem

https://elearning.bgetem.de/

Maries Problem

https://elearning.bgetem.de/



copy-right by: Claudia Winkler, CLT 2013

Daten: Zertifikate im git

```
...
20220605_nma_Zertifikat.pdf
20231215_nma_Zertifikat.pdf
20231221_ftu_Zertifikat.pdf
20240107_cet_Zertifikat.pdf
```

[Page 1/87]

Mach mal ‘ne Shell auf!

Mach mal ‘ne Shell auf!

- `ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024`

- `sort ~/c.2024 userlist | uniq -u`

Mach mal ‘ne Shell auf!

- `ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024`

- `sort ~/c.2024 userlist | uniq -u`

LS(1) User Commands LS(1)

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs
(the current directory by default).

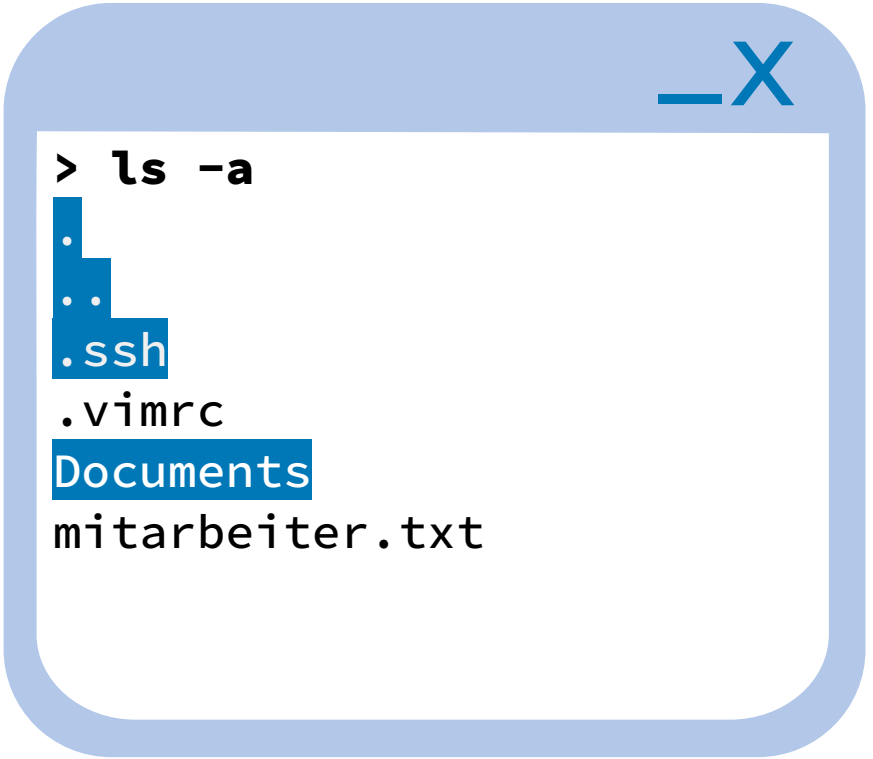
Mandatory arguments to long options
are mandatory for short options too.

Manual page ls(1) line 1/242

-a -d -x --author

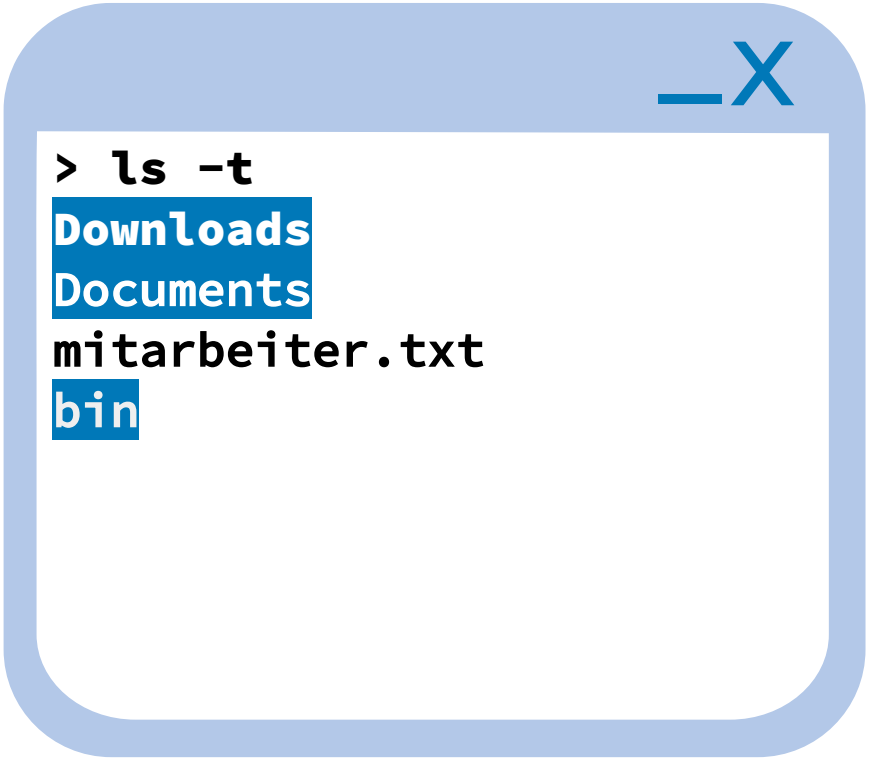
-a, --all

do not ignore entries starting with .



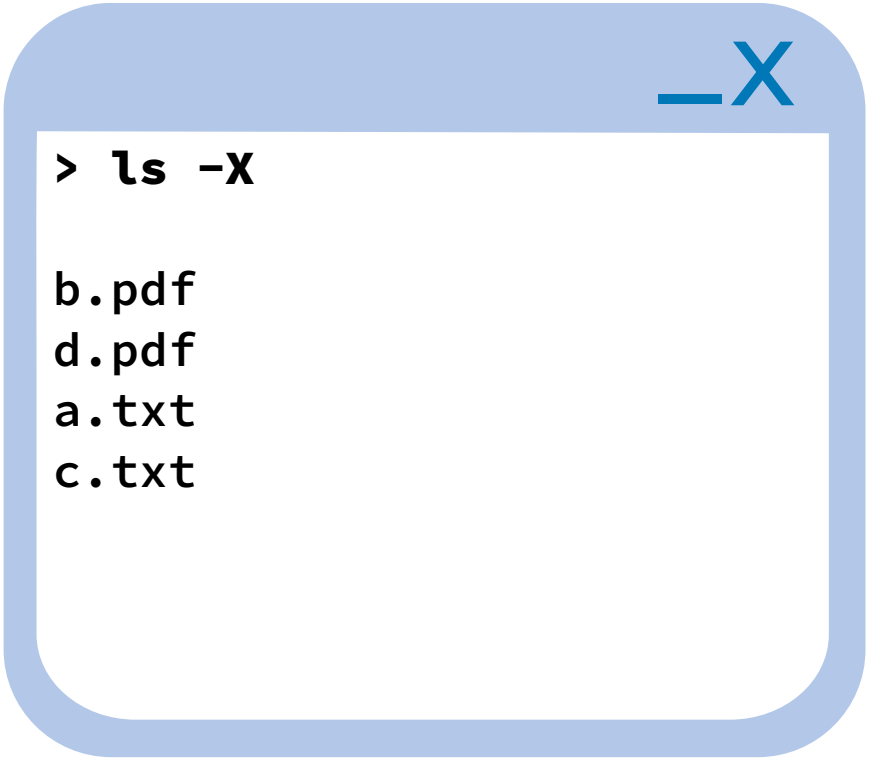
-t

sort by time, newest first



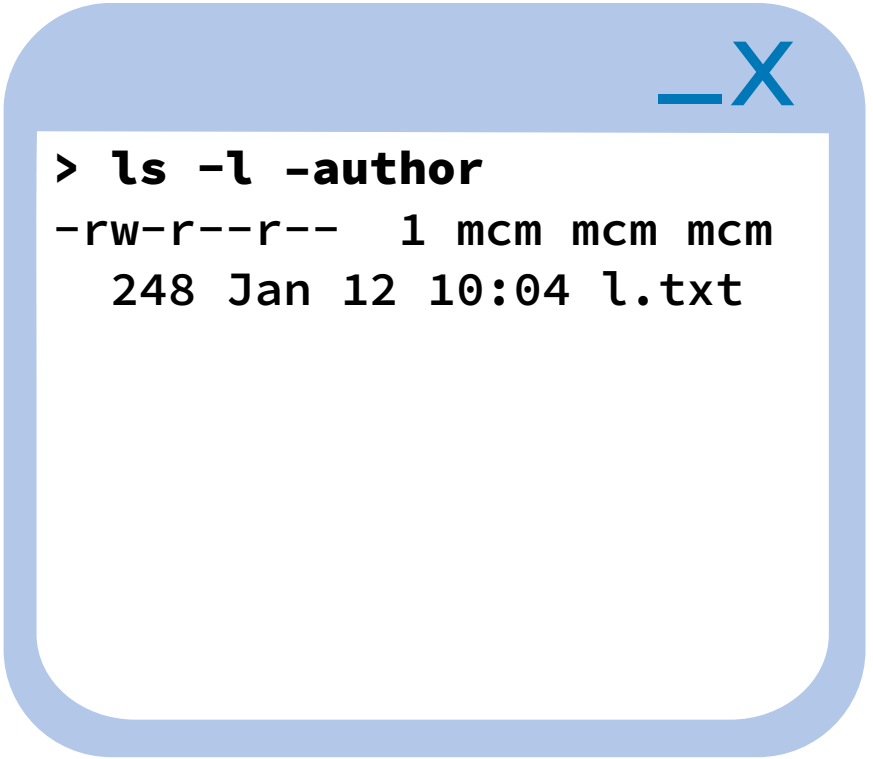
-X

sort alphabetically by entry extension



--author

with -l, print the author of each file



WTF?!

Nicht alle Optionen sind heute für dich nützlich!

Daten: Zertifikate im git

- `ls 202312* 2024*`

```
20231215_nma_Zertifikat.pdf
20231221_ftu_Zertifikat.pdf
20240107_cet_Zertifikat.pdf
```

Mach mal ‘ne Shell auf!

- `ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024`

- `sort ~/c.2024 userlist | uniq -u`

Mach mal ‘ne Shell auf!

```
▶  ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024
```

[stdout → stdin]

Ein Shell-Programm hat …

2 Ausgaben: → **stdout (1)**

→ stderr (2)

1 Eingabe → stdin

1 Umgebung aka Environment → eine Liste von Variablen:

```
VARIABLE_A=DiesUndDas
```

Ein Shell-Programm hat …

2 Ausgaben: → stdout (1)

→ **stderr (2)**

1 Eingabe → stdin

1 Umgebung aka Environment → eine Liste von Variablen:

```
VARIABLE_A=DiesUndDas
```

Ein Shell-Programm hat …

2 Ausgaben: → stdout (1)

→ stderr (2)

1 Eingabe → **stdin**

1 Umgebung aka Environment → eine Liste von Variablen:

```
VARIABLE_A=DiesUndDas
```

Ein Shell-Programm hat …

2 Ausgaben: → stdout (1)

→ stderr (2)

1 Eingabe → stdin

1 Umgebung aka Environment → **eine Liste von Variablen:**

```
VARIABLE_A=DiesUndDas
```

Mach mal ‘ne Shell auf!

```
ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024
```

[stdout → stdin]

Daten: Zertifikate im git

- `ls 202312* 2024*`

```
20231215_nma_Zertifikat.pdf
20231221_ftu_Zertifikat.pdf
20240107_cet_Zertifikat.pdf
```

Mach mal ‘ne Shell auf!

- `ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024`

- `sort ~/c.2024 userlist | uniq -u`

SED(1) User Commands SED(1)

NAME

sed - stream editor for filtering and transforming text

SYNOPSIS

sed [-V] [--version] [--help] [-n] [--quiet] [--silent]
[-l N] [--line-length=N] [-u] [--unbuffered]
[-E] [-r] [--regexp-extended]

DESCRIPTION

Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline).

xkcd #208: „Regular Expressions“

Mach mal ‘ne Shell auf!

```
▶  ls 202312* 2024* | sed 's/^.*-\
(...\)-.*/\1/' > ~/c.2024
```

das Ergebnis wird in die Datei geschrieben

Daten: Zertifikate im git

- `cat ~/c.2024`

```
nma
ftu
cet
```

Mach mal ‘ne Shell auf!

- ```
  ls 202312* 2024* | sed 's/^.*-\
  (...\)-.*/\1/' > ~/c.2024
  ```

- ```
  sort ~/c.2024 userlist | uniq -u
  ```

Daten: Zertifikate im git

- `sort ~/c.2024 userlist`

```
cet
cet
ftu
nma
nma
```

SORT(1) User Commands SORT(1)

NAME

sort - sort lines of text files

SYNOPSIS

sort [OPTION]... [FILE]...

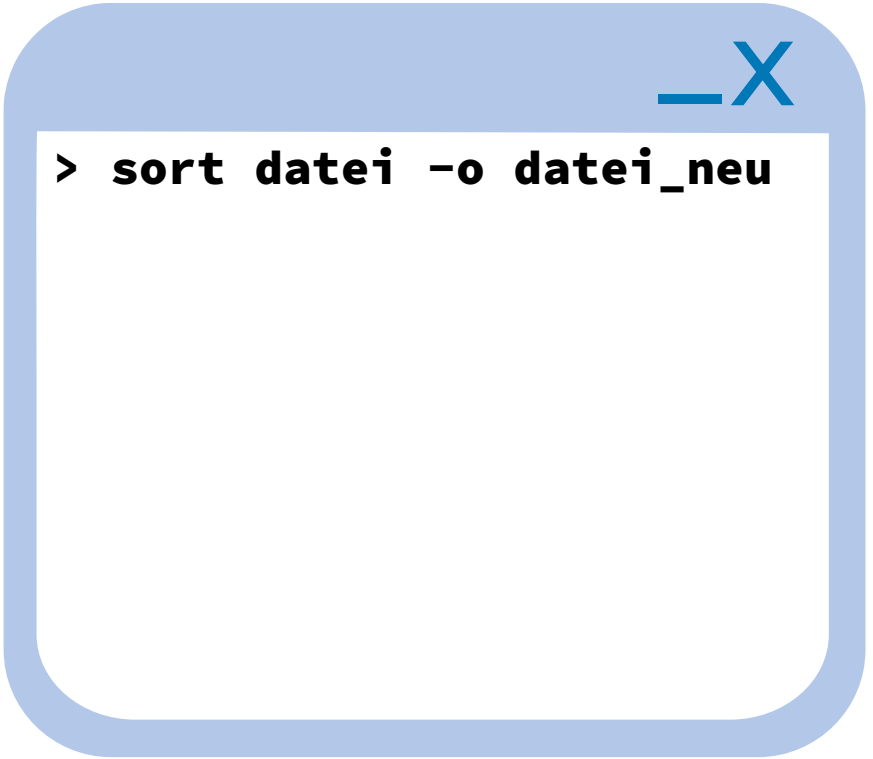
sort [OPTION]... --files0-from=F

DESCRIPTION

Write sorted concatenation of all FILE(s) to standard output.

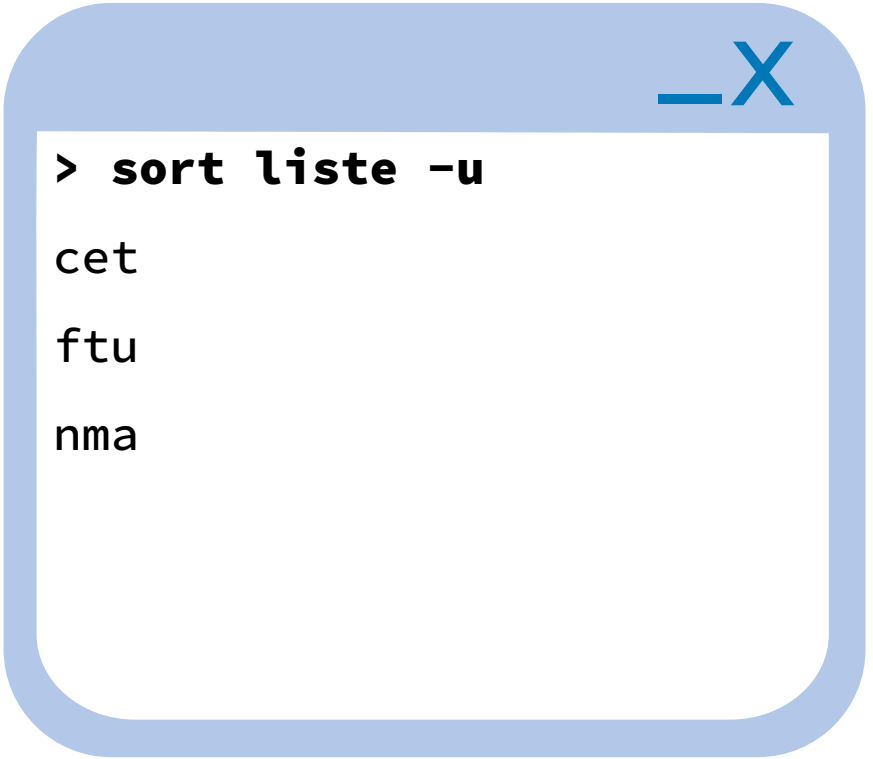
-o, --output=FILE

write result to FILE instead of standard output



-u, --unique

with -c, check for strict ordering; without -c, output only the first of an equal run



Daten: Zertifikate im git

- `sort ~/c.2024 userlist`

```
cet
cet
ftu
nma
nma
```

Mach mal ‘ne Shell auf!

- ```
  ls 202312* 2024* | sed 's/^.*-\(...\)-.*/\1/' > ~/c.2024
  ```

- ```
  sort ~/c.2024 userlist | uniq -u
  ```

UNIQ(1) User Commands UNIQ(1)

NAME

uniq - report or omit repeated lines

SYNOPSIS

uniq [OPTION]... [INPUT [OUTPUT]]

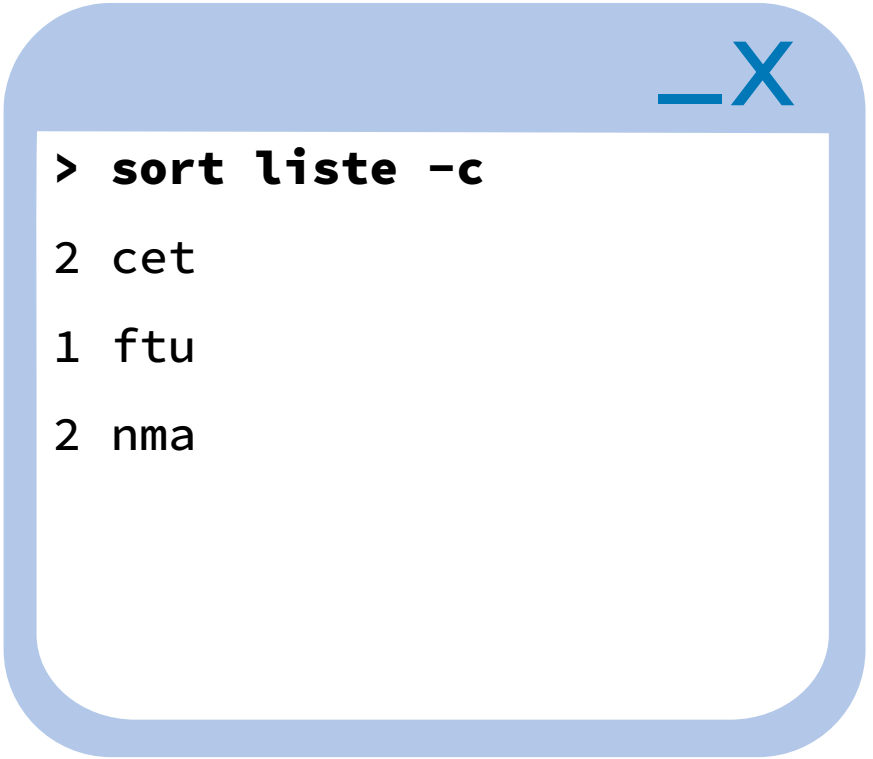
DESCRIPTION

Filter adjacent matching lines from INPUT (or standard input), writing to OUTPUT (or standard output).

With no options, matching lines are merged to the first occurrence.

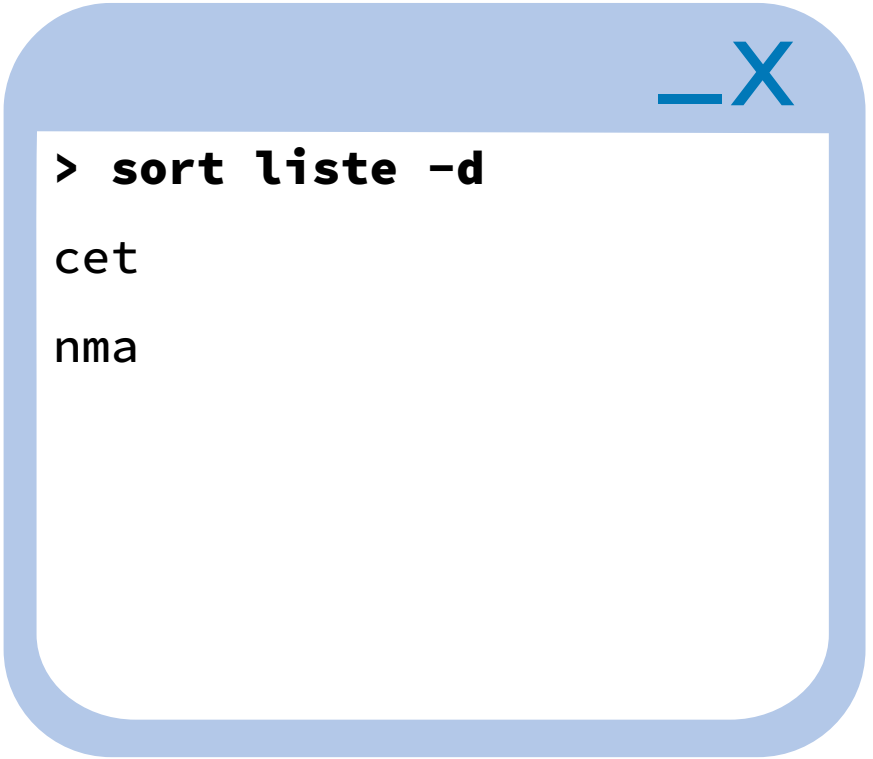
-c, --count

only print duplicate lines, one for each group



-d, --repeated

only print duplicate lines, one for each group



-u, --unique

only print unique lines

Daten: Zertifikate im git

- `sort ~/c.2024 userlist | uniq -u`

```
ftu
```

By the way …

… we are hiring

jobs.pengutronix.de

Weitere (nützliche) Tools?

NAME

wc - print newline, word, and byte counts for each file

SYNOPSIS

wc [OPTION]... [FILE]...

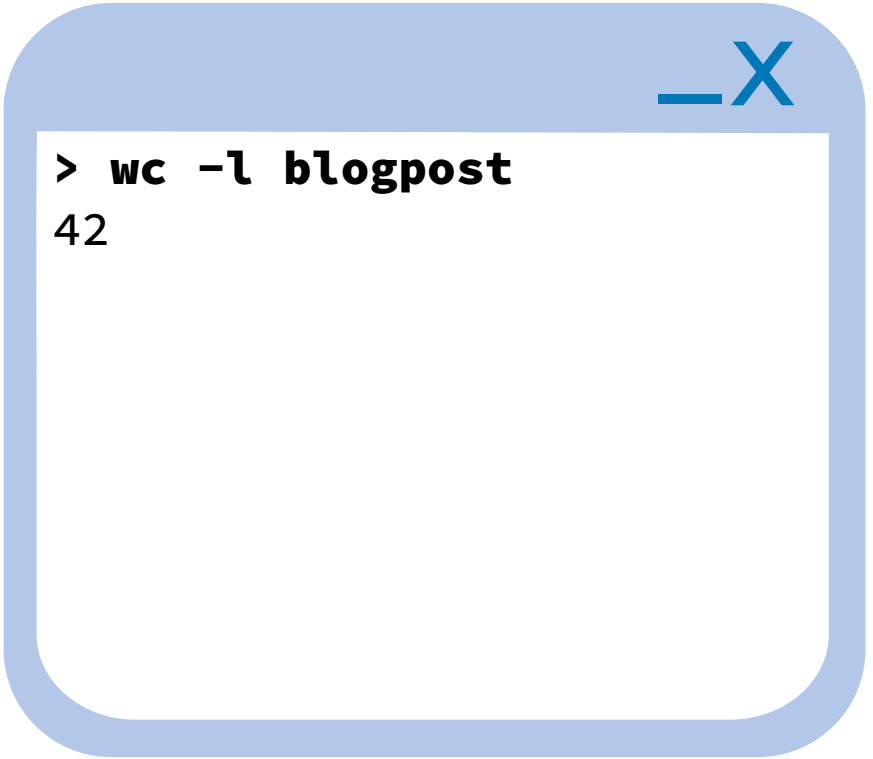
wc [OPTION]... --files0-from=F

DESCRIPTION

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. A word is a non-zero-length sequence of printable characters delimited by white space

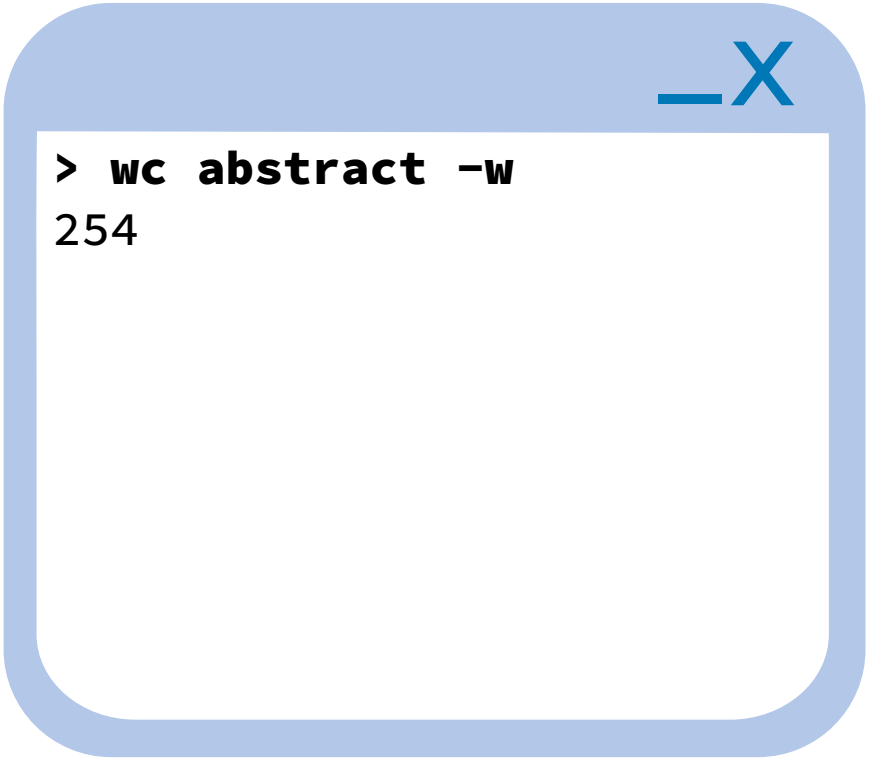
-l, --lines

print the newline counts



-w, --words

print the word counts



GREP(1) User Commands GREP(1)

NAME

grep, egrep, fgrep, rgrep - print lines that match patterns

SYNOPSIS

grep [OPTION...] PATTERNS [FILE...]

grep [OPTION...] -e PATTERNS ... [FILE...]

grep [OPTION...] -f PATTERN_FILE ... [FILE...]

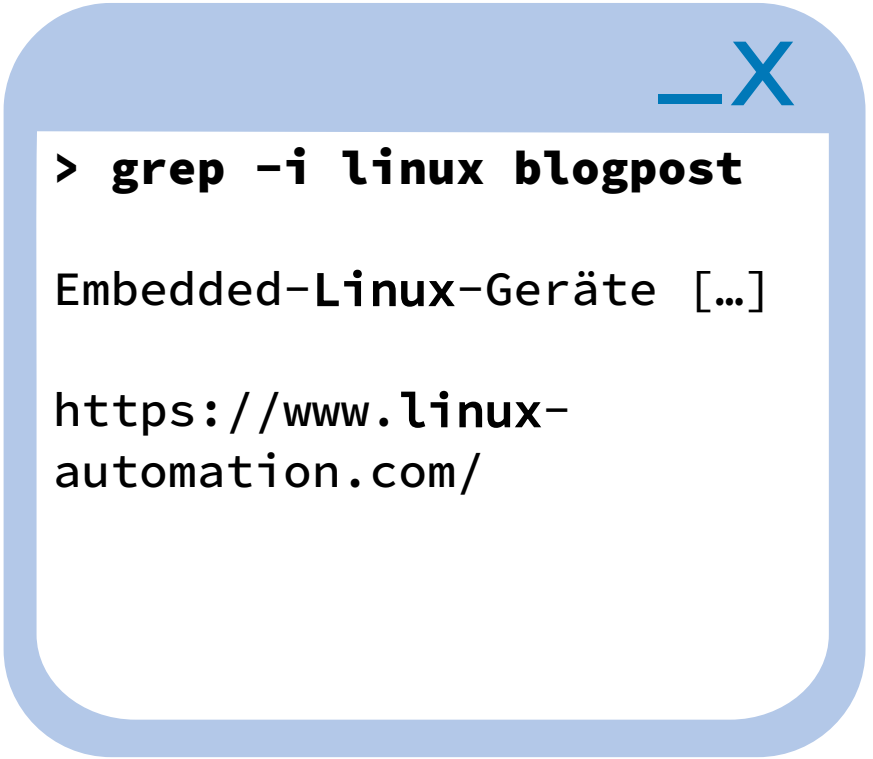
DESCRIPTION

grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by newline characters, and grep prints each line that matches a pattern.

-i, --ignore-case

Ignore case distinctions in patterns and input data, so that

characters that differ only in case match each other.



-C NUM, -NUM, --context=NUM

Print NUM lines of output context.

Places a line containing a group separator (--) between contiguous groups of matches. With the -o or -only-matching option, this has no effect and a warning is given.

_X

```
> grep num .vimrc -C 2

syntax enable

set number
set relativenumber

"set cursorline
```

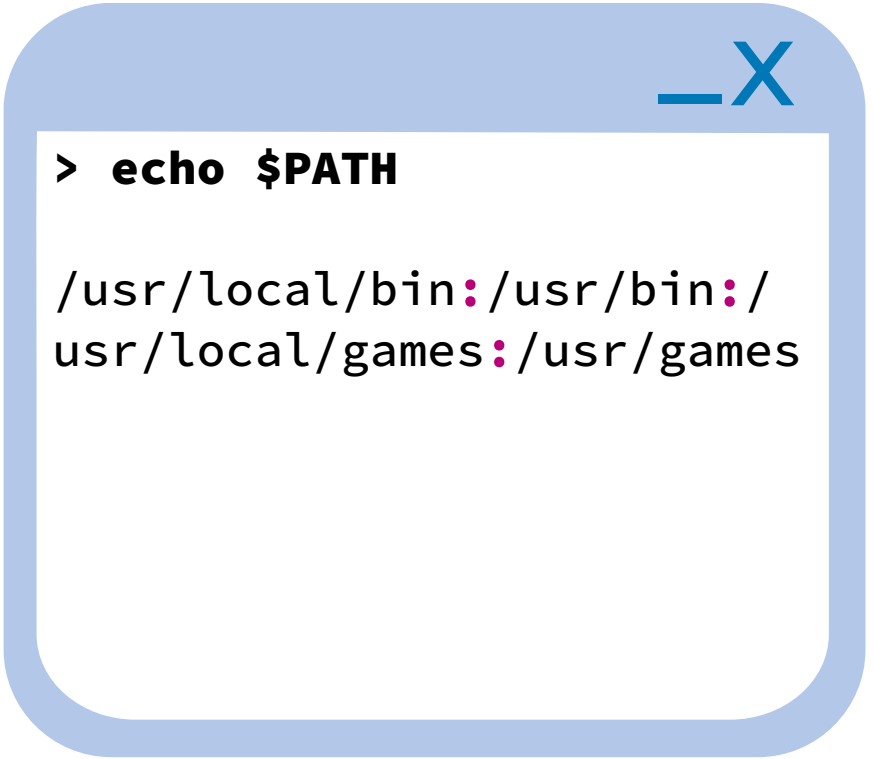
Wie finde ich Tools?

DASH(1) BSD General Commands

Path Search

[…]

The shell searches each entry in PATH in turn for the command. The value of the PATH variable should be a series of entries separated by colons. Each entry consists of a directory name. The current directory may be indicated implicitly by an empty directory name, or explicitly by a single period.



See Also

apropos(1), **groff**(1), **less**(1), **manpath**(1), **nroff**(1), **troff**(1), **whatis**(1), **zsoelim**(1), **manpath**(5), **man**(7), **catman**(8), **mandb**(8)

Documentation for some packages may be available in other formats, such as **info**(1) or HTML.

See Also

apropos(1), **groff**(1), **less**(1), **manpath**(1), **nroff**(1), **troff**(1), **whatis**(1), **zsoelim**(1), **manpath**(5), **man**(7), **catman**(8), **mandb**(8)

Documentation for some packages may be available in other formats, such as **info**(1) or HTML.

Was interessiert euch noch?

MAN(1) Manual pager utils MAN(1)

The table below shows the section numbers of the manual
followed by the types of pages they contain.

1	Executable programs or shell commands
2	System calls (functions provided by the kernel)
3	Library calls (functions within program libraries)
4	Special files (usually found in /dev)
5	File formats and conventions, e.g. /etc/passwd
6	Games
7	Miscellaneous (including macro packages and conventions)
8	System administration commands (usually only for root)
9	Kernel routines [Non standard]