

# sortk ユーザーズガイド

sortk コマンドは、テキストファイルの行を並べ替えて、標準出力に書き出します。



## CSV

拡張子が `.csv` であるテキストファイルは、各行が、コンマで分離された多数の語句からなります。CSV のファイルは、Excel などの表計算ソフトウェアで容易に表示できます。sortk コマンドは、CSV の欄にある語句を比較して、行を並べ替えます。

## 区切り文字

CSV の行を並べ替えるには、コマンドプロンプトに sortk コマンドのパラメータとして、`-t` (マイナスティーコンマ) を付加してください。`-t` およびコンマの間に空白を挿入しないでください。

### 空白

区切り文字が指定されていない場合、sortk コマンドは、区切り文字として空白を使用します。

## 標準入力

CSV ファイル名が指定されていない場合、sortk コマンドは、標準入力から読み込みます。

## 新しいファイル

下記のとおり、標準出力からファイルにリダイレクトすることで、新しいファイルを作成できます。

```
sortk -t, < csv > text.tmp
```

### 命名

csv を読み込んで、新しいファイルの作成および命名を行うには、コマンドプロンプトに sortk コマンドのパラメータとして、`-o text.tmp` を付加してください。`-o` および `text.tmp` の間に空白を挿入してください。`text.tmp` は、新しいファイルの名前です。

```
sortk -t, -o text.tmp csv
```

## 第 2 欄を比較

sortk コマンドは、たとえば、第 2 欄にある語句を比較して、CSV のテキストファイルの行を並べ替えることができます。第 2 欄にある語句を比較するには、sortk コマンドのパラメータとして、`-k2,2` を付加してください。`-k` および数字の間に空白を挿入しないでください。

# sortk -t, -k2,2 KeyWords.csv

```
C:\¥Q>sortk -t, -k2,2 KeyWords.csv
合図,アズ,
アクセス,アケス,
ASCII,アスキ,
アルファベット,アルファベツト,
アンインストール,アンインストール,
1バイトのカタカナ,イチバイトカタカナ,
インストーラー,インストールアラ,
インストール,インストール,
インターネット,インターネット,
Windows,ウインドウズ,
Windowsロゴキー,ウインドウズロゴキ,
ウェブサイト,ウェブサイト,
英語,エイゴ,
英字,エイジ,
Excel,エクセル,
閲覧,エツラン,
大文字,オオモジ,
改行,カクキョウ,
拡張子,カクサウジ,
下線,カシ,
かな漢字変換,カナジカン,
カレントディレクトリ,カレントディレクトリ,
```

## sortk の由来

sortk(ソーティーケー)という名前は、UNIX の `sort -t, -k` に由来します。FreeBSD, Linux, Solaris などの UNIX は、`sort` コマンドを用意しています。sortk は、UNIX の `sort` コマンドに似ています。

## 日本語

sortk コマンドは、半角カタカナの濁点、半濁点を無視できます。Windows のコマンドプロンプトで、EUC-JP のテキストファイルを並べ替えることができます。0Ah のみで改行できます。

# インストール

sortk コマンドは、Windows のコンソールアプリケーションです。32 ビット版 Windows で構築しました。

## パスを通す手順の例

1. トリシーカー (<http://tori.tobihiro.jp/PDF-zip-C.html>) というウェブサイトから sortk-C.zip をダウンロードしてください。
2. sortk-C.zip を展開してください。
3. Cドライブに sortk-C フォルダを移動してください。
4. 利用者は、sortk-C フォルダの移動および改名が可能です。
5. Windows ロゴキー(田キー)を押しながら Pause を押してください。
6. メニューからシステムの詳細設定を選択してください。
7. 詳細設定タブをクリックしてください。
8. [環境変数]ボタンを押してください。キーワード: **システム、詳細、環境変数**
9. システム環境変数として、Path が見えるまでスクロールしてください。
10. Path をクリックして選択してください。
11. [編集]ボタンを押してください。
12. Windows 10 の場合、[テキストの編集]ボタンを押してください。
13. 右向き矢印(→)キーまたは End キーを押してください。カーソルが末尾に移動します。
14. フォルダのことをディレクトリとも言います。セミicolon( ; )は、ディレクトリどうしを区切る記号です。 ;C:¥sortk-C を入力してください。
15. 各ウィンドウで OK ボタンを押してください。

### フォルダごと削除

アンインストール(プログラムの削除)を行うには、sortk-C フォルダごと削除してください。Path から C:¥sortk-C を削除してください。

## 著作権

Sogaya(そがや)は、sortk の著作権を保有しています。ソースコードを添付しました。

## 免責条項

sortk コマンドのご利用によって発生するいかなる損害も、Sogaya は、責任を負わないものとします。

## 構築

Dev-C++の IDE において、C のプロジェクトとして GCC で sortk を構築しました。

## gcc (GCC) 3.4.2 (mingw-special)

### bcc32

Borland 無料コマンドラインコンパイラで構築できます。makefile.mak を添付しました。

## Borland C++ 5.5.1 for Win32

## ユーザーズガイド

2019 年 8 月 16 日 Revision 1.01902

## sortk コマンドのパラメータ

sortk コマンドは、CSV の欄にある語句、整数、浮動小数点数を比較して、行を並べ替えます。利用者は、CSV の欄を選択できます。

## コマンドプロンプト

Windows のコマンドプロンプトは、シフト JIS が多用されます。MS932 は、いわゆるシフト JIS ですが、規格外です。たとえば、丸付きの数字(①②③⑩⑳)、ローマ数字(I II III IV X)、1 バイトのカタカナは、EUC-JP で表現しにくい。

### CHCP コマンド

コマンドプロンプトの文字コードがシフト JIS に設定されていることを確認するには、パラメータを付加しないで、コマンドプロンプトに CHCP を入力して、Enter を押してください。932 が表示されます。MS932 が使用されることがわかります。どの Windows も最初から CHCP コマンドを用意しています。MS932 は、CP932 とともに Windows-31J とも言います。



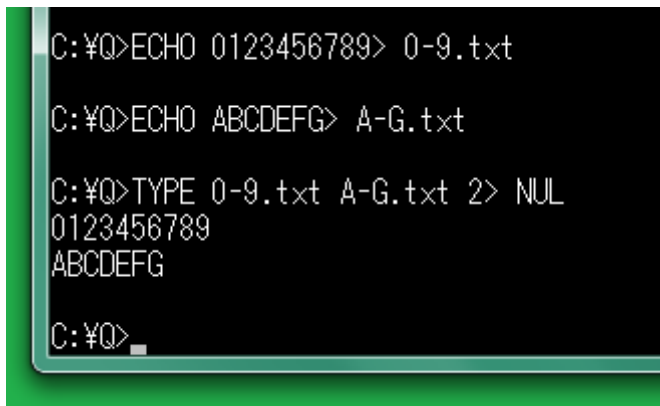
```
C:\¥Q>CHCP
現在のコード ページ: 932
C:\¥Q>
```

## TYPE コマンド

コマンドプロンプトでシフト JIS のテキストファイルを閲覧する方法として、TYPE コマンドが利用できます。どの Windows も最初から MORE コマンドを用意しています。

## ファイルを連結する

UNIX の `cat` コマンドは、ファイルを連結できますが、Windows の `TYPE` コマンドでファイルを連結するには、`2> NUL` を付加してください。NUL がファイルであるかのように使用します。2>は、標準エラー出力からファイルに内容をリダイレクトします。NUL にリダイレクトされた内容は、表示されません。

A screenshot of a Windows command prompt window with a black background and white text. The prompt shows the following commands and their outputs:

```
C:¥Q>ECHO 0123456789> 0-9.txt
C:¥Q>ECHO ABCDEFG> A-G.txt
C:¥Q>TYPE 0-9.txt A-G.txt 2> NUL
0123456789
ABCDEFG
C:¥Q>
```

## MORE コマンド

`TYPE` コマンドは、いきなり最後のページを表示しますが、`MORE` コマンドは、1 ページから順に表示できます。スペースキーで進行します。Qで終了します。どの Windows も最初から `MORE` コマンドを用意しています。

### 中止したい合図

コマンドプロンプトに、中止したい合図を入力するには、`Ctrl` を押しながら、`C` を押してください。`Ctrl + C` は、この操作を意味します。`^C` が画面に表示される場合があります。

## atnd コマンド

添付の `atnd` は、シフト JIS 用のコンソールアプリケーションです。シフト JIS のファイルの最初から 7 行を表示するには、コマンドプロンプトに、たとえば、下記のコマンドおよびパラメータを入力して、`Enter` を押してください。`-h` の直後に `7` を入力してください。`-h7` は、`-b6 -p7` を意味します。

```
atnd -h7 Gongitsune.txt
```

```
C:¥Q>atnd -h7 Gongitsune.txt
こん狐
新美南吉
—
これは、私《わたし》が小さいときに、村の茂平《もへい》というおじいさんからきいたお話です。
むかしは、私たちの村のちかくの、中山《なかやま》というところに小さなお城があって、中山さまというおとのさまが、おられたそうです。
C:¥Q>
```

## FC コマンド

テキストファイルを比較するために、FC コマンドを試すことができます。どの Windows も最初から FC コマンドを用意しています。

```
C:¥Q>sortk -t, -k2,2 -o text.tmp KeyWords.csv
C:¥Q>sortk -t, -k2,2 KeyWords.csv > new.csv
C:¥Q>FC text.tmp new.csv
ファイル text.tmp と NEW.CSV を比較しています
FC: 相違点は検出されませんでした
C:¥Q>
```

## バージョン番号

sortk コマンドのバージョン番号を確認するには、-V を付加してください。

## sortk -V

### 短い説明

パラメータに関する短い説明を表示するには、--help を付加してください。画面からテキストファイルに、リダイレクトするには、-s を前置してください。

```
sortk -s --help > help.txt
```

```
C:\>sortk -s --help > help.txt
```

```
C:\>NOTEPAD help.txt
```

help - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

sortk C-1.110

```
sortk -t, -o new csv # Read csv to produce a new file.
sortk -t, < csv > new # Read csv to produce a new file.
sortk -t, -a4 -h1011 csv # Compare in Field-1, Field-3 and Field-4.
sortk -t, -j2 csv # Compare phrases in Field-2.
sortk -t, -k2,2 csv # Compare phrases in Field-2.
sortk -t, -k2,3 csv # Compare phrases from Field-2 to Field-3.
sortk -t, -k2 csv # Compare phrases from Field-2 to finish.
sortk -t, -k1 -$ csv # Check if all lines share the field count.
sortk -t, -k2,2 -# csv # Add ID to each summary.
sortk -t, -k2,3 -# -! csv # Report ID if any summaries match each other.
sortk -t, -k2,2 -8 csv # UTF-8
sortk -t, -k2,2 -b csv # Skip space in the first part of Field-2.
sortk -t, -k2,2 -c- csv # Skip combining characters.
sortk -t, -k2,2 -e csv # EUC-JP
sortk -t, -k2,2 -f csv # EIGHT equals eight.
sortk -t, -k2,2 -g csv # Compare 0.12 with 0.78 in Field-2.
sortk -t, -k2,2 -i csv # Skip control characters.
sortk -t, -k2,2 -l csv # Show how large the buffer is.
sortk -t, -k2,2 -n csv # Compare 3456 with 9012 in Field-2.
sortk -t, -k2,2 -p csv # P and Q are omitted from ASCII characters.
sortk -t, -k2,2 -q csv # Commas disappear if placed in quotes.
sortk -t, -k2,2 -r csv # Arrange in descending order.
sortk -t, -k2,2 -u csv # Display a line for each unique summary.
sortk -t, -k2,2 -y csv # Do not remove binary.tmp
sortk -t, -k2,2 -y -% csv # Produce manager.txt
sortk -t, -k2,2 -z256 csv # Use a buffer of 256 bytes.
sortk -t¥1 -k2,2 file # Wiki table
sortk -t¥1 -k2,2 -x- csv # Do not check if the file name includes .csv
sortk -t¥x7C -k2,2 file # Wiki table
sortk -U131072 -k2,2 csv # The number of lines shall not exceed 131072.
sortk -v¥x1F -k2,3 csv # 1Fh connects phrases from Field-2 to Field-3.
sortk -w ~ -k2,4 csv # ~ is used as a quote.
sortk -w ~ -k2,4 csv # ~ is used as a quote.
```

パラメータ	ほぼ同等な入力	説明
-!		要約が合致した場合、ID を表示する。
-\$		どの行も同じ欄数を有しているか点検する。
-#		要約に ID を付与する。
-%		設定ファイルとして、manager.txt を作成する。
-8	-e	0Ah のみで改行する。
-a4	-h1111000000	第 1 欄から第 4 欄まで選択する。
	-k1,4	
-b		比較開始欄の前半にある空白を無視する。
-c-		半角カタカナの濁点、半濁点を無視する。
-e	-8	0Ah のみで改行する。
-f		大文字と小文字を区別しないで比較する。
-g		浮動小数点数を比較する。
-h1011		第 2 欄を除外する。
-i		タブ、改ページなど制御文字を無視する。
-j2	-k2,2	第 2 欄のみ選択する。
-k2		第 2 欄から最終欄まで選択する。
-k2,3	-a3 -h011	第 2 欄から第 3 欄まで選択する。
-l		何バイトのバッファを使用するか表示する。
-n		整数を比較する。
-o new	> new	新しいファイルを命名する。
-p		濁点の代用文字として、P を無視する。
-q		引用符で囲まれた区切り文字を削除する。
-r		降順で並べ替える。
-s		標準エラー出力ではなく標準出力に書き出す。
-t,	.csv の拡張子	区切り文字としてコンマを使用する。
-t¥x7C	-t¥l	Wiki の表の区切り記号として縦線を使用する。
-U131072		行数は、131072 を超えてはならない。
-u		要約ごとに 1 行のみ表示する。
-v¥x1F		選択された語句を 1Fh で連結する。
-VC	--help	パラメータの短い説明を表示する。
-w~		~ が引用符として利用される。
-x-		.csv がファイル名に含まれるか点検しない。
-Y		binary.tmp を削除してはならない。
-z256		256 バイトのバッファを使用する。



## ¥x および十六進数

上記のパラメータのうち、-t, -v, -w は、十六進数で1バイト文字を指定できます。たとえば、-w¥x7E は、-w~と同じ結果を生じます。

## ¥v で大なり記号を意味する

AWK, C#などのプログラミング言語では、¥v は、垂直タブ(0Bh)を意味する場合がありますが、sortk コマンドのパラメータでは、大なり記号(>)を意味します。すなわち、-t¥v は、-t¥x3E と同じ結果を生じます。下表では、-t の例を示します。-v, -w の表も同様になります。

パラメータ	ほぼ同等な入力	1バイト文字	説明
-t¥A	-t¥x3C	<	小なり記号
-t¥^	-t¥x3C	<	小なり記号
-t¥	-t¥x7C		縦線
-t¥s	-t¥x20		空白
-t¥t	-t¥x09		タブ、水平タブ
-t¥v	-t¥x3E	>	大なり記号

```
C:¥Q>sortk -t¥| -k2,2 equivalent.txt
|t&yen;A|t&yen;x3C|&lt;|小なり記号| |
|t&yen;| |t&yen;x7C|&#124;|縦線|
|t&yen;s|t&yen;x20| |空白|
|t&yen;t|t&yen;x09| |タブ、水平タブ|
|t&yen;v|t&yen;x3E|&gt;|大なり記号|
|t&yen;^|t&yen;x3C|&lt;|小なり記号|
|~パラメータ|~ほぼ同等な入力|~1バイト文字|~説明|
C:¥Q>
```

## ¥| は縦線を意味する

Wiki の表は、区切り記号として縦線を使用します。上記の equivalent.txt は、Wiki の表のテキストファイルです。第1欄および第6欄は、空です。第2欄のみ比較して、並べ替えました。Wiki では、たとえば、以下の文字列は、円記号(¥)に置換されます。

**&yen;**

## かな漢字変換

コマンドプロンプトで、かな漢字変換を行うには、Alt を押しながら[半角／全角]キーを押してください。Windows 7, Windows 10 の場合、Alt を押す必要がありません。かな漢字変換モードから脱出するには、[半角／全角]キーを押してください。

### F8 で半角カタカナに変換

コマンドプロンプトをかな漢字変換モードに移行して、ローマ字入力法で半角カタカナに変換する手順の例を示します。

1. ECHO を入力してください。スペースキーを押して空白を入力してください。まだ Enter キーを押さないでください。
2. Alt を押しながら[半角／全角]キーを押してください。かな漢字変換モードに移行します。
3. H キーを押してください。
4. E キーを押してください。へが表示されます。
5. F8 を押してください。へからへに変化します。
6. Enter キーを押してください。かな漢字変換が確定します。
7. もう一回 Enter キーを押してください。ECHO コマンドおよびパラメータが入力されます。
8. コマンドプロンプトに半角カタカナのへが表示されます。
9. かな漢字変換モードから脱出するには、[半角／全角]キーを押してください。

## パラメータからコンマを排除

sortk コマンドのパラメータとして、-t,(マイナスティーコンマ)も-k2,3(マイナスケーニコマサン)もコンマが含まれていますが、コンマの使用を回避できます。

### 範囲を表現するハイフン

コンマではなくハイフンを使用できます。-k2,3 ではなく-k2-3 が指定されても、sortk コマンドは、第 2 欄から第 3 欄までにある語句を比較します。

### 十六進数で入力

コンマの十六進数は、2Ch です。-t,ではなく-t¥x2C が指定されても、sortk コマンドは、区切り文字としてコンマを使用します。

### .csv の拡張子

何も指定されなければ、sortk コマンドは、区切り文字として空白(20h)を使用します。行が並べ替えられるテキストファイルの拡張子が.csv である場合、sortk コマンドは、、自動的に、区切り文字としてコンマを使用します。拡張子の点検を抑制するには、sortk コマンドのパラメータとして、-x-を付加してください。

```

C:¥Q>sortk -t, -k10,20 -o text.tmp Field-20.csv
C:¥Q>sortk -t¥x2C -k10-20 -o yen_x2C.csv Field-20.csv
C:¥Q>sortk -k10-20 -o extension.csv Field-20.csv
C:¥Q>FC text.tmp yen_x2C.csv
ファイル text.tmp と YEN_X2C.CSV を比較しています
FC: 相違点は検出されませんでした
C:¥Q>FC text.tmp extension.csv
ファイル text.tmp と EXTENSION.CSV を比較しています
FC: 相違点は検出されませんでした
C:¥Q>

```

## 要約および詳細のブロック

利用者は、多数の欄を選択できます。sortk コマンドは、CSV の表を読み込んで、利用者選択の欄で要約を構成して、要約に詳細を結合します。要約のみ比較しながら要約および詳細のブロックを並べ替えます。詳細のみ標準出力に書き出します。

### 要約を構成

たとえば、21 欄もある CSV で、まず、第 1 欄から第 5 欄までを選択するには、sortk コマンドのパラメータとして、-a5 を付加してください。次に、第 2 欄を選択から除外するには、sortk コマンドのパラメータとして、-h10111 を付加してください。欄を選択することの真偽を示すために、1 または 0 を入力してください。10111 のうち 0 は、第 2 欄を選択しないことを意味します。結果として、4 欄で要約を構成します。ただし、-h10111 が指定されない場合、sortk コマンドは、-h11111 が指定されたとみなします。

-h10111					
-h	1	0	1	1	1
<b>真偽</b>	第 1 欄選択	第 2 欄を除外する	第 3 欄選択	第 4 欄選択	第 5 欄選択

### 語句の連結

利用者は、多数の欄を選択できます。利用者選択の欄で要約を構成するとき、語句を連結する文字として、ベル文字 (07h) が使用されます。語句連結文字は、sortk の内部で要約に使用されますが、詳細として書き出されません。

### 制御文字と空白を比較

二桁の十六進数で語句連結文字を設定できます。07h ではなく、たとえば 1Fh を使用

するには、sortk コマンドのパラメータとして、`-v%x1F` を付加してください。下記の例では、まず、`-v%xHH`(H は、十六進数です。[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F]) のパラメータを付加しない sortk コマンドを実行しました。次に`-v%x7F` を付加してみました。

```
C:¥Q>sortk -t, -k1,3 cities.csv
Kyoto,Kyoto,Japan
Kyoto City,Kyoto Prefecture,Japan
Nara,Nara,Japan
Nara City,Nara Prefecture,Japan
Osaka,Osaka,Japan
Osaka City,Osaka Prefecture,Japan

C:¥Q>sortk -t, -k1,3 -v%x7F cities.csv
Kyoto City,Kyoto Prefecture,Japan
Kyoto,Kyoto,Japan
Nara City,Nara Prefecture,Japan
Nara,Nara,Japan
Osaka City,Osaka Prefecture,Japan
Osaka,Osaka,Japan

C:¥Q>
```

### 語句が短いほど上に来るべきだ

たとえば、小文字の `z` は、`7Ah` です。`7Fh` は、`7Ah` より大きいため、`Kyoto City` の下に `Kyoto` が来ました。語句が短いほど上に来るように、`07h`, `1Fh` など、`20h` より小さい数を設定するのが好適です。

## 引用符を変更したい

引用符が出現したら、sortk コマンドは、もうひとつの引用符が出現するまで、区切り文字を無視しますが、自由な表現として、いわゆる引用符( `"` )が欄内に出現する場合は、**sortk コマンドが認識する引用符**を他の文字に変更しなければなりません。

### 引用符を設定

チルダ( `~` )が出現した場合もうひとつのチルダが出現するまで区切り文字を無視するように設定するには、sortk コマンドのパラメータとして、`-w~`を付加してください。sortk コマンドは、引用符として`~`を使用します。いわゆる引用符( `"` )は、もはや引用符として認識されません。

```
C:\¥Q>sortk -t, -k2,2 Mark-14.txt
35,~And going a little farther, he fell on the ground and prayed that, if it wer
e possible, the hour might pass him.~
36,~And he said, "Abba, Father, all things are possible for you. Remove this cup
 from me. Yet not what I will, but what you will."~
34,~And he said to them, "My soul is very sorrowful, even to death. Remain here
 and watch."~

C:\¥Q>sortk -t, -k2,2 -w~ Mark-14.txt
35,~And going a little farther, he fell on the ground and prayed that, if it wer
e possible, the hour might pass him.~
34,~And he said to them, "My soul is very sorrowful, even to death. Remain here
 and watch."~
36,~And he said, "Abba, Father, all things are possible for you. Remove this cup
 from me. Yet not what I will, but what you will."~
```

### 語句が短いとみなされた

上記の例では、まず、`-wA`(`A` は、ASCII の文字です)のパラメータを付加しない `sortk` コマンドを実行しました。次に`-w~`を付加してみました。`-w~`が付加されていない場合、`sortk` コマンドは、`And he said` の次も区切り文字としてコンマを使用して、語句が短いほど上に来るように行を並べ替えます。このため、マルコ 14 章 36 節がマルコ 14 章 34 節の上に来ました。

### いわゆる引用符もコンマも無視するべきだ

自然な表現として、引用符もコンマも `sortk` コマンドに無視されるべきである場合、`-w~` を付加してください。CSV であるにもかかわらずコンマを無視したい範囲を`~`で囲んでください。

## 制御文字

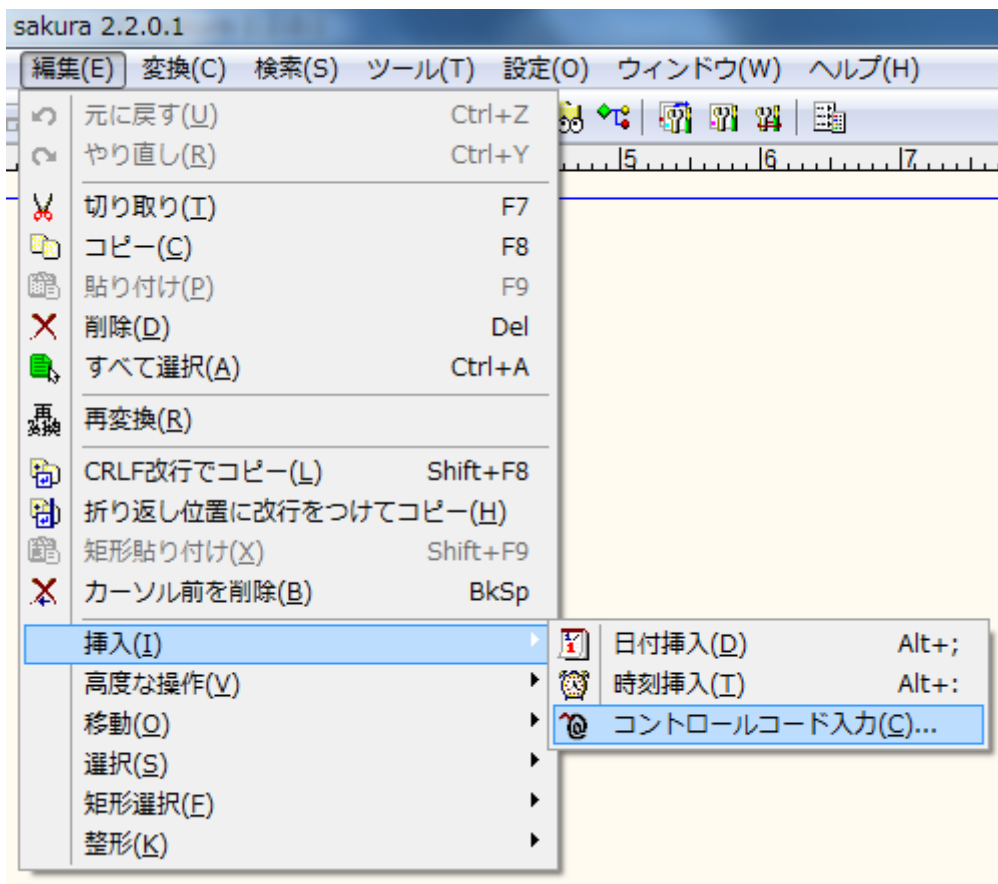
多機能のテキストエディタは、制御文字を入力する機能があると思われます。たとえば、サクラエディタの場合、編集メニューから挿入を選択して、サブメニューからコントロールコード入力を選択できます。

### EmEditor

EmEditor Free Version 5.00 の場合、編集メニューから高度な操作を選択して、特殊文字を入力を選択できます。

### MIFES

MIFES 10 の場合、編集メニューから制御コードの挿入を選択できます。



### タブおよび改ページ

タブ、改ページ、エスケープシーケンスなどの制御文字を無視して比較するには、sortk コマンドのパラメータとして、-i を付加してください。制御文字の例

十六進数	数値	英語	略号	説明
07h	7	alert	BEL	ベル、警告
08h	8	back space	BS	バックスペース、削除
09h	9	horizontal tab	HT	タブ、水平タブ
0Ah	10	line feed	LF	改行
0Bh	11	vertical tab	VT	垂直タブ
0Ch	12	form feed	FF	改ページ
0Dh	13	carriage return	CR	復帰
1Ah	26	substitute	SUB	MS-DOS テキストファイルの終わり
1Bh	27	escape	ESC	エスケープシーケンスの開始

### 表計算ソフトウェア

テストファイルとして添付の Field-21.csv は、第 1 行のみ第 21 欄が記入されています。  
Field-21.csv を Excel 2007 で読み込んだ例

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	85	52	86	48	68	22	59	44	63	61	92	37	9	83	55	58	51	31	60	2	
3	18	53	10	36	22	69	60	98	2	20	39	25	29	44	99	60	60	26	3	73	
4	93	15	38	47	92	70	49	2	38	69	29	9	21	27	86	89	73	25	31	47	
5	32	43	69	31	69	38	13	47	18	60	16	13	30	66	2	80	88	30	92	30	
6	99	29	69	42	17	87	81	79	28	21	72	26	60	97	49	60	99	14	52	66	
7	25	4	79	44	93	20	63	94	43	93	44	93	29	64	86	97	49	80	70	24	
8	13	1	76	59	17	21	45	22	96	88	68	46	88	92	25	94	8	50	64	40	
9	67	93	45	92	76	73	41	52	46	83	39	76	70	76	30	85	77	18	92	43	
10	33	21	28	62	4	51	53	24	39	44	59	2	72	7	14	47	27	30	74	0	
11	84	67	32	53	65	25	84	47	21	80	29	93	21	63	73	24	44	75	79	68	
12	9	38	66	3	14	28	54	67	74	37	54	34	27	82	35	70	7	43	2	43	
13	97	58	86	22	59	80	53	71	90	91	58	37	33	41	59	53	10	6	99	27	
14	24	23	24	46	34	8	74	81	77	51	24	92	89	29	17	30	63	29	3	21	
15	87	56	5	3	86	61	83	59	81	40	87	98	24	90	27	7	22	38	74	4	
16	41	81	72	43	93	53	45	54	95	9	40	70	88	39	28	99	69	85	80	66	

### 最終の欄が空である

上記のとおり、Excel 2007 で読み込んだ場合、最終の欄が空であるか判明します。最終の欄が空である場合、sortk コマンドは、最終の欄をカウントしません。

### 欄をカウント

利用者は、第 1 行から最終行まで選択できますが、sortk コマンドは、ファイルから行を読み込んで、各行で欄をカウントします。どの行も同じ欄数を有しているか点検するには、sortk コマンドのパラメータとして、-\$を付加してください。まず、sortk コマンドは、第 1 行をカウントします。次に、各行が第 1 行と同じ欄数を有しているか点検します。さらに、第 1 行とは欄数が異なる場合、標準出力に行を書き出します。

## sortk -t, -k1 -\$ isbn-r.csv

```
C:\¥Q>sortk -t, -k1 -$ isbn-r.csv
134,1998,17.00mm,ISBN 3-540-63205-0,Springer,The Human Hippocampus,"Functional Anatomy, Vascularization and Serial Sections with MRI",By,
127,2013,17.05mm,ISBN 978-0-89042-556-5,American Psychiatric Association,Desk Reference to the Diagnostic Criteria From DSM-5,,By,
117,2010,12.37mm,ISBN 978-3-642-07505-6,Springer,Signaling Pathways for Translation,"Stress, Calcium, and Rapamycin",By,
102,2011,25.00mm,ISBN 978-0-12-385522-0,Elsevier,Advances in Experimental Social Psychology,Volume 44,By,
C:\¥Q>
```

### 不規則な表

テストファイルとして添付の isbn-r.csv は、9 欄ありますが、最終の欄は記入されていない行があります。

### CSV の入力ミスを発見

CSV エディタ、表計算ソフトウェアではなくテキストエディタで CSV を作成した場合、入

カミスで表がずれるおそれがあります。表の欄数(表の列数)が異なる部分を入力することで、CSV の入力ミスを発見できる場合があります。

## テストファイルおよび変換表

test\_files というディレクトリには、テストファイルおよび変換表があります。変換表のファイル名は、MS932~Shift\_JIS.bin です。

### 元の変換表をコピーしたい

dayu コマンドを実行するたびに新しい変換表(バイナリファイル)が作成されます。

```
C:¥sortk-C¥test_files¥MS932~Shift_JIS.bin
```

を C:¥sortk にコピーするために、利用者は、C:¥sortk-C で ctm.bat というバッチファイルを実行できます。

## ctm

### 行番号

sortk コマンドは、順序が確定しない場合、行番号で順序を確定できます。sortk コマンドのパラメータとして、-#を付加してください。

### 制御文字を無視するべきだ

下記の例では、mini.txt という CSV にタブ、改ページ、エスケープシーケンスが含まれています。-i 無しの sortk コマンドは、文字または数字を制御文字と比較します。-i 無しの sortk コマンドの結果として、40 より上に 86 が来ました。01 より上に 94 が来ました。しかし、小さい数が上に来るように、制御文字を無視するべきです。

### シフト JIS のテキストファイル

シフト JIS のテキストファイルである場合、-i 付きの sortk コマンドは、2 バイトの文字、ASCII、半角カタカナのみで要約を構成します。すなわち、1 バイトの文字として、制御文字は、要約に含まれません。

### 浮動小数点数

浮動小数点数を比較するには、-g を付加してください。10 欄まで選択できます。

### 語句を比較できない

浮動小数点数を比較するために -g が指定された場合、sortk コマンドは、もはや語句を比較できません。



## 桁数が同一ではない

桁数が同一ではない整数を比較するには、`-n` を付加してください。20 欄まで選択できます。`-n` が指定された場合、`sortk` コマンドは、もはや語句を比較できません。

```
C:¥Q>sortk -t, -k2,2 -n mini.txt
4956回,01,
5035回,12,
4908回,14,
5043回,15,
5000回,17,
4976回,18,
4944回,19,
4972回,20,
4429回,24,
4960回,25,
4840回,27,
4759回,28,
5049回,32,
4851回,38,
5006回,39,
5014回,40,
4973回,41,
4967回,51,
4902回,59,
4647回,64,
5025回,68,
5030回,70,
5009回,74,
5032回,83,
4881回,86,
4983回,90,
4954回,93,
5037回,94,
C:¥Q>
```

### 桁数が同一である

下記の例では、`mini.txt` という CSV にタブ、改ページ、エスケープシーケンスが含まれています。制御文字を無視するために `-i` を付加した `sortk` コマンドは、整数でも浮動小数点数でもなく語句を比較します。すなわち、数ではなく数字を比較します。比較対象の数が、すべて 2 桁である場合、`-i` を付加した例を示します。

```
C:¥Q>sortk -t, -k2,2 -i mini.txt
4956回,01,
5035回,12,
4908回,14,
5043回,15,
5000回,17,
4976回,18,
4944回,19,
4972回,20,
4429回,24,
4960回,25,
4840回,27,
4759回,28,
5049回,32,
4851回,38,
5006回,39,
5014回,40,
4973回,41,
4967回,51,
4902回,59,
4647回,64,
5025回,68,
5030回,70,
5009回,74,
5032回,83,
4881回,86,
4983回,90,
4954回,93,
5037回,94,
C:¥Q>
```

## 欄の前半にある空白

空白は、制御文字ではありません。比較開始欄の前半にある空白を無視するには、`sortk` コマンドのパラメータとして、`-b` を付加してください。下記の例では、添付の `atnd` コマンドでファイルの先頭から 3 行をコマンドプロンプトに表示しました。空白が無視されたため、並べ替えた結果が異なります。空白を無視するために `-b` を付加した `sortk` コマンドは、整数でも浮動小数点数でもなく語句を比較します。

```
C:\>sortk -t, -k1,3 isbn-r.csv | atnd -h3
71,2014,21.00mm,ISBN 978-0-444-63425-2,Elsevier,Dopamine,,By,Marco Diana,
73,2000,38.58mm,ISBN 0-19-853768-9,Oxford,Temporal Logic,,By,Dov M. Gabbay,
73,2010,21.77mm,ISBN 978-0-262-51311-1,MIT,What Is Addiction?,,By,Don Ross,

C:\>sortk -t, -k1,3 -b isbn-r.csv | atnd -h3
100,1993,28.08mm,ISBN 0-8493-4441-7,CRC,Limbic Motor Circuits and Neuropsychiatry,,By,Peter W. Kalivas,
100,2010,12.30mm,ISBN 978-0-7618-5041-0,Hamilton,Come Along,We Are Truth-Bound,By,Jacinta Respondowska,
100,2010,33.85mm,ISBN 978-1-84882-127-9,Springer,Atlas of Epilepsies,Volume 1,By,C. P. Panayiotopoulos,

C:\>
```

### 小数点が発見されない

浮動小数点数を比較するために-g が付加された sortk コマンドは、小数点を発見しません。小数点が発見されない場合、0.0 になります。

## 0.0

### 数字に到達するまで無視される

浮動小数点数を比較するために-g が付加された sortk コマンドは、比較開始欄の前半にある空白を無視します。負号、小数点、数字に到達するまで、空白だけではなく文字を無視します。整数を比較するために-n が付加された sortk コマンドは、比較開始欄の前半にある空白を無視します。負号、数字に到達するまで、空白だけではなく文字を無視します。

## ECHO コマンド

Windows の環境変数として、Path があります。PATH とも書きます。PATH の内容を表示するには、コマンドプロンプトに ECHO コマンドおよびパラメータを入力して、Enter を押してください。

## ECHO %PATH%

### %ERRORLEVEL%

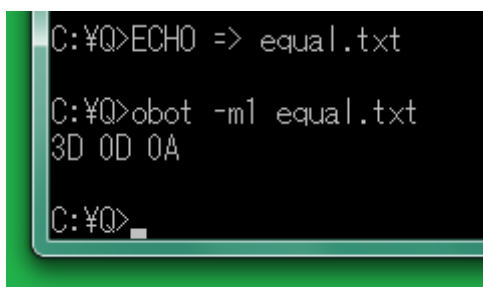
ECHO コマンドは、コマンドプロンプトに値を表示します。終了するときプログラムが Windows に返した値を表示するには、パラメータとして %ERRORLEVEL% を使用します。

## ECHO %ERRORLEVEL%

## テキストファイルを作成

ECHO コマンドの出力をリダイレクトすれば、テキストファイルに内容を書き込むことができます。たとえば、等号(=)を内容とするテキストファイルを作成するには、コマンドプロンプトに以下のコマンドおよびパラメータを入力して、Enter を押してください。等号および改行がテキストファイルに書き込まれます。

# ECHO => equal.txt



```
C:\>ECHO => equal.txt
C:\>obot -m1 equal.txt
3D 0D 0A
C:\>
```

## 制限

32 ビット版 Windows において、Dev-C++という IDE で、C のプロジェクトとして sortk コマンドを構築しました。ソースコードを添付しました。用途によっては、sortk コマンドの利用者は、限界、制限を感じるかもしれません。

### 要約のサイズ

利用者は、多数の欄を選択できます。sortk コマンドは、利用者選択の欄で要約を構成します。sutai.h というヘッダーファイルによれば、要約に含める浮動小数点数は、最高 10 個です。1 個の浮動小数点数は、8 バイトです。すなわち、sortk コマンドは、80 バイトまで正しく比較します。

### 比較される語句のサイズ

sortk コマンドは、最大 79 バイトまで比較して行の順序が確定することを想定しています。C の文字列は、終端として、1 バイトの零が必要であるため、80 バイトではなく 79 バイトが最大値です。

### 要約に ID を付与できる

80 バイトまで合致した場合も順序を確定するために、要約に ID を付与するには、sortk コマンドのパラメータとして、-#を付加してください。要約に ID を付与した効果については、後述します。

### segmented.h

利用者は、多数の欄を選択できますが、segmented.h というヘッダーファイルによれば、欄数の上限は、3072 です。欄数の上限を変更するには、segmented.h を修正して、sortk を再構築してください。ただし、欄数が多すぎると、sortk コマンドの実行

が低速になるおそれがあります。

ヘッダーファイル	名前	数値	説明	問題
sutai.h	BLOCK_SIZE	3072	要約および詳細のブロックのサイズ	行が長すぎるため、詳細を完全には記憶できない。
	DOUBLE_ARRAY_SIZE	10	要約に含める浮動小数点数の個数	要約の上限が10 欄しかない。10 欄が合致する。行を完全には比較できない。
	INTEGER_ARRAY_SIZE	20	要約に含める整数の個数（浮動小数点数の2倍の個数）	要約の上限が20 欄しかない。20 欄が合致する。行を完全には比較できない。
	END	80	要約に含める1バイト文字の個数（浮動小数点数の8倍の個数）	要約の上限が79 バイトしかない。79 バイトが合致する。 <b>行を正しく比較したとはいえない。</b>
	LIMIT_OF_MY_MEMORY	128	要約が合致した行の数	記憶力に限界がある。
segmented.h	FIELD_COUNT	3072	欄数の上限	3072 欄しか選択できない。

## 大文字と小文字を区別しない

大文字と小文字を区別しないで比較するには、sortk コマンドのパラメータとして、-f を付加してください。ただし、ASCII のみ効果があります。

## 行数

ファイルの行数は、16384 を超えてはなりません。行数の上限を変更するには、sortk コマンドのパラメータとして、-U 数を付加してください。たとえば、16384 ではなく、

131072 を上限とするには、-U131072 を付加してください。下記の例では、-U20 を付加してみました。atnd コマンドで 5156 行もあることがわかりましたが、第 1 行から第 20 行までのみ並べ替えることができました。

```
C:¥Q>atnd -LF -p20 numbers7.csv
numbers7.csv: 5156 lines
maximum LF-to-LF span of 13 bytes (line-1000)
20,502,2264

C:¥Q>sortk -t, -k2,2 -U20 numbers7.csv
15,019,6419
4,105,0171
16,179,0524
1,191,1149
3,194,0097
13,229,4673
8,234,7507
9,243,2664
12,247,9722
18,291,3062
11,326,0254
10,340,2436
17,475,8697
20,502,2264
5,592,0265
7,708,3950
14,743,1409
6,792,5482
19,852,1013
2,988,7921

C:¥Q>
```

## 改行に到達しないほど小さいバッファ

sortk コマンドは、-z 数でバッファのサイズを設定できます。どの行も末尾まで読み込めるように、バッファのサイズを設定してください。極端に小さいサイズが指定された場合、sortk コマンドは、1 回の読み込みで改行に到達しないため、改行文字が含まれていないエラーを表示して中止します。バッファのサイズが指定されていない場合、sortk コマンドは、自動的に最長の行より大きいサイズにバッファを設定します。

### EOF の前で改行してください

各行は、改行文字が必要です。**ファイルの終わり (EOF) の直前で改行してください。**制御文字として、1Ah をテキストファイルの途中に挿入してはなりません。

# CON

昔のコンピュータの入力は、パンチカードでした。キーボードがパンチカードの後継者となりました。コマンドプロンプトが 80 桁であるのは、80 桁のパンチカードの名残らしい。パソコンの標準入力は、キーボードです。Windows 10 の場合、既定値(デフォルト)は、80 桁ではなく 120 桁です。コマンドプロンプトには、CON が用意されています。キーボードがファイルであるかのように、CON を入力できます。

## COPY コマンドと CON を併用する

MORE コマンドと異なり、COPY コマンドは、コピーしたい情報を標準入力から入力できません。COPY コマンドは、Y, N, A を標準入力から入力できます (Yes, No, All)。新しいファイル(equal.txt)に書き込むには、コマンドプロンプトに、以下のコマンドおよびパラメータを入力して、Enter を押してください。Ctrl を押しながら Z を押して、標準入力モードから脱出してください。^Z が表示されます。

## COPY CON serial.txt



```
C:\¥Q>COPY CON serial.txt
0
^Z
        1 個のファイルをコピーしました。

C:\¥Q>obot -m1 serial.txt
30 0D 0A

C:\¥Q>_
```

# NUL

パソコンの標準出力は、画面です。画面ではない装置に情報を転送すれば、**画面に出力されません**。NUL は、画面でもファイルでもコマンドでもありません。画面ではない装置がファイルであるかのように、NUL を入力できます。

## dayu コマンドと連携する

添付の dayu コマンドは、MS932~Shift\_JIS.bin という変換表を編集できます。変

換表は、バイナリファイルです。もうひとつの機能として、dayu コマンドは、シフト JIS のテキストファイルを規格化できます。

パラメータ	設定ファイル	説明
-r-		シフト JIS のテキストファイルを規格化しない。
-e-		MS932~Shift_JIS.bin(変換表)を編集しない。
	twenty.txt	変換表の編集およびテキストの規格化を行う。

## すべて下線になる

dayu コマンドが MS932 から Shift\_JIS にテキストファイルを規格化するとき、丸付きの数字(①②③⑩⑳)、ローマ数字(ⅠⅡⅢⅣⅩ)などの機種依存文字は、すべて下線( \_ )に変更されます。1 バイトのカタカナ(半角カタカナ)は、Shift\_JIS に含まれていますが、dayu コマンドは、半角カタカナを廃止して、ASCII の下線( \_ )に変更します。

## 1 文字ずつ編集

①を下線ではなく1で代用するには、dayu コマンドのパラメータとして-m① -s1を付加してください。通常の文字および代用文字のシフト JIS 文字コードが十六進数で表示されます。

### 標準入力モードから脱出

規格化したいファイルが設定されていない場合、dayu コマンドは、自動的に標準入力から読み込んで標準出力に書き出します。標準入力のモードから脱出するには、たとえば、Ctrl を押しながら Z を押してください。^Z がコマンドプロンプトに表示されます。^Z は、ファイルの終わり(EOF)を意味します。

### 規格化しない

変換表を編集するが、シフト JIS のテキストファイルを規格化しないように命令するには、dayu コマンドのパラメータとして、-r-を付加してください。dayu コマンドは、標準入力から読み込もうとしません。

### どの変換表を編集するか

dayu コマンドがアクセスできる変換表のパス名を表示するには、コマンドのパラメータとして、-wを付加してください。パス名ではなく単なるファイル名が表示された場合、カレントディレクトリにある変換表です。

**dayu -m① -s1 -r- -w**

### 編集しない

シフト JIS のテキストファイルを規格化するが、変換表を編集しないように命令するには、dayu コマンドのパラメータとして、-e-を付加してください。



```
C:¥Q>COPY CON maruni.txt
①は、下線ではない。
②は、下線だ。
③も下線だ。
^Z
    1 個のファイルをコピーしました。

C:¥Q>dayu -m① -s1 -r- -w
C:¥sortk-C¥MS932~Shift_JIS.bin is primarily accessible to dayu
8740 => 8250

C:¥Q>DIR | FINDSTR "MS932~Shift_JIS.bin"
2019/08/15  13:16                31,528 MS932~Shift_JIS.bin

C:¥Q>COPY MS932~Shift_JIS.bin C:¥sortk-C /Y
    1 個のファイルをコピーしました。

C:¥Q>dayu -e- maruni.txt text.tmp

C:¥Q>TYPE text.tmp
1は、下線ではない。
_は、下線だ。
__も下線だ。

C:¥Q>
```

## バッチファイル

コマンドおよびパラメータが長い場合、コマンドプロンプトに入力するとき、間違いやすい。間違えないで、同様の文字列を何回も入力するのは、困難です。テキストエディタで、事前にバッチファイルを編集して、バッチファイルをコマンドプロンプトに入力して、cmd.exe に一括処理(バッチ処理)を実行させることができます。

### バッチファイルの先頭

バッチファイルの各行にコマンドを記述します。コマンドプロンプトがバッチファイルを実行するとき、バッチファイルの行が画面に表示されて、コマンドが自動的に入力されるように見えます。バッチファイルの行が表示されるのを抑制するには、バッチファイルの先頭に以下のコマンドを記述してください。

## @ECHO OFF

## バージョン番号

短い説明を表示するには、dayu コマンドのパラメータとして-VC を付加してください。

```
C:\¥Q>dayu -VC
dayu C-1.070

dayu MS932 # Read MS932 to produce dayu.txt
dayu MS932 SJIS # Read MS932 to produce a new file.
dayu < MS932 # Every line includes ODh.
dayu -b MS932 # Produce binary.tmp
dayu -b MS932 MS932 # Produce and rename binary.tmp
dayu -b -y MS932 # Do not remove binary.tmp
dayu -d MS932 # Do not search a table.
dayu -e- MS932 # Do not edit MS932~Shift_JIS.bin
dayu -f bars.txt MS932 # Edit MS932~Shift_JIS.bin according to bars.
dayu -i MS932 # Skip control characters except 0Ah.
dayu -k ナ -aN # ナ is substituted with N.
dayu -L MS932 # ODh is removed from every line.
dayu -L- MS932 # Every line includes ODh.
dayu -m① -s1 -r- # Edit MS932~Shift_JIS.bin without reading MS
dayu -m① -s1 MS932 # ① is substituted with 1.
dayu -q MS932 # Acquire a table name from twenty.txt
dayu -t MS932 SJIS # MS932 has the same time as SJIS.
dayu -w MS932 # -w shows which table dayu uses.
```

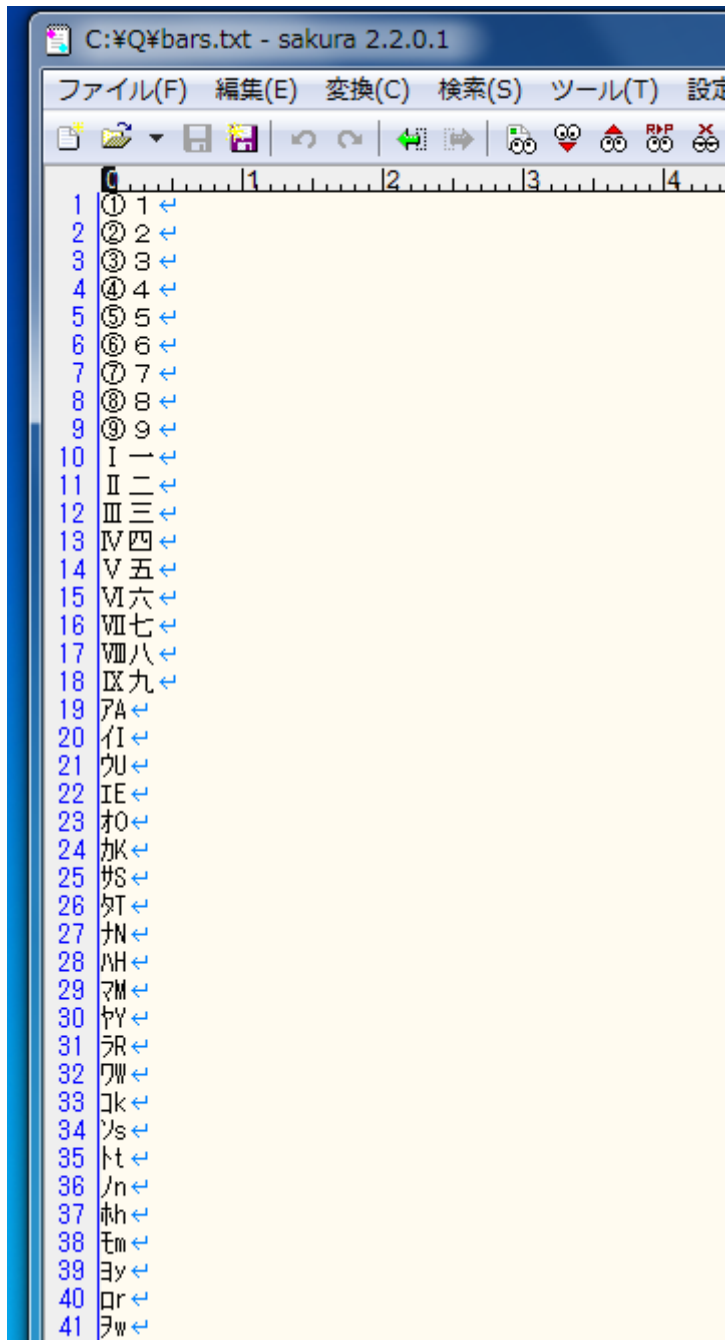
パラメータ	密接に関係するファイル	説明
-b	シフト JIS テキスト	binary.tmp を作成します。
-CRLF	シフト JIS テキスト	ODh および 0Ah で改行します。
-d	MS932~Shift_JIS.bin	変換表を探索してはならない。
-e-	MS932~Shift_JIS.bin	変換表を編集しません。
-f	代用文字のリスト。 MS932~Shift_JIS.bin	多数の代用文字にわたって変換表を編集します。
-i	シフト JIS テキスト	0Ah 以外の制御文字を削除します。
-k ナ	MS932~Shift_JIS.bin	ある半角カタカナを廃止します。
-aN	MS932~Shift_JIS.bin	ASCII の 1 文字で代用します。
-LF	シフト JIS テキスト	0Ah のみで改行します。
-m①	MS932~Shift_JIS.bin	ある機種依存文字を廃止します。
-s1	MS932~Shift_JIS.bin	Shift_JIS の 1 文字で代用します。
-r-	シフト JIS テキスト	シフト JIS テキストを規格化しません。
-w	MS932~Shift_JIS.bin	どの変換表を使用するか表示します。
-y	シフト JIS テキスト	binary.tmp を削除してはならない。

## 代用文字のリスト

事前に利用者が代用文字のリストを用意した場合、dayu コマンドは、代用文字のリストにしたがって、一度に多数の文字を編集できます。

### 通常文字および代用文字

dayu コマンドは、代用文字リストのファイルから 906 行まで読み込むことができます。代用文字リストのファイルの各行は、通常文字と、代用文字とからなります。



## ファイル名を指定

利用者が代用文字リストのファイルを用意した場合、`dayu` コマンドのパラメータとして、`-f`を付加して、空白で区切って、さらにファイル名を付加してください。

```
dayu -f bars.txt -r- -w
```

```
C:¥Q>dayu -f bars.txt -r- -w
C:¥sortk-C¥MS932~Shift_JIS.bin is primarily accessible to dayu

C:¥Q>DIR | FINDSTR "MS932~Shift_JIS.bin"
2019/08/15 13:22          31,528 MS932~Shift_JIS.bin

C:¥Q>COPY MS932~Shift_JIS.bin C:¥sortk-C /Y
      1 個のファイルをコピーしました。

C:¥Q>COPY CON haishi.txt
①②③も I II III も下線ではない。
アイエもアカクもカキクも下線ではない。
^Z
      1 個のファイルをコピーしました。

C:¥Q>dayu -e- haishi.txt text.tmp

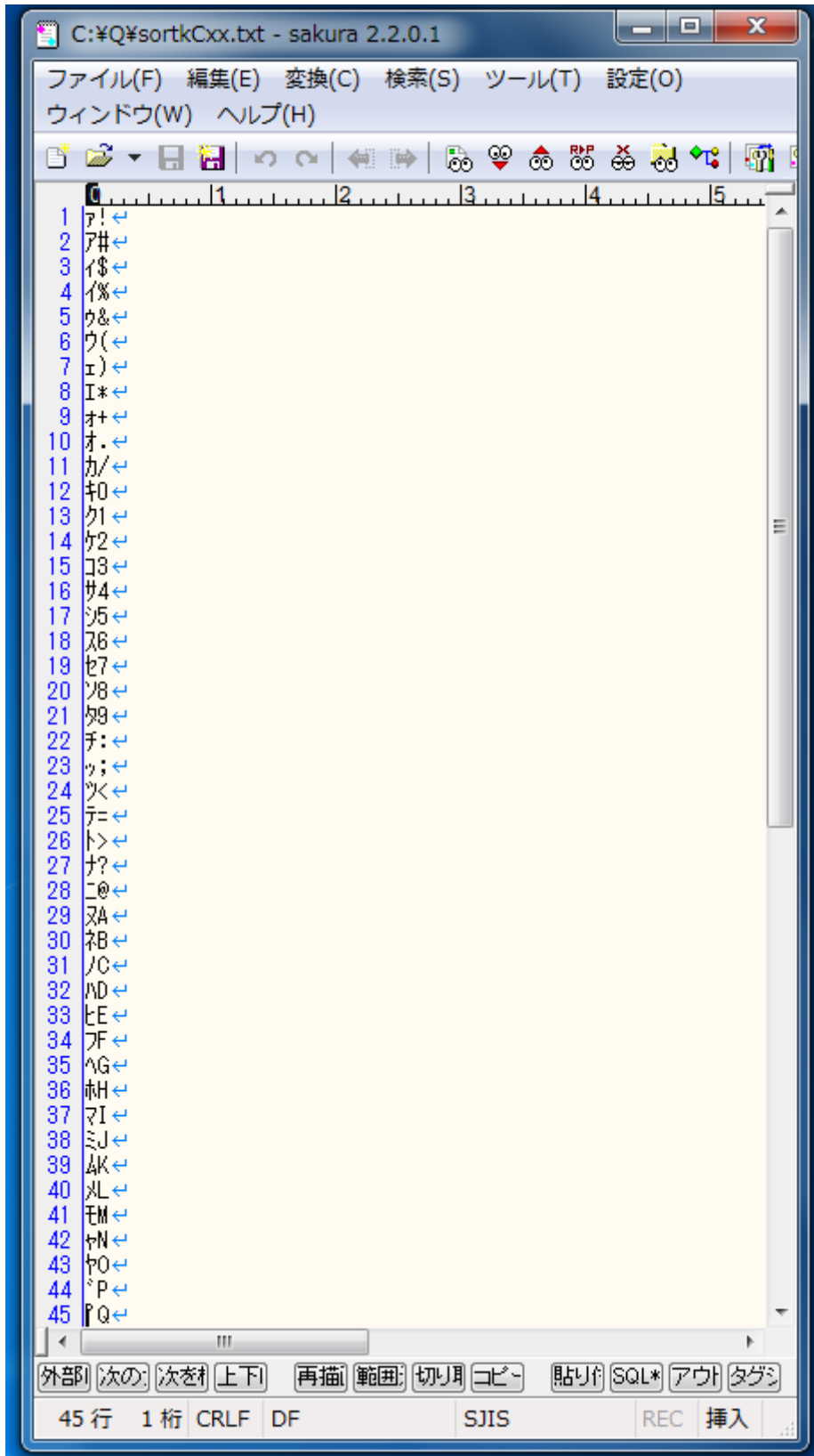
C:¥Q>TYPE text.tmp
 1 2 3 も一 二 三 も下線ではない。
AIUEO も AKSTN も Okstnhmyrw も下線ではない。
C:¥Q>
```

## 並べ替えに適した代用文字リスト

上記の `bars.txt` では、ソフトウェアで並べ替えた結果が昇順になりません。代用文字リストのファイルを作成するとき、並べ替えも考慮したい。

### sortk コマンドに適した代用文字リスト

機種(キシユ)、著作権(チョサクケン)などの拗音を直音の付近に並べ、実行可能(ジッコウカノウ)、接頭辞(セツウジ)などの促音を直音の付近に並べ、合図(アイズ)、英語(エイゴ)などの濁音を清音の付近に並べ、添付(テンブ)などの半濁音を清音の付近に並べるのに適した代用文字リストが必要です。C++版の `sortk` コマンドに適した代用文字リストは、`sortkCxx.txt` です。Cxx は、C++ を意味します。半角カタカナの濁点を廃止して、P で代用します。



## 第 2 欄を比較

sortk コマンドは、たとえば、第 2 欄にある語句を比較して、CSV のテキストファイルの行を並べ替えることができます。第 2 欄にある語句を比較するには、sortk コマンドのパラメータとして、-k2,2 を付加してください。-k および数字の間に空白を挿入しないでください。半角カタカナの代用として ASCII を使用している例を示します。sortkCxx.txt にしたがって、dayu コマンドが変換表を編集した例(-p については、後述します)

```
C:¥Q>dayu -f sortkCxx.txt -r- -w
C:¥sortk-C¥MS932~Shift_JIS.bin is primarily accessible to dayu

C:¥Q>DIR | FINDSTR "MS932~Shift_JIS.bin"
2019/08/15 13:29          31,528 MS932~Shift_JIS.bin

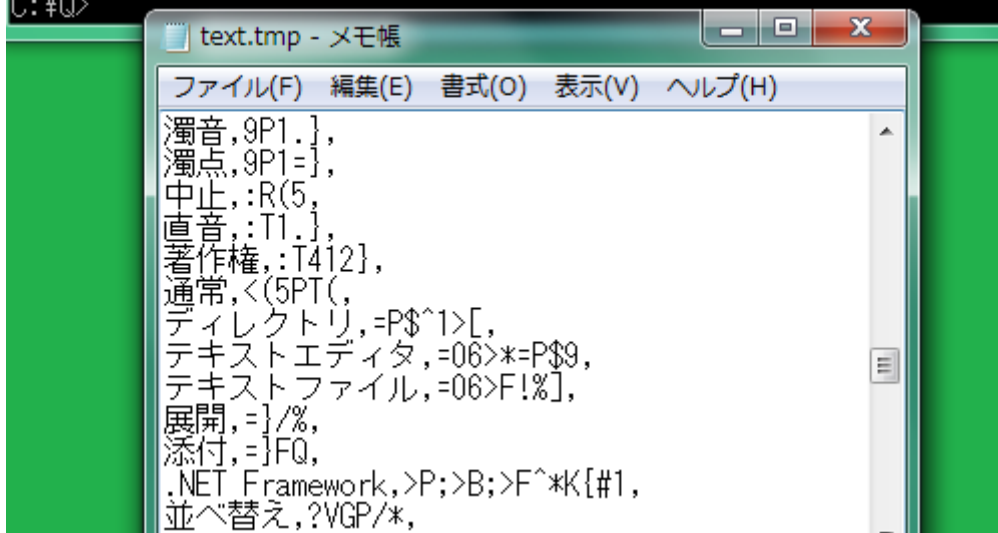
C:¥Q>COPY MS932~Shift_JIS.bin C:¥sortk-C /Y
1 個のファイルをコピーしました。

C:¥Q>dayu -e- KeyWords.csv KeyWords_ascii.csv

C:¥Q>sortk -t, -k2,2 -c- -p -o text.tmp KeyWords_ascii.csv

C:¥Q>NOTEPAD text.tmp

C:¥Q>
```



### 濁点を無視

半角カタカナは、清音のカナおよび濁点を組み合わせることで濁音を表現しますが、清音および濁音は、ほぼ同等の位置になるように並べ替えたい。上記の例のとおり、sortk コマンドが、濁点、半濁点を無視するように、sortk コマンドのパラメータとして、-c- を付加してください。dayu コマンドで、半角カタカナの代用として ASCII を使用して

いる場合、さらに、-p を付加してください。

### 新しいファイルを命名

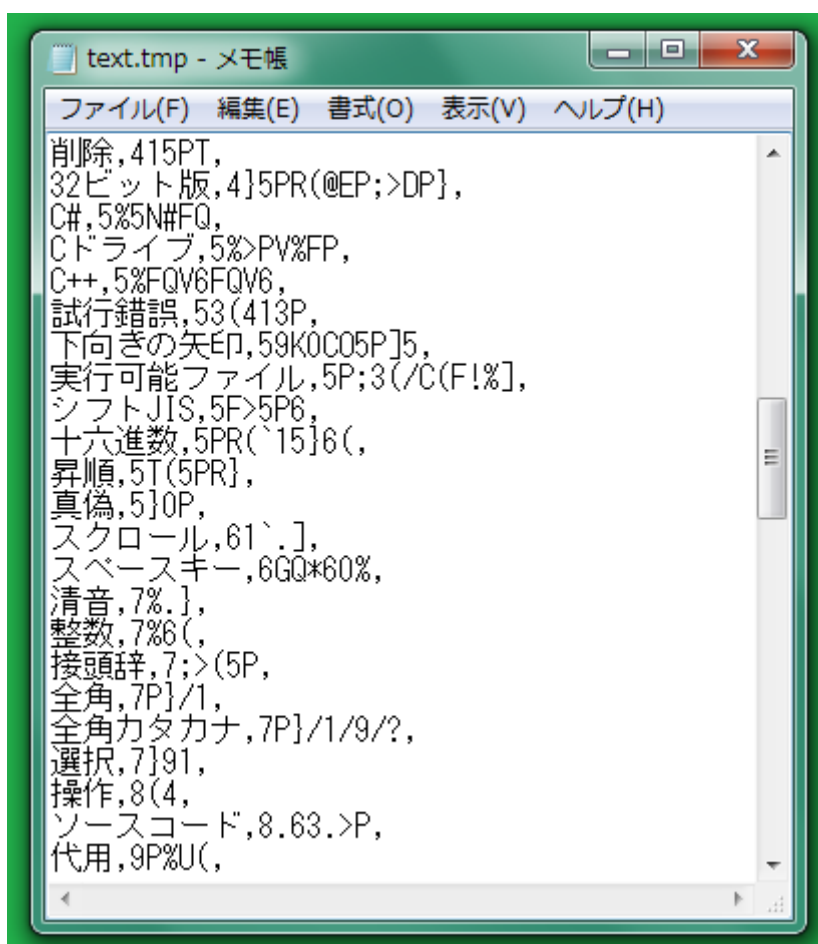
新しいファイルを作成するには、sortk コマンドのパラメータとして、-o text.tmp を付加してください。text.tmp は、新しいファイルの名前です。

### 五十音順

まず、-r が付加された dayu コマンドは、sortkCxx.txt にしたがって変換表を編集しました。次に、-e が付加された dayu コマンドは、シフト JIS テキストファイルを規格化しました。半角カタカナの読み方は、ASCII に置換されました。さらに、sortk コマンドは、第 2 欄に関して行を並べ替えました。

### 確認する例

たしかに五十音順に並べ替えることができました。たとえば、ディレクトリは、テキストエディタの上に来ました。もうひとつの例として、実行可能ファイルは、**下向きの矢印**の下に、シフト JIS の上に来ました。



## 設定ファイル

dayu の設定ファイルは、twenty.txt です。利用者がカレントディレクトリから

twenty.txt を削除しない限り、dayu は、twenty.txt にしたがって設定します。twenty.txt は、15 行からなります。第 1 行にパス名ではなくファイル名が記入されており、第 6 行の設定内容が空である例

行	設定内容	コメント	説明
1	MS932~Shift_JIS.bin	table	新しい変換表のファイル名またはパス名
2	KeyWords.csv	MS932	規格化したいテキスト
3	dayu.txt	Shift_JIS	規格化されたテキスト
4	㊿	normal	機種依存文字。通常文字
5	廿	substitute	代用文字
6		bars.txt	代用文字のリスト
7	0	-b	binary.tmp を作成する
8	-CRLF	-CRLF	0Dh および 0Ah で改行する
9	0	-d	変換表を探索しない
10	0	-e-	変換表を編集しない
11	0	-i	制御文字を削除する
12	0	-r-	テキストを規格化しない
13	0	-t	規格化前後のファイルは、同じ日時を有する
14	0	-w	どの変換表を使用するか表示する
15	0	-y	binary.tmp を削除してはならない
16	C:¥ MS932~Shift_JIS.bin	priority	利用者指定の変換表のパス名

## 新しい変換表のパス名

dayu が twenty.txt を開くことができた場合も、設定を dayu コマンドのパラメータで変更できます。ただし、dayu コマンドのパラメータでは、twenty.txt の第 1 行に記載の新しい変換表のファイル名またはパス名を変更できません。**コマンドがインストールされたディレクトリにある変換表**を新しい変換表のパス名として記入した例

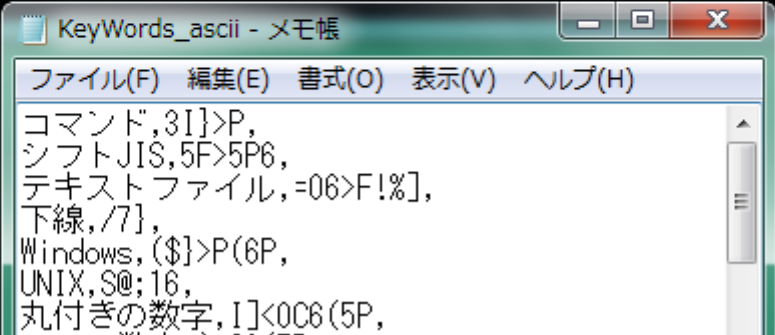


```
C:¥Q>TYPE twenty.txt
C:¥sortk-C¥MS932~Shift_JIS.bin table
KeyWords.csv MS932
KeyWords_ascii.csv Shift_JIS
① normal
1 substitute
sortkCxx.txt bars.txt
0 -b
-CRLF -CRLF
0 -d
0 -e-
0 -i
0 -r-
0 -t
-w -w
0 -y
C:¥MS932~Shift_JIS.bin priority

C:¥Q>dayu
C:¥sortk-C¥MS932~Shift_JIS.bin is primarily accessible to dayu

C:¥Q>DIR KeyWords_ascii.csv | FINDSTR "2019/08"
2019/08/15 14:37 2,255 KeyWords_ascii.csv

C:¥Q>NOTEPAD KeyWords_ascii.csv
C:¥Q>
```



### 変換表が上書きされる

コマンドがインストールされたディレクトリにある変換表が新しい変換表のパス名として記入されている場合、dayu コマンドは、利用者に警告しないで変換表を上書きします。

### 利用者が上書きする

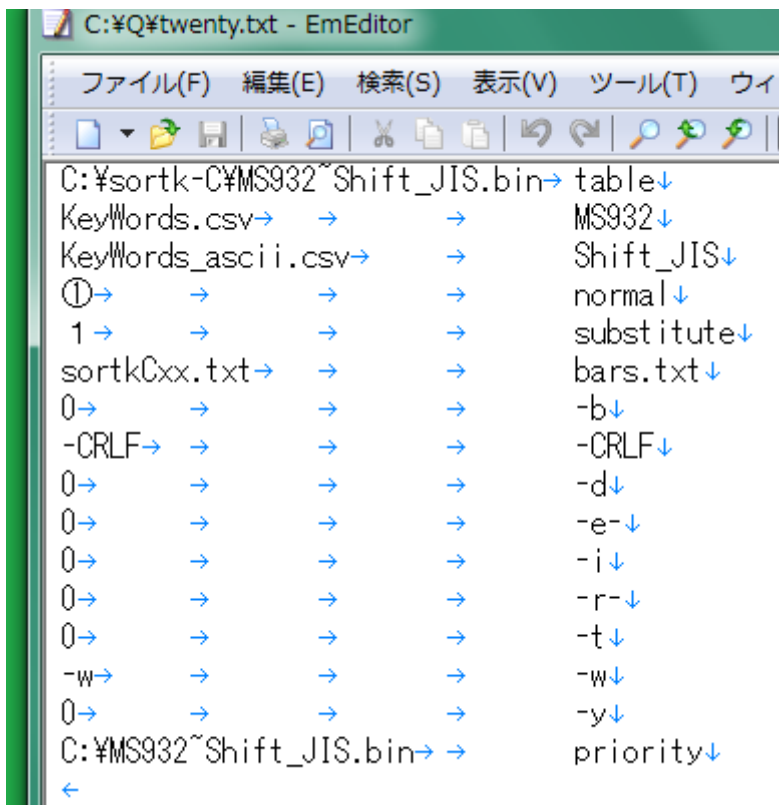
twenty.txt の第 1 行にパス名ではなくファイル名が記入されている場合、dayu コマンドは、カレントディレクトリに新しい変換表を作成します。dayu がアクセスできるディレクトリに MS932~Shift\_JIS.bin をコピーしてください(上書きしてください)。

### 設定ファイルは必須ではない

twenty.txt の第 1 行に新しい変換表のパス名またはファイル名を記入できますが、設定ファイルを開くことができない場合、dayu コマンドは、利用者に警告しないで、カレントディレクトリに新しい変換表を作成します。dayu がアクセスできるディレクトリに MS932~Shift\_JIS.bin をコピーしてください。

### 設定内容、タブ、コメントの順に記入

設定内容を左に詰めて記入してください。設定内容のみで改行できますが、シングルクォーツ、縦線、タブの次にコメントを記入できません。Perl, PHP, Python, Ruby などのスクリプト言語とは異なり、dayu は、シャープ( # )の次にコメントを記入できません。空白(20h)の次にコメントを記入できません。空白ではなくタブを利用してください。Windows のメモ帳では、タブが機能しますが空白に見えます。EmEditor は、右向き矢印でタブを表示できます。



```
C:\¥Q¥twenty.txt - EmEditor
ファイル(F) 編集(E) 検索(S) 表示(V) ツール(T) ウィ
C:\¥sortk-C¥MS932~Shift_JIS.bin→ table↓
Keywords.csv→ → → MS932↓
Keywords_ascii.csv→ → Shift_JIS↓
①→ → → normal↓
1→ → → substitute↓
sortkCxx.txt→ → → bars.txt↓
0→ → → -b↓
-CRLF→ → → -CRLF↓
0→ → → -d↓
0→ → → -e↓
0→ → → -i↓
0→ → → -r↓
0→ → → -t↓
-w→ → → -w↓
0→ → → -y↓
C:\¥MS932~Shift_JIS.bin→ → priority↓
←
```

### サクラエディタ

サクラエディタでは、タブが脱字記号(caret[ ^ ])で表示されます。

### MIFES

MIFES は、大なり記号( > )でタブを表示できます。

## 0 ではないならば有効

0 を記入すると無効になります。0Dh および 0Ah で改行するには、第 8 行に、0 ではない設定として、-CRLF を記入してください。

## 第 6 行の設定内容が空である

代用文字リストのファイル名が空である場合、多数の代用文字にわたって変換表を編集する機能は、無効になります。第 4 行および第 5 行にしたがって、1 文字を編集します。

### 第 6 行に sortkCxx.txt を記入

代用文字リストのファイル名が空ではない場合、第 6 行にしたがって、多数の代用文字にわたって変換表を編集します。第 6 行に sortkCxx.txt を記入した例

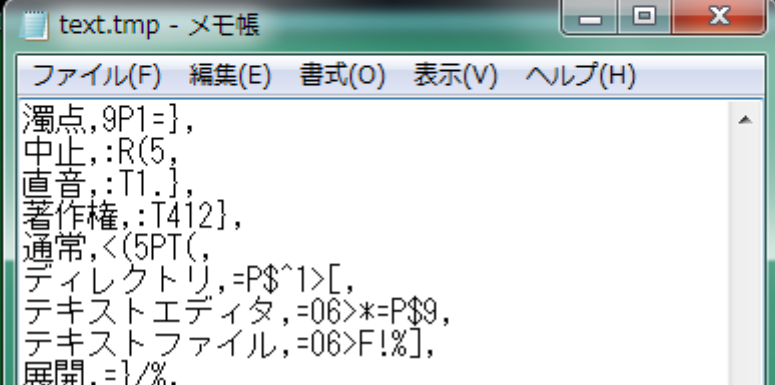
```
C:¥Q>TYPE twenty.txt
C:¥sortk-C¥MS932~Shift_JIS.bin table
Keywords.csv MS932
Keywords_ascii.csv Shift_JIS
① normal
1 substitute
sortkCxx.txt bars.txt
0 -b
-CRLF -CRLF
0 -d
0 -e
0 -i
0 -r
0 -t
-w -w
0 -y
C:¥MS932~Shift_JIS.bin priority

C:¥Q>dayu
C:¥sortk-C¥MS932~Shift_JIS.bin is primarily accessible to dayu

C:¥Q>sortk -t, -k2,2 -c- -p -o text.tmp Keywords_ascii.csv

C:¥Q>NOTEPAD text.tmp

C:¥Q>
```



```
text.tmp - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
濁点,9P1=},
中止,:R(5,
直音,:T1.},
著作権,:T412},
通常,<(5PT(,
ディレクトリ,=P$^1>[,
テキストエディタ,=06>*=P$9,
テキストファイル,=06>F!%},
展開,=}/%,
```

## ファイルの日時

規格化したいファイルの日時と、規格化されたファイルの日時が同一になるように命令するには、dayu コマンドのパラメータとして-t を付加してください。

```
C:¥Q>dayu -e- -t haishi.txt haishi_sjis.txt

C:¥Q>DIR haishi*.txt | FINDSTR "2019/05"
2019/05/25 13:57          74 haishi.txt
2019/05/25 13:57          74 haishi_sjis.txt

C:¥Q>
```

## パラメータで設定ファイルを否定できる

たとえば、twenty.txt の第 14 行に-w を記入したにもかかわらず、どの変換表を使用するか表示しないように再設定するには、dayu コマンドのパラメータとして、-w を付加してください。

### 設定ファイルを削除

設定ファイルに記入した設定をパラメータの入力で変更できないと思われる場合、レントディレクトリから設定ファイルを削除してください。-L は、-LF を同じ結果を生じません。-L- は、-CRLF と同じ結果を生じます。

説明	記入	否定する入力	説明
変換表を探索してはならない。	-d	-d-	変換表を探索する。
制御文字を削除する。	-i	-i-	制御文字を削除しない。
規格化の前後でファイルの日時が同一である。	-t	-t-	規格化の前後でファイルの日時が相違している。
変換表のパス名を表示する。	-w	-w-	変換表のパス名を表示しない。
変換表を編集しない。	-e-	-e	変換表を編集する。
シフト JIS テキストを規格化しない。	-r-	-r	シフト JIS テキストを規格化する。
0Dh および 0Ah で改行する。	-CRLF	-LF	0Ah のみで改行する。
0Dh を削除する。	-L	-L-	0Dh を削除しない。

## ID として行番号を使用する

下記の例では、まず、-# を付加しない sortk コマンドを実行しました。次に、要約に ID を付与するために -# を付加して sortk コマンドを実行しました。第 10 欄から第 20 欄まで比較して順序が確定しない場合も、行番号で順序が確定します。atnd -h は、UNIX の head コマンドに似ています。

```
C:\%Q>sortk -t, -k10,20 Field-22.csv | atnd -h23
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,12,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,8,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,13,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,1,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,15,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,17,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,18,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,11,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,7,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,2,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,35,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,34,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,39,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,33,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,37,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,38,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,36,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,19,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,20,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,21,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,22,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,23,
```

```
C:\%Q>sortk -t, -k10,20 -# Field-22.csv | atnd -h23
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,12,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,8,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,13,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,1,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,15,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,17,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,18,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,11,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,7,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,2,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,19,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,20,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,21,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,22,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,23,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,24,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,25,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,26,
```

#### 要約が合致した行番号

要約が合致した行の番号を出力するには、-# -!を付加してください。

```
C:\%Q>sortk -k1,7 -g -# -! Field-9.csv
1.0,2.0,3.0,4.0,5.0,6.0,7.0,12.0,
1.0,2.0,3.0,4.0,5.0,6.0,8.0,8.0,
1.0,2.0,3.0,4.0,5.0,6.0,10.0,13.0,
1.0,2.0,3.0,4.0,5.0,6.0,20.0,1.0,
1.0,2.0,3.0,4.0,5.0,6.0,22.0,15.0,
1.0,2.0,3.0,4.0,5.0,6.0,26.0,17.0,
1.0,2.0,3.0,4.0,5.0,6.0,27.0,10.0,
1.0,2.0,3.0,4.0,5.0,6.0,29.0,18.0,
1.0,2.0,3.0,4.0,5.0,6.0,44.0,11.0,
1.0,2.0,3.0,4.0,5.0,6.0,49.0,7.0,
1.0,2.0,3.0,4.0,5.0,6.0,51.0,2.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,19.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,20.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,21.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,22.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,23.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,24.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,25.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,26.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,27.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,28.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,29.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,30.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,31.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,32.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,33.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,34.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,35.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,36.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,37.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,38.0,
1.0,2.0,3.0,4.0,5.0,6.0,59.0,39.0,
1.0,2.0,3.0,4.0,5.0,6.0,60.0,3.0,
1.0,2.0,3.0,4.0,5.0,6.0,62.0,40.0,
1.0,2.0,3.0,4.0,5.0,6.0,63.0,14.0,
1.0,2.0,3.0,4.0,5.0,6.0,69.0,16.0,
1.0,2.0,3.0,4.0,5.0,6.0,73.0,4.0,
1.0,2.0,3.0,4.0,5.0,6.0,77.0,9.0,
1.0,2.0,3.0,4.0,5.0,6.0,88.0,5.0,
1.0,2.0,3.0,4.0,5.0,6.0,99.0,6.0,
Line-39
Line-38
Line-37
Line-36
```

### 最後に表示される

本来の出力が表示されたのちに、行番号が表示されます。

### 欄をどう選択するか

本来、sortk コマンドは、要約も ID も標準出力に書き出しません。しかし、欄をどう選択すればよいか判断するために、利用者は、ある行の要約が他の行の要約と合致したかどうか知りたいかもしれません。要約が合致した行の番号を出力するには、sortk コマンドのパラメータとして、-! を付加してください。本来の出力が表示されたのちに、行番号が表示されます。

### 記憶力の限界

要約が合致したとき、sortk コマンドは、行の番号を記憶します。記憶力には、限界があります。記憶した行が重複している場合があります。のべ 128 行まで記憶できます。

### 要約ごとに 1 行を表示

要約が合致した行がある場合、その要約は、1 行の詳細のみ表示するには、sortk コマンドのパラメータとして -u を付加してください。

```
C:¥Q>sortk -t, -k10,20 -u Field-22.csv
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,12,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,8,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,13,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,1,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,15,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,17,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,18,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,11,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,7,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,2,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,35,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,60,3,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,62,40,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,63,14,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,69,16,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,73,4,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,77,9,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,88,5,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,99,6,
C:¥Q>
```

### DMC 版の sortk コマンド

DMC 版の sortk コマンドは、STLport 4.5.3 を使用します。DMC 版 sortk コマンドの特徴として、順序が確定しない場合、-# を付加しない限り、行の登録が拒否されます。すなわち、-u が付加された sortk コマンドとほぼ同じ結果を生じます。DMC 版 sortk コマンドの -u は、-# と同じ結果を生じます。



sortk	ライブラリ	パラメータ	同等な入力	説明
C#	.NET Framework 3.5	-u		要約ごとに 1 行ぶんのテキストデータを出力するため、sutai.xml のサイズを縮小できる。
		-u -i		並べ替えた後、要約ごとに 1 行のみ表示する。
C		-u		
C++		-u		
		-~		要約ごとに 1 行ぶんのバイナリデータを出力するため、binary.tmp のサイズを縮小できる。
DMC	STLport 4.5.3	なし		
		-u-	-#	ID が要約に含まれるため、要約ごとに多数の行を表示できる。

## binary.tmp を再利用する


sortk コマンドは、いったん binary.tmp というバイナリファイルを作成して、binary.tmp を読み込んで、要約および詳細のブロックを並べ替えて、標準出力に詳細のみ書き出します。通常は、sortk コマンドは、自動的に binary.tmp を削除します。削除しないように命令するには、sortk コマンドのパラメータとして-y を付加してください。

### 設定ファイル

添付の keyun コマンドは、binary.tmp を再利用できます。keyun コマンドの設定ファイルを書き出すには、sortk コマンドのパラメータとして-%を付加してください。

```
sortk -t, -k2,2 -y -% -o text.tmp isbn-r.csv
```

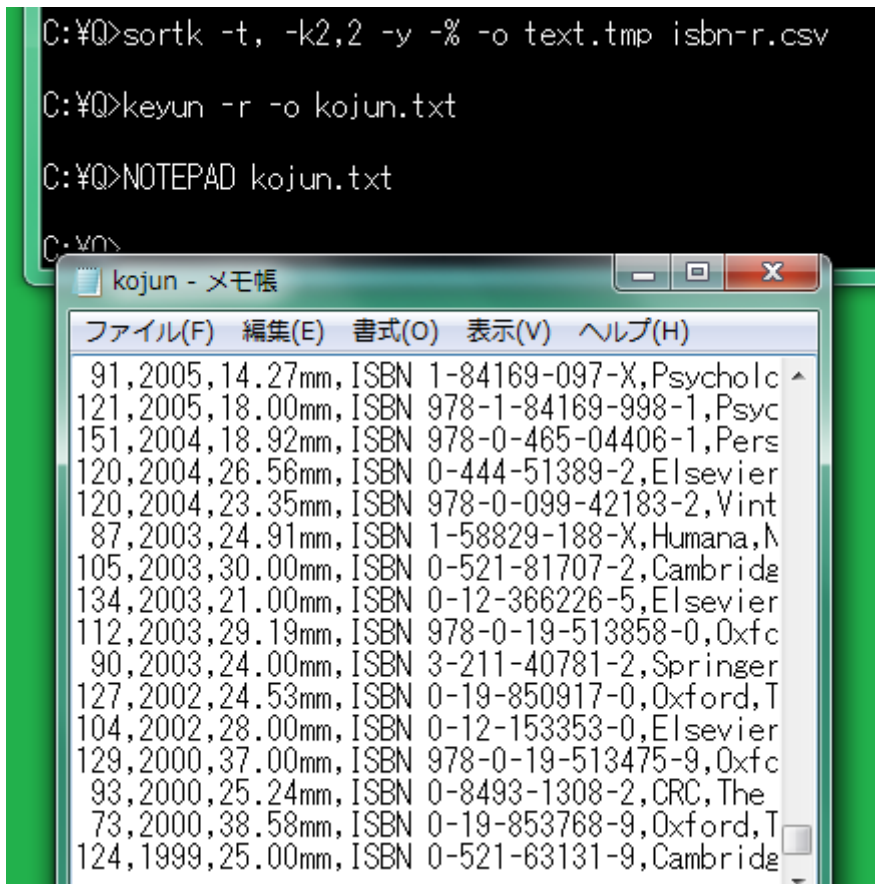
```
C:¥Q>sortk -t, -k2,2 -y -% -o text.tmp isbn-r.csv
C:¥Q>keyun -o shojun.txt
C:¥Q>NOTEPAD shojun.txt
C:¥Q>
```



Year	Thickness	ISBN	Publisher
124	25.00mm	0-521-63131-9	Cambridge
73	38.58mm	0-19-853768-9	Oxford, T
93	25.24mm	0-8493-1308-2	CRC, The
129	37.00mm	978-0-19-513475-9	Oxford, T
104	28.00mm	0-12-153353-0	Elsevier
127	24.53mm	0-19-850917-0	Oxford, T
90	24.00mm	3-211-40781-2	Springer
112	29.19mm	978-0-19-513858-0	Oxford, T
134	21.00mm	0-12-366226-5	Elsevier
105	30.00mm	0-521-81707-2	Cambridge
87	24.91mm	1-58829-188-X	Humana, N
120	26.56mm	0-444-51389-2	Elsevier
120	23.35mm	978-0-099-42183-2	Vint
151	18.92mm	978-0-465-04406-1	Pers
121	18.00mm	978-1-84169-998-1	Psyc
91	14.27mm	1-84169-097-X	Psycholc

### 降順

keyun コマンドの設定ファイルは、manager.txt です。manager.txt にしたがって、keyun コマンドは、binary.tmp を読み込んで、行を並べ替えます。下記の例では、降順に並べ替えるために、keyun コマンドのパラメータとして-r を付加しました。



パラメータ	ほぼ同等な入力	説明
-e		0Ah のみで改行する。
-f		大文字と小文字を区別しない。
-o new	-o- > new	binary.tmp を読み込んで、新しいファイルを作成する
-o-		標準出力に書き出す。
-r		降順に並べ替える。
-s --help		画面からファイルに、短い説明をリダイレクトできる。
-u		要約ごとに 1 行のみ表示する。
-y		binary.tmp を削除してはならない。
-V		バージョン番号を表示する。
-VC	--help	パラメータの例および短い説明を表示する。


### 標準出力

設定ファイルに記入の新しいファイル名を無効にして、標準出力に書き出すには、keyun コマンドのパラメータとして -o- を付加してください。

### 要約ごとに1行のみ表示

要約が合致した行がある場合、その要約は、1行のみ表示するには、keyun コマンドのパラメータとして-u を付加してください。下記の例では、要約が合致した行がありません。第20欄に59が記入されています。まず、sortk コマンドで、要約が合致した多数の行をtext.tmpに書き出しました。次に、-uを付加したkeyunコマンドで、要約ごとに1行のみ表示しました。

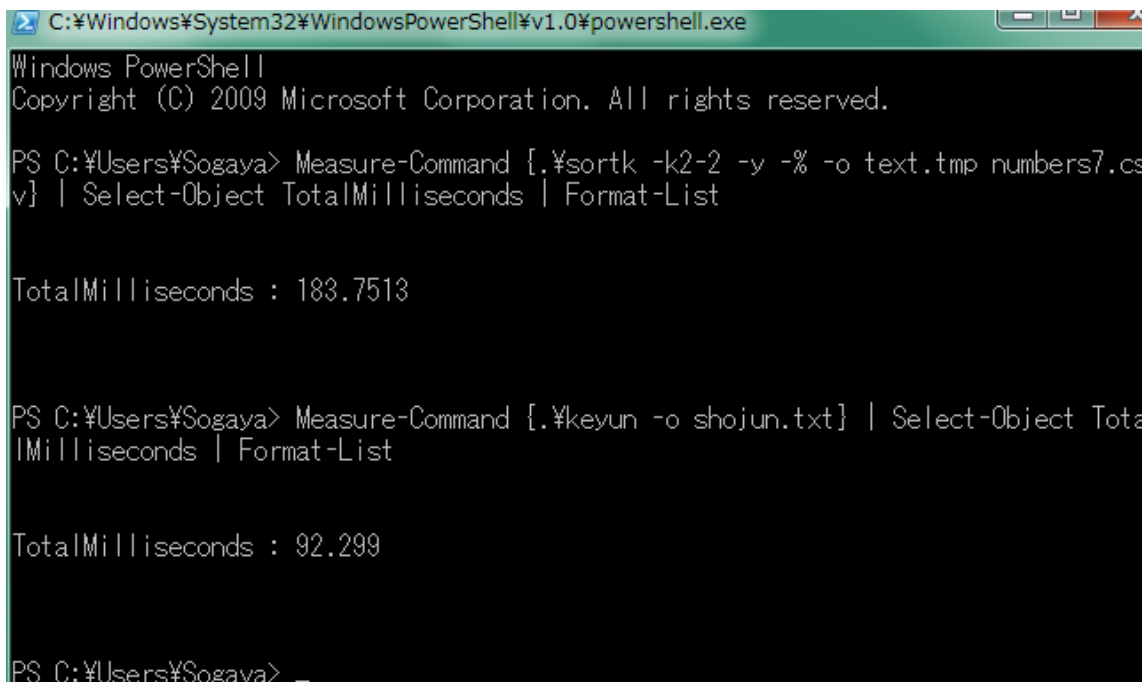
```
C:¥Q>sortk -t, -k10,20 -y -% -o text.tmp Field-22.csv
C:¥Q>NOTEPAD text.tmp
C:¥Q>keyun -u -o-
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,12,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,8,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,13,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,1,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,15,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,17,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,18,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,11,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,7,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,2,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,35,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,60,3,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,62,40,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,63,14,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,69,16,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,73,4,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,77,9,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,88,5,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,99,6,
C:¥Q>
```



## 実行時間を測定してみた

PowerShell の Measure-Command コマンドレットで、sortk コマンドおよび keyun コマンドの実行時間をミリ秒の単位で測定してみました。sortk.exe, keyun.exe, numbers7.csv を Sogaya ディレクトリにコピーして、Measure-Command コマンドレットのスクリプトブロックに下記のコマンドおよびパラメータを入力しました。

```
.%sortk -k2-2 -y -% -o text.tmp numbers7.csv
```



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Sogaya> Measure-Command {.%sortk -k2-2 -y -% -o text.tmp numbers7.csv} | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 183.7513

PS C:\Users\Sogaya> Measure-Command {.%keyun -o shojun.txt} | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 92.299

PS C:\Users\Sogaya> _
```

### keyun コマンドは無駄が無い

keyun コマンドは、binary.tmp を再利用するため、要約および詳細のブロックを作成する工程を省略できます。keyun コマンドは、sortk コマンドより速いことがわかります。

### パラメータが優先される

keyun コマンドの設定ファイルは、manager.txt です。manager.txt にかかわらず、keyun コマンドは、パラメータを優先します。たとえば、binary.tmp を再利用するとき、manager.txt の第10行にかかわらず、-rが付加されたkeyunコマンドは、降順に並べ替えます。

## ファイルの行数

添付の atnd コマンドは、ファイルの行数をカウントできます。ファイルの行数を表示するには、atndコマンドのパラメータとして、-LFを付加してください。ファイル名および行数が表示されます。また、最長の行の長さが記録されており、バイト数および行番号

が表示されます。さらに、第 1 行が表示されます。第 1 行ではなく第 9 行を表示するには、atnd コマンドのパラメータとして、-p9 を付加してください。

```
C:¥Q>atnd -LF -p9 nineTen.txt
nineTen.txt: 17 lines
maximum LF-to-LF span of 30 bytes (line-9)
123456789,1234567890,123456789
C:¥Q>
```

#### atnd コマンドの-t

ファイルの終了部分にある 5 行を表示するには、atnd コマンドのパラメータとして、-t5 を付加してください。atnd コマンドは、自動的にファイルの行数をカウントします。たとえば、1024 行である場合、-t5 は、-b4 -p1024 を意味します。atnd -t は、UNIX の tail コマンドに似ています。

```
C:¥Q>atnd -LF LineNo.txt
LineNo.txt: 1024 lines
maximum LF-to-LF span of 34 bytes (line-1000)
Line-1: 639 lines to Line-640

C:¥Q>atnd -b4 -p1024 LineNo.txt
Line-1020: 380 lines from Line-640
Line-1021: 381 lines from Line-640
Line-1022: 382 lines from Line-640
Line-1023: 383 lines from Line-640
Line-1024: 384 lines from Line-640

C:¥Q>atnd -t5 LineNo.txt
Line-1020: 380 lines from Line-640
Line-1021: 381 lines from Line-640
Line-1022: 382 lines from Line-640
Line-1023: 383 lines from Line-640
Line-1024: 384 lines from Line-640

C:¥Q>_
```

パラメータ	ほぼ同等な入力	説明
-a2		後続の 2 行を表示する。
-b3		先行する 3 行を表示する。
-c4		先行する 4 行および後続の 4 行を表示する。
-h8	-b7 -p8	第 1 行から第 8 行まで表示する。
-i90-123	-b33 -p123	第 90 行から第 123 行まで表示する。
	-p90 -a33	
-l		何バイトのバッファーを使用するか表示する。
-LF		行数をカウントする。最長の行のバイト数を記録する。
-LF -=DIR		ラベルとして DIR: を表示する。
-LF -s,		コマンドの間にある欄の長さの最大値を記録する。
-LF -s¥x7C		2 桁の十六進数で区切り文字を指定する。
-n		行の先頭に番号を表示する。
-p567		第 567 行を表示する。
-t5	-b4 -p1024	ファイルの終わりにある 5 行を表示する。ファイルが 1024 行である場合、-b4 -p1024 と同じ結果になる。
-V		バージョン番号を表示する。
-VC		バージョン番号、短い説明を表示する。
-z80		80 バイトのバッファーを使用する。

#### 標準入力から読み込む

ファイル名が指定されていない場合、atnd コマンドは、標準入力(stdin)から読み込みます。利用者は、情報源(標準入力)をラベルとして命名できます。下記の例では、DIR というラベルを入力しました。

#### ファイルにコマンドの出力をリダイレクト

下記の例では、まず、DIR コマンドの出力を画面から一時ファイル(text.tmp)に、リダイレクトしました。次に、ファイルの行数をカウントするために-LF を付加して atnd コマンドを実行しました。ファイル名として、text.tmp が表示されました。

#### ファイル名ではなく stdin を表示

一時ファイルに、リダイレクトするのではなく、DIR コマンドの出力を標準入力として利用できます。すなわち、DIR コマンドの出力を atnd コマンドに入力するパイプを構成しました。情報源として、ファイル名ではなく stdin が表示されました。

### stdin ではなく DIR を表示

最後に、stdin ではなく DIR が表示されるように、-=DIR を付加した atnd コマンドによるパイプを構成しました。

```
C:¥s jisEUCjpCxx>DIR *.txt > text.tmp

C:¥s jisEUCjpCxx>atnd -LF -p11 text.tmp
text.tmp: 14 lines
maximum LF-to-LF span of 67 bytes (line-14)
2019/01/31 11:22          340 sortkCxx.txt

C:¥s jisEUCjpCxx>DIR *.txt | atnd -LF -p11
stdin: 14 lines
maximum LF-to-LF span of 67 bytes (line-14)
2019/01/31 11:22          340 sortkCxx.txt

C:¥s jisEUCjpCxx>DIR *.txt | atnd -LF -p11 -=DIR
DIR: 14 lines
maximum LF-to-LF span of 67 bytes (line-14)
2019/01/31 11:22          340 sortkCxx.txt

C:¥s jisEUCjpCxx>
```

### obot コマンドでファイルの行数をカウント

obot コマンドでファイルの行数をカウントするには、-LF を付加してください。ただし、atnd コマンドは、obot コマンドより速いらしい。

```
C:¥Q>obot -VC
obot C-1.070

obot < text          # Produce standard_input_text.tmp
obot -b2048 text     # Read 2048 bytes.
obot -c20 text       # Produce a table that has 20 fields or less.
obot -LF text        # Count lines.
obot -m3 text        # 3 lines can be represented in hexadecimal as 20
obot -m3 -s¥x2C text # CSV
obot -m45 -r20 text  # Produce a table that has 20 rows or less.
obot exe             # EXE can be represented in hexadecimal as 4D 5A.
obot -V              # version
obot -VC             # version with comments
obot -y              # Do not remove standard_input_text.tmp

C:¥Q>obot -LF numbers7.csv
5156 lines
```