

sort

COLLABORATORS

	<i>TITLE :</i> sort	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		August 27, 2022

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	sort	1
1.1	sort v1.80 Dokumentation - Inhalt	1
1.2	sort v1.80 Dokumentation - Disclaimer	2
1.3	sort v1.80 Dokumentation - Einführung	2
1.4	sort v1.80 Dokumentation - Anforderungen	3
1.5	sort v1.80 Dokumentation - History	3
1.6	sort v1.80 Dokumentation - Installation	6
1.7	sort v1.80 Dokumentation - Autor	6
1.8	sort v1.80 Dokumentation - Neu	7
1.9	sort v1.80 Dokumentation - Funktionen	7
1.10	sort v1.80 Dokumentation - Funktionen - a	9
1.11	sort v1.80 Dokumentation - Funktionen - A	10
1.12	sort v1.80 Dokumentation - Funktionen - bB	11
1.13	sort v1.80 Dokumentation - Funktionen - cC	12
1.14	sort v1.80 Dokumentation - Funktionen - d	14
1.15	sort v1.80 Dokumentation - Funktionen - eE	15
1.16	sort v1.80 Dokumentation - Funktionen - f	16
1.17	sort v1.80 Dokumentation - Funktionen - g	17
1.18	sort v1.80 Dokumentation - Funktionen - h	18
1.19	sort v1.80 Dokumentation - Funktionen - iI	20
1.20	sort v1.80 Dokumentation - Funktionen - jJ	21
1.21	sort v1.80 Dokumentation - Funktionen - l	23
1.22	sort v1.80 Dokumentation - Funktionen - nN	24
1.23	sort v1.80 Dokumentation - Funktionen - o	25
1.24	sort v1.80 Dokumentation - Funktionen - pP	27
1.25	sort v1.80 Dokumentation - Funktionen - qQ	28
1.26	sort v1.80 Dokumentation - Funktionen - r	29
1.27	sort v1.80 Dokumentation - Funktionen - sS	31
1.28	sort v1.80 Dokumentation - Funktionen - t	32
1.29	sort v1.80 Dokumentation - Funktionen - u	33

1.30	sort v1.80 Dokumentation - Funktionen - v	34
1.31	sort v1.80 Dokumentation - Funktionen - w W	36
1.32	sort v1.80 Dokumentation - Funktionen - x	37
1.33	sort v1.80 Dokumentation - Funktionen - y Y	38
1.34	sort v1.80 Dokumentation - Funktionen - z Z	39
1.35	sort v1.80 Dokumentation - Beispiele	41
1.36	sort v1.80 Dokumentation - Tips & Tricks	42
1.37	sort v1.80 Dokumentation - Known Bugs	42
1.38	sort v1.80 Dokumentation - Registrierung	42
1.39	sort v1.80 Dokumentation - Verfügbarkeit	43

Chapter 1

sort

1.1 sort v1.80 Dokumentation - Inhalt

Sort v1.80 25.11.98

DAS Sortierprogramm für Ihren Computer
Copyright © 1996-1998 Rüdiger Werner
Alle Rechte vorbehalten.

Disclaimer

Das Offizielle.

Einführung

Wofür nimmt man das Programm?

Anforderungen

Was brauche ich, um das Programm zu nutzen?

History

Die Geschichte.

Installation

Vor dem Sortieren.

Author

Über mich.

Neu

Welche Funktionen sind neu?

Funktionen

Welche Funktionen gibt es?

Beispiele

Einige Beispielaufrufe.

Tips & Tricks

Einige nützliche Tips.

Known Bugs
Known Bugs.

Registrierung
Wie bekomme ich die Vollversion.

Verfügbarkeit
Wo und für welche Computer
gibt es dieses Programm?

1.2 sort v1.80 Dokumentation - Disclaimer

Disclaimer

Das Programm wird ohne Garantie irgendeiner Art zur Verfügung gestellt. Alle Risiken, die mit der Verwendung dieses Programms verbunden sind, tragen Sie. Der Autor übernimmt keine Verantwortung bzw. Haftung für die Beschädigung oder den Verlust Ihrer Daten durch die Benutzung dieses Programms. Obwohl alles dafür getan wurde, um sicherzustellen, daß das Programm sicher funktioniert, kann der Autor das nicht 100% garantieren.

Die unregistrierte Version dieses Programms ist frei kopierbar. Es ist keine kommerzielle Nutzung ohne eine schriftliche Genehmigung des Autors gestattet. Die Dateien in diesem Archiv müssen in unveränderter Form belassen werden.

Die registrierte Version dieses Programms ist NICHT frei kopierbar.

Bitte lesen Sie die Anleitung sorgfältig, bevor Sie das Programm nutzen.

Inhalt

1.3 sort v1.80 Dokumentation - Einführung

Ich hatte einige sehr große Files zu sortieren. Zuerst ↔ versuchte ich es mit dem Workbench sort Befehl ... und brach nach einer Weile ab. Der Computer schien sich aufgehängt zu haben, nichts passierte mehr.

Dann versuchte ich es mit Sortierprogrammen vom Aminet. Aber ich konnte keins finden, das schnell genug war und zusätzlich Änderungen am File vornehmen konnte.

So hatte ich die Möglichkeit, alles zu vergessen oder selbst ein Programm zu schreiben, das meinen Anforderungen genügte. Wie man sieht, habe ich mich zum letzteren entschlossen.

Die Grundfunktion von sort ist natürlich, ASCII-files alphabetisch oder numerisch zu sortieren. Aber das ist nicht alles. Ich habe viele Funktio-

nen implementiert, um die Files zusätzlich zu modifizieren.

Eine ausführliche Beschreibung aller Optionen ist
hier
zu finden.

Das Programm wird von der Shell gestartet.

Inhalt

1.4 sort v1.80 Dokumentation - Anforderungen

Es wird ein Computer gebraucht (am besten ein Amiga :-) mit ←
möglichst
viel Speicher. Und natürlich Files zum Sortieren.

Das Programm ist für verschiedene
Computersysteme
erhältlich.

Inhalt

1.5 sort v1.80 Dokumentation - History

Version 1.00 - 1.20

interne Versionen
auf PC programmiert, da kein richtiger C-Compiler für den Amiga zu finden
war.

Version 1.21
auf AMIGA portiert

Version 1.22
einen Bug entfernt, der das Programm bei Zeilen >1024 Zeichen zum Absturz
brachte. Wenn die Zeile länger ist, wird der Rest abgeschnitten.

Version 1.23
-n|N Option hinzugefügt
das Programm in Englisch übersetzt, den Code optimiert.

Version 1.24
einige Verbesserungen bei der Switchauswertung, einige Widersprüche
entfernt

Version 1.25
komplett neue Laderoutine, braucht weniger Speicher

Version 1.30
verbesserter Sortieralgorithmus
jetzt doppelte Geschwindigkeit beim Sortieren

Version 1.31

das Löschen von Zeilen optimiert, dadurch ist das Sortieren schneller.
einige Farben hinzugefügt

Version 1.32

optimierte Switchauswertung
-j|J Switch hinzugefügt; jetzt können beliebig viele Files zusammengefügt
werden, lediglich begrenzt durch den Arbeitsspeicher
ein paar kleinere Bugs entfernt

Version 1.33

-f[y] und -l[y] Switch hinzugefügt
jetzt ist es möglich, ein File in einem Bereich zu sortieren bzw. zu
modifizieren.

Version 1.34

einen Bug entfernt
durch einige Veränderungen in der Sortieroutine hat sich die
Geschwindigkeit um etwa 10% erhöht.

Version 1.35

-s Switch hinzugefügt

Version 1.36

-P Switch hinzugefügt
die -s Option kann nun auch auf einen Bereich angewendet werden
Sortiergeschwindigkeit wieder etwas erhöht.

Version 1.37

-c Switch hinzugefügt
jetzt können Files nach einer vorgegebenen Spalte sortiert werden einen
Fehler in der Anzeige entfernt, beim Zusammenfügen von Files wurde genau
ein File weniger verlangt wurde, als gewünscht.

Version 1.38

einen Check hinzugefügt, der testet, ob das Ausgabefile schon vorhanden ist

Version 1.38a

Bugfix.
Die -o Option funktionierte nicht mehr durch den Ausgabefiletest

Version 1.41

die Laderoutine verändert, ein "Pointerbuffering" in den Sortieralgo-
rithmus implementiert. Dadurch verkürzt sich die Sortierzeit um 50-75% im
Vergleich zur Version 1.38, abhängig vom File, das sortiert werden soll.
das Programm verwendet jetzt die Mempoools Routinen © by Jochen Wiedmann
zur Speicherverwaltung, die Zeit für Allokieren und Freigeben von Speicher
hat sich auf 2 (zwei) % reduziert.

Version 1.42

-c Option verändert
jetzt kann eine Länge vorgegeben werden, die zum Sortieren nach einer
Spalte verwendet wird.
-v Switch hinzugefügt
wenn diese Option gesetzt ist, werden keine Meldungen außer Fehler
ausgegeben.
-S Switch hinzugefügt

ein einzelnes Zeichen kann nun durch ein anderes ersetzt werden

Version 1.43

Bugfix

Die join Option funktionierte nicht mehr richtig und die Meldungen sahen etwas komisch aus

-C Switch hinzugefügt

im Prinzip das gleiche wie -c, aber wenn -E / -N gesetzt ist, wird die Spalte, nach der sortiert wird, auch zum Finden der Zeilen zum Löschen verwendet.

Version 1.44

-i Switch hinzugefügt

einen neuen Algorithmus zum Vorbereiten der Liste zum Sortieren programmiert Die Sortiergeschwindigkeit hat sich um das sechsfache erhöht.

Version 1.45

neue Laderoutine für die Amigaversion programmiert

Braucht nur noch die Hälfte der Zeit zum Laden und praktisch keine Zeit mehr zum "Aufräumen" beim Programm beenden.

Speicherbedarf um etwa 30% reduziert

mempools werden nicht mehr benötigt

Version 1.46

einen Bug entfernt, der evtl. bei Zeichen mit einem ASCIIwert <32 auftrat keine Begrenzung der Zeilenlänge mehr (Amiga version)

Version 1.47

einen Rechenfehler in der Zeilenberechnung entfernt

Version 1.48

-q|Q Switch hinzugefügt

jetzt kann ein File nach einem Suchkriterium in zwei aufgeteilt werden

Version 1.49

ein paar kleinere Verbesserungen

Version 1.50

-t Switch hinzugefügt

dabei werden Leerzeichen und Tabs am Ende einer Zeile entfernt

-h Switch hinzugefügt

mit diesem Switch werden eine vorgegebene Zahl von Headerzeilen in den zu sortierenden Files ignoriert. Es wird nach dem ersten und allen weiteren File unterschieden.

-i Switch verändert

jetzt können mehrere Strings zum Ignorieren vorgegeben werden

-I Switch hinzugefügt

wie -i, aber es wird in zwei Passes sortiert. Zuerst werden die vorgegebenen Strings nicht ignoriert, dann werden alle doppelten Zeilen gelöscht, dann wird ein zweites Mal sortiert, diesmal werden alle vorgegebenen Strings ignoriert.

-E|N Switch verändert

die -t Option wird beim Setzen dieser Switches jetzt automatisch gesetzt Sortiergeschwindigkeit erhöht

Version 1.60

-b|B Switch hinzugefügt. Das Program verfügt jetzt auch über eine

numerische Sortiermöglichkeit

-I Switch verändert. Wird diese Option ausgewählt, wird jetzt auch die -E Option gesetzt.

- die System Schlüsselwörter werden jetzt auch verarbeitet.

Version 1.61

Bugfix

die Geschwindigkeit beim Schreiben des zweiten Files (-q|Q) erhöht

Version 1.62

Bugfix

Version 1.70

-w Switch hinzugefügt. Zeigt Zeilen, die gelöscht werden

-z Switch hinzugefügt. Erlaubt mehrere Zeilen pro Datensatz

Routine zum Auswerten der Switches überarbeitet.

Version 1.80

-A Switch hinzugefügt. Hängt das outfile an eine evtl. bereits vorhandene Datei an.

-u Switch hinzugefügt. Löscht Zeilen, die vollständig in einer anderen enthalten sind.

-y|Y Switch hinzugefügt. Die Sortierspalte kann jetzt variabel nach einem Suchmuster festgelegt werden. Diese Option arbeitet auch mit der -i|I Option zusammen.

-Z Switch hinzugefügt. Damit können auch binäre (!) Dateien mit ASCII-Anteil sortiert werden.

Voraussetzung ist eine konstante Datensatzlänge.

Inhalt

1.6 sort v1.80 Dokumentation - Installation

Das ist ganz einfach:

Kopieren Sie einfach das Programm in das C: Verzeichnis oder in irgendein Verzeichnis, das aber im Pfad stehen sollte.

Dafür können Sie das Installscript verwenden, wenn Sie den Commodore Installer auf Ihrem System haben.

Das Programm kann auch resident gemacht werden, wenn genügend Arbeitsspeicher zur Verfügung steht.

Inhalt

1.7 sort v1.80 Dokumentation - Autor

Rüdiger Werner

E-Mail: wernerr@iee1.et.tu-dresden.de

URL: <http://www.iee.et.tu-dresden.de/~wernerr/>
<http://www.iee.et.tu-dresden.de/~wernerr/amiga.html>
<http://www.iee.et.tu-dresden.de/cgi-bin/cgiwrap/search.sh>

Snail-Mail: Rüdiger Werner
Wintergartenstraße 14
01307 Dresden
GERMANY

Inhalt

1.8 sort v1.80 Dokumentation - Neu

Was ist neu?

- das Programm versteht die
 Schlüsselwörter
 des
System-Sortierprogramms (FROM/A, TO/A, COLSTART/K, NOCASE/S, NUMERIC/S)
 - mit der A Option überschreibt das Programm eine evtl. vorhandene Datei
nicht, sondern hängt die Zeilen hinten an
 Infos
 - mit der u Option werden Zeilen gelöscht, die komplett in einer
anderen enthalten sind.
 Infos
 - die y|Y Option legt die Sortierspalte nach einem Suchmuster fest ↔
 .
- Diese Option arbeitet auch mit der -i|I Option zusammen.
 Infos
 - mit der Z Option können auch binäre (!) Dateien mit ASCII Anteil
sortiert werden. Voraussetzung ist eine konstante Datensatzlänge.
 Infos

Inhalt

1.9 sort v1.80 Dokumentation - Funktionen

- a
 - A
 - b|B
 - c|C
 - d
-

-e|E
-f
-g
-h
-i|I
-j|J
-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z

die System Schlüsselwörter

FROM/A, TO/A,
COLSTART/K
,
NOCASE/S
,
NUMERIC/S

können statt der Switches angegeben werden.

Achtung: In der unregistrierten Version sind
die a, c|C, e, g, n, o, v, w und x - Option verfügbar.

Inhalt

1.10 sort v1.80 Dokumentation - Funktionen - a

```
-a  
-A  
-b|B  
-c|C  
-d  
-e|E  
-f  
-g  
-h  
-i|I  
-j|J  
-l  
-n|N  
-o  
-p|P  
-q|Q  
-r  
-s|S  
-t  
-u  
-v  
-w|W  
-x  
-y|Y  
-z|Z  
-a Sortieren des Files in aufsteigender Reihenfolge (default)
```

Inhalt

|<
Funktionen
>>

1.11 sort v1.80 Dokumentation - Funktionen - A

-a
-A
-b|B
-c|C
-d
-e|E
-f
-g
-h
-i|I
-j|J
-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y

-z|Z
-A wenn die Datei, in die die Zeilen geschrieben werden sollen,
bereits vorhanden ist, werden die Zeilen an diese Datei angehängt

Inhalt

<<

Funktionen

>>

1.12 sort v1.80 Dokumentation - Funktionen - b|B

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u
-v
-w|W
-x
-y|Y
-z|Z

-b dieser Switch schaltet die numerische Sortierfunktion an. Alle Zeilen, die mit einer Zahl beginnen, werden nach diesen sortiert und stehen, sollten Zeilen ohne Nummern auftauchen, vor diesen. Die Zeilen ohne Nummern werden alphabetisch sortiert.

entspricht dem Schlüsselwort NUMERIC/S

-B wie oben

Der Unterschied besteht darin, daß alle Zeilen, die nicht mit einer Nummer beginnen, gelöscht werden.

Inhalt

<<

Funktionen

>>

1.13 sort v1.80 Dokumentation - Funktionen - c|C

-a
-A
-b|B
-c|C
-d
-e|E
-f
-g
-h
-i|I
-j|J

-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z
-c[z,l] dieser Switch legt die Spaltennummer <z> fest, ab der ↔
sortiert wird.

<l> gibt an, wieviele Zeichen zum Sortieren verwendet werden sollen. Das ist z.B. nützlich zum Sortieren der Aminet RECENT files nach dem Typ. (-c20,10)

Wenn 'l' nicht angegeben wird, wird danach gefragt, wenn eine '0' gegeben wird, (-c10,0) wird die Länge automatisch auf die Zeilenlänge gesetzt.

Wenn doppelte Zeilen gelöscht werden, wird ab dem Anfang der Zeile verglichen.

entspricht dem Schlüsselwort COLSTART/K

-C[z,l] wie oben

Der Unterschied ist, daß der Vergleich der Zeilen zum Löschen ab der gegebenen Spalte beginnt.

Der Vergleich wird immer bis zum Ende der Zeilen geführt, die Anzahl der Zeichen zum Sortieren spielt keine Rolle.

Inhalt

<<

Funktionen

>>

1.14 sort v1.80 Dokumentation - Funktionen - d

```
-a
-A
-b|B
-c|C
-d
-e|E
-f
-g
-h
-i|I
-j|J
-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z
-d Sortieren des Files in absteigender Reihenfolge
```

Inhalt

<<

Funktionen

>>

1.15 sort v1.80 Dokumentation - Funktionen - e|E

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-e alle Leerzeilen werden gelöscht.

-E alle Leerzeilen und alle doppelten Zeilen werden nach dem Sortieren gelöscht.

Beachten Sie daß Leerzeichen und Tabs am Zeilenende entfernt werden.

Inhalt

<<

Funktionen

>>

1.16 sort v1.80 Dokumentation - Funktionen - f

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z

-f[y] gibt die erste Zeile des Arbeitsbereiches an.

Wenn diese Option angegeben ist, werden ALLE Operationen nur in diesem Bereich ausgeführt.

Es ist nicht zwingend, eine letzte Zeile festzulegen.

Wenn <y> größer ist, als die Gesamtanzahl der Zeilen, wird eine Fehlermeldung ausgegeben und das Programm abgebrochen.

Inhalt

<<

Funktionen

>>

1.17 sort v1.80 Dokumentation - Funktionen - g

-a
-A
-b|B
-c|C
-d
-e|E
-f
-g
-h
-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-g Ist diese Option angegeben, werden Groß- und Kleinbuchstaben beim Sortieren nicht unterschieden.

entspricht dem Schlüsselwort NOCASE/S

Inhalt

<<

Funktionen

>>

1.18 sort v1.80 Dokumentation - Funktionen - h

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-h[x,y] Diese Option entfernt eine vorgegebene Anzahl von Zeilen am Anfang eines Files.

Das ist nützlich, wenn mehrere Files gleichen Aufbaus zusammengefügt werden sollen und der Header gar nicht oder nur einmal und am Anfang des sortierten Files auftauchen soll.

wenn <x> oder <y> nicht angegeben werden, wird nicht danach gefragt, der Wert wird mit 0 angenommen,
<x> gibt die Zahl der entfernenden Zeilen für das erste File und <y> die für alle weiteren Files an, die zusammengefügt werden sollen.

Inhalt

<<

Funktionen

>>

1.19 sort v1.80 Dokumentation - Funktionen - i|I

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y


```
-z|Z  
-i[<filename | str]
```

der/die angegebene(n) String(s) werden beim Sortieren ignoriert, wenn sie am Zeilenanfang auftreten. Sie werden NICHT entfernt. Wenn im zu ignorierenden String Leerzeichen sind, ist es empfehlenswert, in der Kommandozeile nur den Switch anzugeben und den String auf Anfrage des Programms einzugeben. Für das Löschen von doppelten Zeilen wird auch hier die gesamte Zeile zum Vergleich herangezogen.

Wird beim Aufruf kein String angegeben, wird wie bisher danach gefragt. Neu ist, daß jetzt mehrere Strings angegeben werden können, jeder einzelne durch ein RETURN abgeschlossen. Ist die Eingabe beendet, einfach nochmal Return drücken.

Ist das erste Zeichen nach dem -i ein '<', so wird die nachfolgende Zeichenkette als Filename interpretiert und versucht, die Liste aus diesem File einzulesen. Das ist nützlich, wenn viele Strings anzugeben sind und das bei mehreren zu sortierenden Files.

Der Aufbau dieses Files ist ein String pro Zeile, egal wie viele Leerzeichen oder sonstiges er enthält. Wenn eine Leerzeile auftritt, wird das Lesen beendet

folgt dem '<' nichts, sind alle Strings von Hand einzugeben.

```
-I[<filename | str]  
siehe -i[string]  
der Unterschied ist, daß das File in zwei Durchgängen sortiert wird. Zuerst wird das File normal, ohne Beachtung der Ignorestrings sortiert. Dann werden alle doppelten Zeilen entfernt. Im zweiten Durchgang werden dann beim Sortieren die angegebenen Strings ignoriert. Das stellt sicher, daß auch wirklich alle doppelten Zeilen entfernt werden. Natürlich braucht das mehr Zeit, da zweimal sortiert werden muß.
```

Wenn diese Option gesetzt wird, so wird automatisch die -E Option gesetzt.

Diese Option arbeitet auch mit der -y Option zusammen. Dabei wird zuerst die Sortierspalte nach Muster gesucht und anschließend die -i|I Option (ggf. mit Liste) ausgeführt.

Inhalt

<<

Funktionen

>>

1.20 sort v1.80 Dokumentation - Funktionen - j|j

-a
-A
-b|B
-c|C
-d
-e|E
-f
-g
-h
-i|I
-j|J
-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z
-j[x] fügt <x> Files zusammen und sortiert sie. Das resultierende ↔
File
wird nach <outfile> geschrieben.

-J[x] fügt <x> Files ohne Sortieren zusammen. Das resultierende File
wird nach <outfile> geschrieben.

Ist die A Option gesetzt, wird das bereits vorhandene File nicht überschrieben, sondern die ausgegebenen Zeilen werden an dieses File angehängt.

Inhalt

<<

Funktionen

>>

1.21 sort v1.80 Dokumentation - Funktionen - I

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-l[y] gibt die letzte Zeile des Arbeitsbereiches an.

Wenn diese Option angegeben ist, werden ALLE Operationen nur in diesem Bereich ausgeführt.

Es ist nicht zwingend, eine erste Zeile festzulegen.

Wenn <y> größer ist, als die Gesamtanzahl der Zeilen, wird der Wert automatisch auf die letzte Zeile des Files gesetzt.

wird <y> nicht gegeben, so wird danach gefragt.

Inhalt

<<

Funktionen

>>

1.22 sort v1.80 Dokumentation - Funktionen - n|N

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z
-n das File wird nicht sortiert, es werden nur Leerzeilen ←
entfernt

-N das File wird nicht sortiert, es werden nur Leerzeilen und
doppelte Zeilen entfernt
Die Verwendung dieser Option ist nur sinnvoll, wenn das File
bereits sortiert ist.
Beachten Sie, daß Leerzeichen und Tabs am Zeilenende entfernt
werden.

Inhalt

<<

Funktionen

>>

1.23 sort v1.80 Dokumentation - Funktionen - o

-a
-A
-b|B
-c|C

-d
-e|E
-f
-g
-h
-i|I
-j|J
-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z

-o das sortierte File wird unter dem gleichen Namen wie das
(erste) eingelesene File gespeichert.

ACHTUNG!!!
IST DAS ORIGINALFILE EINMAL ÜBERSCHRIEBEN,
KANN ES NICHT WIEDERHERGESTELLT WERDEN!!!

Inhalt

<<

Funktionen

>>

1.24 sort v1.80 Dokumentation - Funktionen - p|P

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-p[*str*] alle Zeilen die das Suchmuster *<str>* nicht enthalten,

werden aus dem File gelöscht.

`-P[str]` alle Zeilen die das Suchmuster `<str>` enthalten,
werden aus dem File gelöscht.

BEMERKUNG: wenn Zeichen wie z.B. `"` angegeben werden, ist es ratsam, diese Option als letzte in der Kommandozeile anzugeben. Der Grund ist, daß die einem `"` folgenden Argumente nicht mehr erkannt werden.

Die Verwendung `-p|P` und `-q|Q` Option ist gleichzeitig möglich. In diesem Fall werden zuerst die Zeilen der `-p|P` Option entsprechend gelöscht und nach dem Sortieren wird das File in die zwei verschiedenen Files geteilt, entsprechend der `q|Q` Option.

Inhalt

<<

Funktionen

>>

1.25 sort v1.80 Dokumentation - Funktionen - `q|Q`

`-a`

`-A`

`-b|B`

`-c|C`

`-d`

`-e|E`

`-f`

`-g`

`-h`

`-i|I`

`-j|J`

`-l`

`-n|N`

`-o`

`-p|P`

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-q[*str*] alle Zeilen die das Suchmuster *<str>* nicht enthalten, werden in ein extra File mit dem Namen *<outfile>.ohne* geschrieben, alle Zeilen die dieses Muster enthalten, in *<outfile>*.

-Q[*str*] alle Zeilen die das Suchmuster *<str>* enthalten, werden in ein extra File mit dem Namen *<outfile>.mit* geschrieben, alle Zeilen die dieses Muster nicht enthalten, in *<outfile>*.

BEMERKUNG: wenn Zeichen wie z.B. " angegeben werden, ist es ratsam, diese Option als letzte in der Kommandozeile anzugeben. Der Grund ist, daß die einem " folgenden Argumente nicht mehr erkannt werden.

Die Verwendung -p|P und -q|Q Option ist gleichzeitig möglich. In diesem Fall werden zuerst die Zeilen der -p|P Option entsprechend gelöscht und nach dem Sortieren wird das File in die zwei verschiedenen Files geteilt, entsprechend der q|Q Option.

Inhalt

<<

Funktionen

>>

1.26 sort v1.80 Dokumentation - Funktionen - r

-a

-A

-b|B

-c|C

-d
-e|E
-f
-g
-h
-i|I
-j|J
-l
-n|N
-o
-p|P
-q|Q
-r
-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z

-r das angegebene File wird nur eingelesen und rückwärts wieder nach <outfile> geschrieben. Andere Optionen werden nicht berücksichtigt.

Inhalt

<<

Funktionen

>>

1.27 sort v1.80 Dokumentation - Funktionen - s|S

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-s[c] das Zeichen <c> wird aus den Zeilen im File/Bereich entfernt

-S[cd] das Zeichen <c> wird im File/Bereich durch <d> ersetzt

BEMERKUNG: wenn Zeichen wie z.B. " angegeben werden, ist es ratsam, diese Option als letzte in der Kommandozeile anzugeben. Der Grund ist, daß die einem " folgenden Argumente nicht mehr erkannt werden.

Inhalt

<<

Funktionen

>>

1.28 sort v1.80 Dokumentation - Funktionen - t

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-t Leerzeichen und Tabs am Ende einer Zeile werden gelöscht.

Das ist nützlich, wenn Zeilen, gelöscht werden sollen, die den gleichen Inhalt haben, mit Ausnahme des white space am Zeilenende.

Diese Option wird nun automatisch gesetzt, wenn die -E|N Option gewählt wird, ist aber auch separat aktivierbar, wenn das File nicht sortiert werden soll und nur diese Zeichen entfernt werden sollen.

Inhalt

<<

Funktionen

>>

1.29 sort v1.80 Dokumentation - Funktionen - u

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-u Zeilen, die komplett in einer anderen enthalten sind, werden gelöscht.

Vergleichsstart ist immer der Zeilenanfang, bei Benutzung der C[z,l] Option die Spalte, die durch z angegeben wird.

Beispiel:

Hausaufgaben

keine Hausaufgaben

Hausaufgabe

Es wird nur die Zeile 'Hausaufgabe' gelöscht.

Inhalt

<<

Funktionen

>>

1.30 sort v1.80 Dokumentation - Funktionen - v

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-v Dieser Switch unterdrückt alle Ausgaben, einschließlich der
Warnung, daß das <outfile> bereits existiert.
Es erscheinen lediglich Fehlermeldungen.

Inhalt

<<

Funktionen

>>

1.31 sort v1.80 Dokumentation - Funktionen - w|W

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-w Zeilen, die gelöscht werden, werden zusätzlich auf dem Bildschirm ausgegeben

-W Zeilen, die gelöscht werden, werden zusätzlich in die Datei

<outfile>.dbl geschrieben

Inhalt

<<

Funktionen

>>

1.32 sort v1.80 Dokumentation - Funktionen - x

-a

-A

-b|B

-c|C

-d

-e|E

-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

```
-w|W  
-x  
-y|Y  
-z|Z  
-x die Sortierfunktion wird abgeschaltet.
```

Inhalt

<<

Funktionen

>>

1.33 sort v1.80 Dokumentation - Funktionen - y|Y

```
-a  
-A  
-b|B  
-c|C  
-d  
-e|E  
-f  
-g  
-h  
-i|I  
-j|J  
-l  
-n|N  
-o  
-p|P  
-q|Q  
-r
```

-s|S
-t
-u
-v
-w|W
-x
-y|Y
-z|Z
-y diese Option legt die Sortierspalte nach einem Suchmuster fest ↔

Diese Option arbeitet auch mit der -i Option zusammen. Dabei wird zuerst die Sortierspalte nach Muster gesucht und anschließend die -i Option (ggf. mit Liste) ausgeführt.

-Y diese Option legt die Sortierspalte nach einem Suchmuster fest. Die Datei wird zuerst ohne Option -Y sortiert, es werden doppelte Zeilen gelöscht und vor dem zweiten Durchlauf wird die Sortierspalte gesetzt. Diese Option arbeitet auch mit der -I Option zusammen. Dabei wird zuerst die Sortierspalte nach Muster gesucht und anschließend die -I Option (ggf. mit Liste) ausgeführt.

Inhalt

<<

Funktionen

>>

1.34 sort v1.80 Dokumentation - Funktionen - z|Z

-a
-A
-b|B
-c|C
-d
-e|E
-f

-g

-h

-i|I

-j|J

-l

-n|N

-o

-p|P

-q|Q

-r

-s|S

-t

-u

-v

-w|W

-x

-y|Y

-z|Z

-z[x] <x> gibt die Zahl der Zeilen pro Datensatz an, d.h. es ↔
können

auch Files sortiert werden, bei denen mehrere Zeilen zu einem
Datensatz gehören.

Die Zahl der Zeilen im File sollte ein Vielfaches von <x> sein,
restliche Zeilen am Ende werden abgeschnitten.

-Z[l,c] <l> gibt die Zahl der Zeichen pro Datensatz an und <c>, wieviele
Zeichen zum Sortieren verwendet werden.

Datensätze dürfen außerhalb der zum Sortieren verwendeten Zeichen
auch binäre (nichtlesbare) Zeichen enthalten.

Die Dateigröße sollte ein Vielfaches der Datensatzgröße sein,
restliche Zeichen am Fileende werden abgeschnitten.

Inhalt

<<

Funktionen

>|

1.35 sort v1.80 Dokumentation - Beispiele

```
sort textfile1
```

fragt nach dem Namen für das Ausgabefile und lädt 'textfile1'. Nach dem Sortieren wird das File nach <outfile> geschrieben. Der Vorschlag für das Ausgabefile ist in diesem Fall 'textfile1.sort'.

```
sort textfile1 RAM:text.out -g
```

lädt 'textfile1', sortiert ohne Unterscheidung von Groß- und Kleinbuchstaben und schreibt das sortierte File nach 'RAM:text.out'

```
sort dir1/textfile1 dir2/textfile2 -j3 textfile3 RAM:text.out -E
```

lädt 'textfile1' 'textfile2' und 'textfile3' von verschiedenen Pfaden, sortiert das entstandene File, entfernt alle Leer- und doppelten Zeilen und schreibt die Liste nach 'RAM:text.out'

```
sort RECENT RECENT.sort -E -c20,10 -g -f6
```

ist das Kommando zum Sortieren der Aminet RECENT files.

Das Sortieren beginnt ab Zeile 6, d.h. der Header bleibt unverändert.

Es wird ab Spalte 20 sortiert, es werden 10 Zeichen zum Sortieren verwendet, das entspricht in diesem Fall der Kategorie der Files.

Bei gleichen Typen werden die Zeilen zusätzlich noch nach dem Zeilenanfang sortiert.

Leer- und doppelte Zeilen werden gelöscht, es wird die gesamte Zeile zum Herausfinden der doppelten Zeilen verwendet.

```
sort RECENT ram:file1 -g -E -c20,10 -pdev/ -q/e
```

sortiert das Aminet RECENT file nach dem Programmtyp, löscht alle Leer- und doppelten Zeilen und alle Zeilen, die den String "dev/" nicht enthalten und teilt das verbleibende File in ein File, dessen Zeilen den String "/e" enthalten und eins, das den Rest dieser Liste enthält (wird nach RAM:file1.ohne geschrieben).

```
sort RECENT1 RECENT2 RECENT3 RECENT4 -j4 ram:file1 -g -E -c20,10 -h0,5 -f6
```

fügt vier Files zusammen (hier vier Aminet RECENT files) entfernt in den files 'RECENT2', 'RECENT3' and 'RECENT4' die ersten fünf Zeilen (den Header) sortiert die Datei nach dem Programmtyp (Spalte 20, 10 Zeichen lang) ohne Unterscheidung der Groß- und Kleinbuchstaben. Alle doppelten Zeilen werden nach dem Sortieren entfernt.

Das entstehende File wird nach RAM:file1 geschrieben.

```
sort file1 -j3 file2 file3 ram:big_file -px -Qy -E -g
```

lädt 'file1', 'file2' und 'file3' löscht alle Zeilen, die den String 'x' nicht enthalten und sortiert die Datei ohne Beachtung der Groß- und Kleinbuchstaben. Danach werden alle Zeilen, die den String 'y' enthalten, nach 'RAM:big_file.mit' geschrieben, der Rest nach 'RAM:big_file'.

`sort woebu1 woebu2 -j2 -E -g -I<ignore_list woebu.gesamt`
fügt die Files 'woebu1' und 'woebu2' zusammen, sortiert das File zuerst ohne Beachtung der Ignore-Strings, löscht alle Leer- und doppelten Zeilen und sortiert erneut, diesmal unter Verwendung der Ignore-Strings. Groß- und Kleinbuchstaben werden nicht beachtet.

Inhalt

1.36 sort v1.80 Dokumentation - Tips & Tricks

Die Optionen können an beliebiger Stelle nach dem Programmnamen ←
gesetzt
werden. Wurde der Parameter für einen Switch nicht angegeben, wird im allgemeinen danach gefragt.
Das Programm erkennt einige Widersprüche zwischen gesetzten Schaltern, um mögliche Fehlfunktionen zu vermeiden.

Beachten Sie, daß der outfilename der letzte aller Filenamen sein muß. Wird kein Name für das Ausgabefile angegeben, so wird als Vorschlag der Name des ersten Infiles mit der Endung '.sort' angeboten und kann durch ein RETURN übernommen werden.

Die Sortierreihenfolge basiert auf der Standard strcmp/stricmp Funktion.

BEMERKUNG: wenn Zeichen wie z.B. " angegeben werden, ist es ratsam, diese Option als letzte in der Kommandozeile anzugeben. Der Grund ist, daß die einem " folgenden Argumente nicht mehr erkannt werden.

Inhalt

1.37 sort v1.80 Dokumentation - Known Bugs

Im Moment keine.

Sollten Sie Fehler im Programm finden, bitte schreiben Sie diese mit der genauen Angabe des Aufrufs und der verwendeten Optionen an meine
Adresse
.

Wenn möglich, senden Sie mir auch ein File, bei dem dieser Fehler auftrat, das erleichtert die Fehlersuche ungemein.

Inhalt

1.38 sort v1.80 Dokumentation - Registrierung

Wenn Sie sich registrieren lassen, erhalten Sie eine ↔
Registriernummer,
die alle Funktionen im sort-Programm verfügbar macht.

Die neuesten Versionen befinden sich im Aminet sowie auf meiner

Amiga Seite
Wenn Sie das Programm gern auf einem anderen Computer verwenden ↔
möchten,
dann sehen Sie
hier
nach, ob Ihr System dabei ist. Wenn nicht, dann
senden Sie
mir
trotzdem eine Nachricht, es sollte möglich sein, es auch
auf Ihren Rechner zu portieren.

Die Registrierung kostet 10 DM oder alternativ 6 US\$ oder £ 4.
Senden Sie den Betrag an diese
Adresse
.

Inhalt

1.39 sort v1.80 Dokumentation - Verfügbarkeit

Das Programm ist zur Zeit für die folgenden Systeme erhältlich:

- AMIGA - Ultrix 4.4 - SUN OS 4.1.3 - Linux
- MS-DOS - Solaris 2.4 - AIX 3.2.5

Das Archiv kann entweder vom Aminet (util/sys/sort1_xx.lha)
oder von meiner WWW-Seite kopiert werden:
<http://www.iee.et.tu-dresden.de/~wernerr/amiga.html>

Die neueste Version ist immer auf meiner WWW-Seite zu finden, für alle
oben aufgeführten Systeme.

Wenn Sie sort auf einem Rechnersystem benutzen möchten, das hier nicht
erwähnt ist, senden Sie mir bitte ein e-mail. Ich werde dann das Programm
auf dieses System portieren.

Inhalt
