

Sjisutf8 ユーザーズガイド

sjisutf8 コマンドは、シフト JIS のテキストファイルを UTF-8 のテキストファイルに変換します。

0Ah のみで改行

Windows と異なり、UNIX は、テキストファイルが 0Ah のみで改行されます。sjisutf8 コマンドは、UTF-8 のテキストファイルを 0Ah のみで改行できます。

原稿どおりに改行

HTML など、シフト JIS のテキストファイルが 0Ah のみで改行されていれば UTF-8 のテキストファイルを 0Ah のみで改行し、シフト JIS のテキストファイルが 0Dh および 0Ah で改行されていれば UTF-8 のテキストファイルを 0Dh および 0Ah で改行するように設定できます。

0Dh および 0Ah で改行

UTF-8 のテキストファイルのいかなる行も 0Dh および 0Ah で改行するように設定できます。

標準入力

Windows のコマンドプロンプトでは、シフト JIS が多用されます。DIR コマンド、TYPE コマンド、FINDSTR コマンドなど、標準出力を使用するコマンドと、sjisutf8 コマンドは、パイプを構成できます。

新しいファイル

標準出力ではなく、新しいファイルを作成します。ファイル名が指定されなかった場合、sjisutf8 コマンドは、sjisutf8.txt を作成します。

命名

sjisutf8.txt を改名するのではなく、利用者は、新しいファイルの作成および命名を行うことができます。

制御文字を削除

タブ、改ページ、エスケープシーケンスの切替え (09h, 0Ch, 1Bh) などの制御文字を削除できます。

インストール

sjisutf8 コマンドは、Windows のコンソールアプリケーションです。C++ という言語で sjisutf8 コマンドのソースコードを書きました。32 ビット版の Windows XP Home において、Dev-C++ 4.9.9.2 という IDE で sjisutf8 コマンドを構築しました。トリシーカーという

ウェブサイトから `Sjisutf8.zip` をダウンロードできます。

パスを通す手順の例

1. トリシーカー (<http://tori.tobihiro.jp/PDF-zip-7z.html>) というウェブサイトから `Sjisutf8.zip` をダウンロードしてください。
2. `Sjisutf8.zip` を展開してください。
3. Cドライブに `Sjisutf8` フォルダを移動してください。
4. Windows ロゴキー(田キー)を押しながら `Pause` を押してください。
5. メニューからシステムの詳細設定を選択してください。
6. 詳細設定タブをクリックしてください。
7. [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**
8. 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。
9. Path をクリックして選択してください。
10. [編集]ボタンを押してください。
11. Windows 10 の場合、[テキストの編集]ボタンを押してください。
12. 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。
13. 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:¥Sjisutf8` を入力してください。
14. 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、`Sjisutf8` フォルダごと削除してください。Path から `C:¥Sjisutf8` を削除してください。

設定ファイル

設定ファイルの名前は、`set8.txt` です。設定ファイルは、必須ではありません。

一時ファイル

`sjisutf8` コマンドは、一時ファイルとして、`binary.tmp`, `set8.txt`, `text.tmp` を作成する場合があります。`sjisutf8` コマンドは、自動的に一時ファイルを削除する場合があります。

放置

`sjisutf8` コマンドが自動的に一時ファイルを削除しないように命令するには、`sjisutf8` コマンドのパラメータとして、`-y` を付加してください。

ファイル	パラメータ	説明
binary.tmp	-b	新しいファイルとして、binary.tmp が作成されます。
	-t	読み込みたいファイルのコピーとして、binary.tmp が作成されます。
set8.txt		set8.txt を開くことができない場合、sjisutf8 コマンドは、自動的にカレントディレクトリに set8.txt を作成します。
text.tmp	利用者は、読み込みたいファイルを入力しなかった。	sjisutf8 コマンドは、標準入力から読み込んで、text.tmp を作成します。

著作権

Sogaya(そがや)は、sjisutf8 コマンドの著作権を保有しています。

免責条項

sjisutf8 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

32 ビット版の Windows XP Home において、Dev-C++ 4.9.9.2 という統合開発環境 (IDE) で sjisutf8 コマンドを構築しました。コンパイラは、GCC 3.4.2 です。ソースコードを添付しました。

ユーザーズガイド

2020 年 7 月 26 日 Revision 1.016

sjisutf8 コマンドのパラメータ

読み込みたいファイルおよび新しいファイルの名前を sjisutf8 コマンドのパラメータとして付加してください。sjisutf8 コマンドは、第 1 のファイルを読み込んで、第 2 のファイルを書き出します。

sjisutf8 Gongitsune.txt Gon_UTF-8.txt

COMP コマンド

どの Windows も最初から FC コマンドを用意しています。バイナリファイルとして比較す

るには、FC コマンドのパラメータとして、/B を付加してください。Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 10 では、COMP コマンドを試すことができます。

先に新しいファイルを命名したい

-o は、次に新しいファイルの名前が来る印です。すなわち、先に新しいファイルを命名するには、sjisutf8 コマンドのパラメータとして、-o new.txt を付加してください。-o および new.txt の間に空白を挿入してください。new.txt は、新しいファイルの名前です。

sjisutf8 -o new.txt Gongitsune.txt

```
C:\¥Q>sjisutf8 -o new.txt Gongitsune.txt  
C:\¥Q>sjisutf8 Gongitsune.txt Gon_UTF-8.txt  
C:\¥Q>COMP Gon_UTF-8.txt new.txt  
Gon_UTF-8.txt と new.txt を比較しています...  
ファイルに違いはありません  
  
(ほかのファイルを比較しますか (Y/N)? n  
C:\¥Q>
```

sjisutf8.txt

sjisutf8 コマンドのパラメータとして、1 個のファイルしか付加されなかった場合、sjisutf8 コマンドは、そのシフト JIS テキストファイルを読み込んで、新しい UTF-8 テキストファイルとして、sjisutf8.txt を書き出します。

```
C:\¥Q>sjisutf8 Gongitsune.txt  
C:\¥Q>sjisutf8 Gongitsune.txt Gon_UTF-8.txt  
C:\¥Q>COMP Gon_UTF-8.txt sjisutf8.txt  
Gon_UTF-8.txt と sjisutf8.txt を比較しています...  
ファイルに違いはありません  
  
(ほかのファイルを比較しますか (Y/N)? n  
C:\¥Q>
```

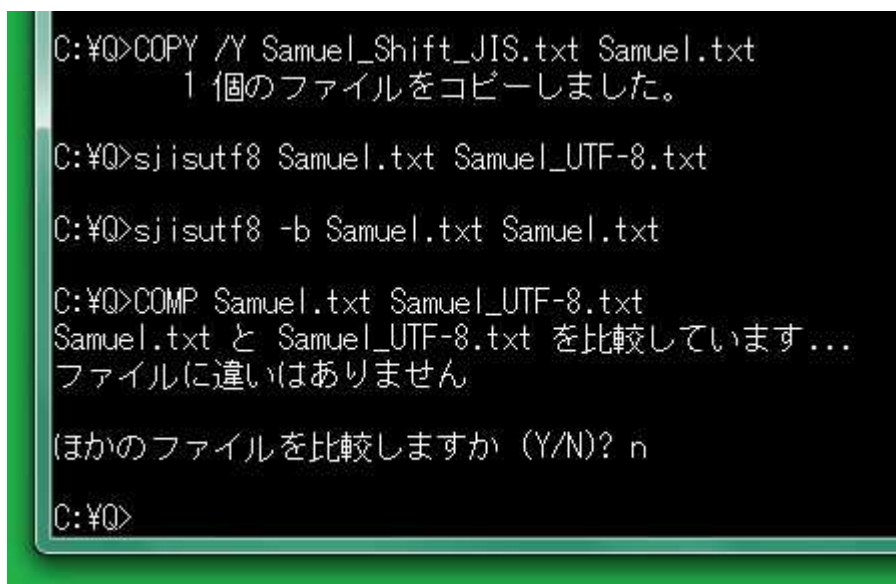
COPY コマンド

どの Windows も最初から COPY コマンドを用意しています。COPY コマンドは、既存のファイルが上書きされることを利用者に警告する場合があります。利用者がこの警告を確認しないで COPY コマンドを強行するには、COPY コマンドのパラメータとして /Y を付加してください。

2 個の同一名称

sjisutf8 コマンドは、第 1 のファイルを読み込んで、第 2 のファイルを書き出しますが、第 1 および第 2 のファイルが同一の名前であるように命令するには、sjisutf8 コマンドのパラメータとして、-b を付加してください。sjisutf8 コマンドは、いったん binary.tmp を作成して、binary.tmp を第 2 のファイルの名前に改名します。たとえば、シフト JIS の Samuel.txt を読み込んで、UTF-8 の Samuel.txt を書き出すには、コマンドプロンプトに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。

sjisutf8 -b Samuel.txt Samuel.txt



```
C:\¥Q>COPY /Y Samuel_Shift_JIS.txt Samuel.txt
1 個のファイルをコピーしました。

C:\¥Q>sjisutf8 Samuel.txt Samuel_UTF-8.txt

C:\¥Q>sjisutf8 -b Samuel.txt Samuel.txt

C:\¥Q>COMP Samuel.txt Samuel_UTF-8.txt
Samuel.txt と Samuel_UTF-8.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:\¥Q>
```

小さいファイルなら-b は不要だ

現在のバージョンの sjisutf8 コマンドは、まずシフト JIS テキストファイルの全体をメモリーにコピーして、ファイルを閉じます。次に、メモリーからファイルに UTF-8 のデータを書き出します。現在のバージョンの sjisutf8 コマンドは、シフト JIS テキストファイルの全体をメモリーにコピーできないおそれがあります。小さいファイルなら-b は不要ですが、大きいファイルなら-b を付加してください。-b 付きの sjisutf8 コマンドは、**1 行のみ**

読み込んで変換して1行を書き出す工程を反復します。

sjisutf8 Samuel.txt Samuel.txt

```
C:¥Q>COPY /Y Samuel_Shift_JIS.txt Samuel.txt
      1 個のファイルをコピーしました。

C:¥Q>sjisutf8 Samuel.txt Samuel_UTF-8.txt

C:¥Q>sjisutf8 Samuel.txt Samuel.txt

C:¥Q>COMP Samuel.txt Samuel_UTF-8.txt
Samuel.txt と Samuel_UTF-8.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:¥Q>
```

放置

-b 付きの sjisutf8 コマンドは、いったん binary.tmp を作成します。sjisutf8 コマンドが自動的に binary.tmp を削除しないように命令するには、sjisutf8 コマンドのパラメータとして、-y を付加してください。

```
C:¥Q>COPY /Y Samuel_Shift_JIS.txt Samuel.txt
      1 個のファイルをコピーしました。

C:¥Q>sjisutf8 -b -y Samuel.txt Samuel_UTF-8.txt

C:¥Q>COMP binary.tmp Samuel_UTF-8.txt
binary.tmp と Samuel_UTF-8.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:¥Q>
```

原稿どおりに改行できない

シフト JIS のテキストファイルが 0Ah のみで改行されていれば UTF-8 のテキストファイルを 0Ah のみで改行し、シフト JIS のテキストファイルが 0Dh および 0Ah で改行されていれば UTF-8 のテキストファイルを 0Dh および 0Ah で改行する機能は、-b 付きの sjisutf8 コマンドでは、無効です。初期段階で、どの行も 0Ah のみで改行するように設

定されます。パラメータとして`-0`を付加した場合、`sjisutf8` コマンドは、どの行も `0Dh` および `0Ah` で改行します。

改行

`0Dh` は、十六進数 (hexadecimal) の `0D` を意味します。テキストファイルに関して、Windows では、`0Dh` および `0Ah` で改行しますが、UNIX では、`0Ah` のみで改行します。`sjisutf8` コマンドは、各行から `0Dh` を削除できます。

設定を無効にする

`-0` は、`-L` より優先されるため、`-L` を指定したい場合、`-0` を無効にしなければなりません。初期段階で、`sjisutf8` コマンドは、`0Ah` のみで改行するように設定されています。この設定を無効にするには、`sjisutf8` コマンドのパラメータとして、`-L-`を付加してください。どの行も `0Dh` および `0Ah` で改行するには、`-0` を付加してください。

パラメータ	改行	説明
<code>-0</code>	<code>0Dh</code> および <code>0Ah</code> で改行する。	<code>-0</code> は、 <code>-L</code> より優先される。
<code>-0-</code>	シフト JIS のファイルにしたがって改行する	シフト JIS ファイルの行が、 <code>0D 0A</code> で改行されていれば、 <code>sjisutf8</code> コマンドは、 <code>0Dh</code> および <code>0Ah</code> で改行する。
<code>-0- -L</code>	<code>0Ah</code> のみで改行する。	省略された場合、 <code>sjisutf8</code> コマンドは、パラメータとして <code>-0- -L</code> が付加されたとみなす。

0A のみで改行

元のシフト JIS ファイルに `0Dh` があろうがなかろうが、`0Ah` のみで改行して UTF-8 のファイルを書き出すには、`sjisutf8` コマンドのパラメータとして、`-LF` を付加してください。`-LF` は、`-0- -L` (マイナスゼロマイナス空白マイナスエル) と同じ結果を生じます。

ファイルの日時

元のファイルが新しいファイルと同じ日時を有するように命令するには、`sjisutf8` コマンドのパラメータとして、`-t` を付加してください。

一時ファイルを削除しない

`-t` 付きの `sjisutf8` コマンドは、一時ファイルとして `binary.tmp` を作成します。`sjisutf8` コマンドが自動的に一時ファイルを削除しないように命令するには、`sjisutf8` コマンドのパラメータとして、`-t -y` を付加してください。


```
C:\¥Q>DIR /O:D "Gon*.txt" | FINDSTR "Gon"
2018/10/11 13:53          10,636 Gongitsune.txt

C:\¥Q>sjisutf8 -t -y Gongitsune.txt Gon_UTF-8.txt

C:\¥Q>COMP binary.tmp Gongitsune.txt
binary.tmp と Gongitsune.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:\¥Q>DIR /O:D "Gon*.txt" | FINDSTR "Gon"
2020/06/15 21:33          10,636 Gongitsune.txt
2020/06/15 21:33          15,746 Gon_UTF-8.txt

C:\¥Q>
```

日時が新しい

1. 利用者は、カレントディレクトリにシフト JIS のファイルを用意する。
2. sjisutf8 コマンドは、シフト JIS のファイルを読み込む。
3. -t 付きの sjisutf8 コマンドは、一時ファイルとして binary.tmp を作成する。
4. binary.tmp は、内容もサイズもシフト JIS ファイルと同じである。
5. -t 付きの sjisutf8 コマンドは、binary.tmp を読み込んで新しいシフト JIS ファイルを作成する。
6. -y が付加されていなければ、sjisutf8 コマンドは、binary.tmp を削除する。
7. sjisutf8 コマンドは、新しいシフト JIS ファイルを読み込んで、UTF-8 のファイルを書き出す。
8. 新しいシフト JIS ファイルは、日時が UTF-8 のファイルと同じである。

sjisutf8 コマンドの変換表

MS932~UTF-8.bin にしたがって、sjisutf8 コマンドは、シフト JIS のテキストファイルを UTF-8 のファイルに変換します。

どの変換表を使用するか

C:\¥Ssjisutf8

に Sjisutf8.exe および MS932~UTF-8.bin をインストールしました。通常は、sjisutf8 コマンドをインストールしたディレクトリにある MS932~UTF-8.bin にアクセスできます。どの変換表を使用するか表示するには、sjisutf8 コマンドのパラメータとして、

-w を付加してください。このデータは、data.txt に保存されています。

```
C:¥Q>sjisutf8 -w Gongitsune.txt
C:¥Sjisutf8¥MS932~UTF-8.bin is primarily accessible to Sjisutf8.

C:¥Q>TYPE data.txt
C:¥Sjisutf8¥MS932~UTF-8.bin is primarily accessible to Sjisutf8.

C:¥Q>
```

カレントディレクトリ

現在のディレクトリをカレントディレクトリと言います。sjisutf8 コマンドをインストールしたディレクトリから作業ディレクトリ(C:¥Q)に、MS932~UTF-8.bin を移動できます。もはやインストールディレクトリから MS932~UTF-8.bin を発見できません。CD コマンドで、作業ディレクトリをカレントディレクトリにしてください。

削除されるおそれ

カレントディレクトリでは、何らかの誤操作のため、MS932~UTF-8.bin が削除されるおそれがあります。バックアップを目的として、事前に適当なディレクトリに MS932~UTF-8.bin のコピーを作成することを推奨します。カレントディレクトリの変換表を使用する場合、sjisutf8 コマンドは、変換表のパス名ではなくファイル名を表示することを下図に示します。

```
C:¥Q>CD ¥Sjisutf8

C:¥Sjisutf8>MOVE MS932~UTF-8.bin C:¥Q
      1 個のファイルを移動しました。

C:¥Sjisutf8>CD ¥Q

C:¥Q>sjisutf8 -w Gongitsune.txt
MS932~UTF-8.bin is primarily accessible to Sjisutf8.

C:¥Q>
```

環境変数 Path

インストールディレクトリにもカレントディレクトリにも変換表を発見できなかった場合、sjisutf8 コマンドは、環境変数 Path に記載の各ディレクトリを探索します。発見されるまで多数のエラーファイルを作成します。err 通し番号-分-秒.txt 形式で各エラーファイルが命名されます。

通し番号をリセット

エラーファイルを作成する前に、sjisutf8 コマンドは、serial.txt を読み込んで、数値に 1 を加算して、serial.txt に加算の結果を書き出します。serial.txt を削除するか、serial.txt に 0 を記入することのいずれかによって、通し番号をリセットできます。

設定ファイル

sjisutf8 コマンドは、インストールディレクトリにある設定ファイルを読み込もうとします。設定ファイルのパス名は、C:¥Sjisutf8¥set8.txt です。インストールディレクトリで設定ファイルを開けない場合、カレントディレクトリにある設定ファイルを読み込もうとします。



一時ファイル

カレントディレクトリでも設定ファイルを開けない場合、sjisutf8 コマンドは、自動的に一時ファイルとして set8.txt を作成します。これは、シフト JIS のテキストファイルです。sjisutf8 コマンドが自動的に set8.txt を削除しないように命令するには、sjisutf8 コマンドのパラメータとして -y を付加してください。

第 1 行に変換表のパス名

設定ファイルの第 1 行に変換表のパス名を記入してください。第 2 行には、たとえば、-0- を記入してください。-0- は、0Dh および 0Ah で改行する機能を無効にします。第 3 行には、-L を記入してください。どの行も 0Ah で改行することを意味します。第 4 行には、必要に応じて、-w を記入してください。以下同様です。

注釈

コマンドプロンプトには、パラメータを空白で区切って並べますが、設定ファイルには、各行に 1 個のパラメータを記入してください。各行は、パラメータ、タブ、注釈の順に記入してください。

変換表を探索する順序

sjisutf8 コマンドは、下記の順序で変換表を探索します。

1. sjisutf8 コマンドがインストールされたディレクトリ
2. カレントディレクトリ
3. 環境変数 Path に記載の各ディレクトリ
4. 設定ファイルの第 1 行に記載のパス名

設定ファイルに記載のパス名を優先

インストールディレクトリ、カレントディレクトリ、Path に記載の各ディレクトリに変換表があろうがなかろうが、設定ファイルの第 1 行からパス名を捕捉して、変換表を開くには、sjisutf8 コマンドのパラメータとして、-a を付加してください。

```
C:\>TYPE C:\Sjisutf8\set8.txt
C:\>ColonYen\MS932~UTF-8.bin
-O-      UNIX
-L       LF

C:\>sjisutf8 -a -w Gongitsune.txt date.txt
C:\>ColonYen\MS932~UTF-8.bin is primarily accessible to Sjisutf8.

C:\>DIR "dat*.txt" | FINDSTR "dat"
2020/06/16  06:29                66 data.txt
2020/06/16  06:29            15,746 date.txt

C:\>
```

変換表の探索を抑制

sjisutf8 コマンドが変換表を探索することを抑制するには、sjisutf8 コマンドのパラメータとして、-d を付加してください。

制御文字

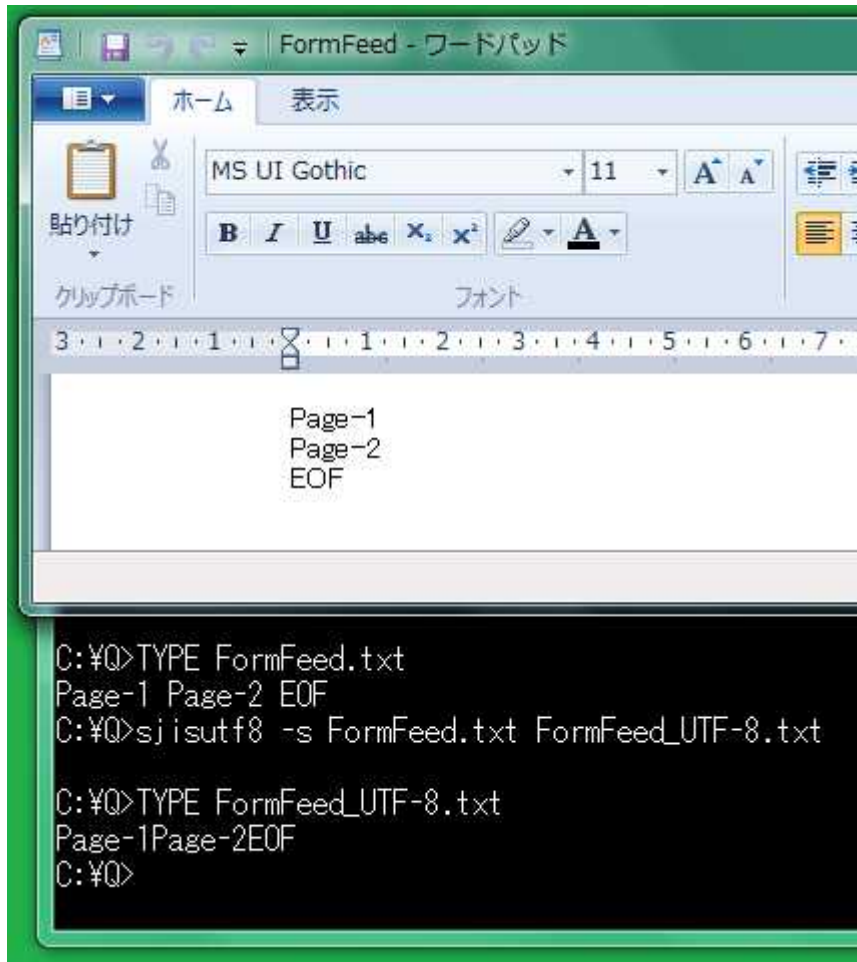
制御文字としては、MS-DOS テキストファイルの終わり(1Ah)、エスケープシーケンスの切替え(1Bh)、ベル(07h)、改ページ、タブ、改行、復帰などがあります。

制御文字を削除

09h, 0Ch, 1Bh などの制御文字を削除するには、sjisutf8 コマンドのパラメータとして、-s を付加してください。-s 付きの sjisutf8 コマンドで 0Ch が削除された例を下図に示します。

ワードパッドは改ページを認識する

Windows のアクセサリには、メモ帳だけではなくワードパッドが含まれています。ワードパッドで FormFeed.txt を開くことができます。コマンドプロンプトと異なり、ワードパッドは、改ページを認識します。3 枚の紙がプリンターから出てきます。



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	end of file	EOF	MS-DOS テキストファイルの終わり
		substitute	SUB	置換
1Bh	27	escape	ESC	エスケープシーケンスの切替え

iconv コマンド

文字コードを変換するために、iconv コマンドが GNU Win32 に用意されています。インターネットで、たとえば、libiconv-1.9.2-1 などのインストーラーを検索してください。接頭辞の lib は、ライブラリを意味します。このインストーラーをダウンロードして、実行して、32 ビット版 Windows 7 のパソコンに iconv.exe をインストールしました。

```
C:\¥Q>iconv --version
iconv (GNU libiconv 1.9)
Copyright (C) 2000-2002 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Bruno Haible.

C:\¥Q>iconv -l | FINDSTR /I "CP932 EUC-JP Shift_JIS"
EUC-JP EUCJP_EXTENDED_UNIX_CODE_PACKED_FORMAT_FOR_JAPANESE CSEUCPKDFMTJAPANESE
MS_KANJI SHIFT-JIS SHIFT_JIS SJIS CSSHIFTJIS
CP932
SHIFT_JISX0213
C:\¥Q>
```

CHCP コマンド

規格として、シフト JIS ではなく 932 のコードページ (MS932) が Windows のコマンドプロンプトに使用されます。MS932 は、いわゆるシフト JIS ですが規格外です。MS932 は、CP932 とも言います。932 を表示するには、コマンドプロンプトに、CHCP コマンドを入力して、パラメータを入力しないで、Enter を押してください。



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

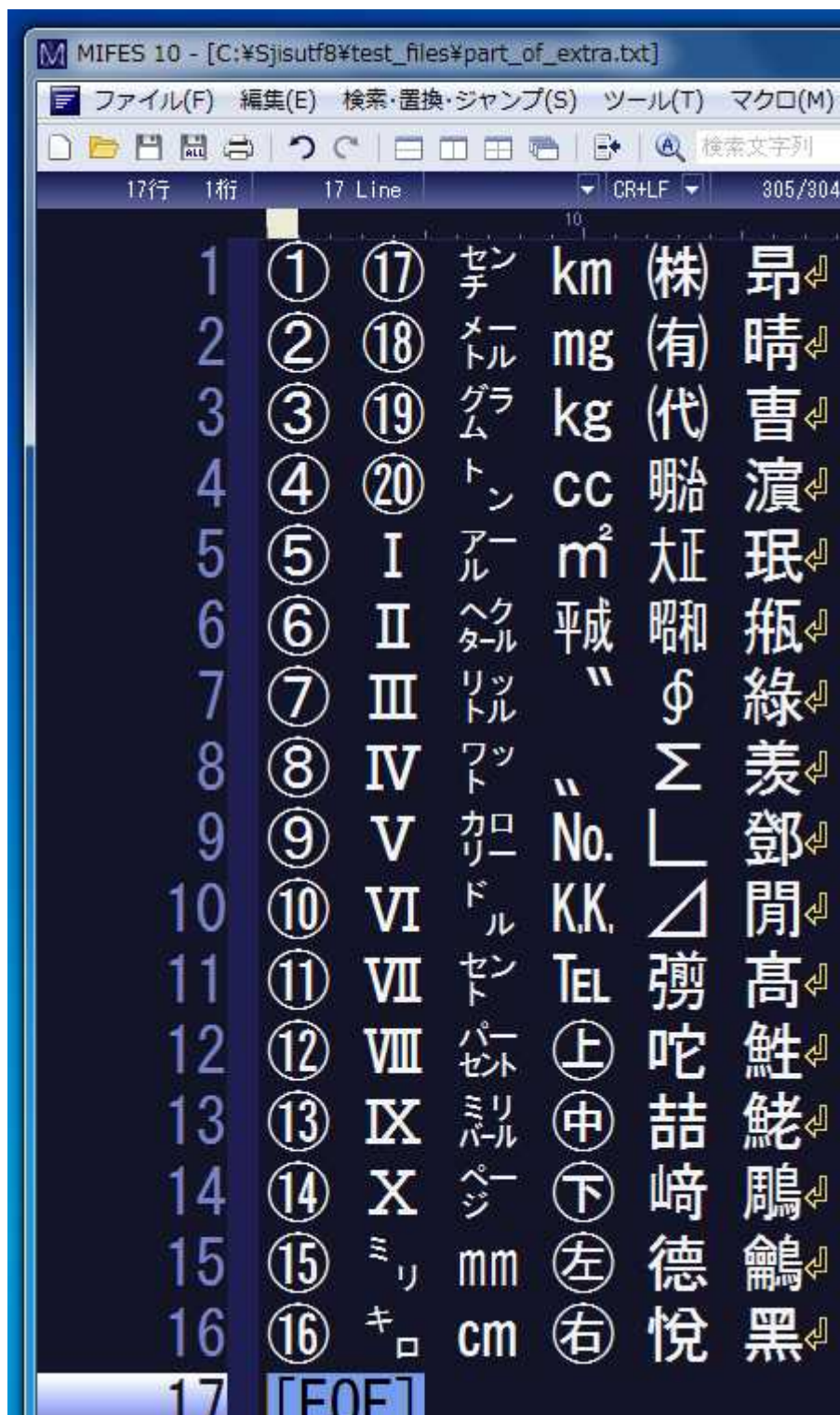
Windows-31J

MS932 は、Windows-31J とも言います。日本語のコードページは、932 ですが、英語 (アメリカ) のコードページは、437 または 1252 です。日本語版 Windows の文字コードは、Windows-31J ですが、英語版 Windows の文字コードは、Windows-1252 です。

機種依存文字

たとえば、丸付きの数字 (①②③⑩②⑩)、ローマ数字 (ⅠⅡⅢⅣⅩ) などは、シフト JIS の規格 (JIS X 0208) に含まれていません。CP932 から UTF-8 へ変換するように設定するには、iconv コマンドのパラメータとして、Shift_JIS ではなく CP932 を付加してください。Windows において、MIFES (マイフェス) というテキストエディタで正しく表示できた機種依存文字の例を下図に示します。フォントは、**MS ゴシック** です。これらは、添

付の extra_CP932.txt に含まれています。



テキストファイル

テキストファイルは、テキストエディタで編集しやすいファイルです。EUC-JP, シフト

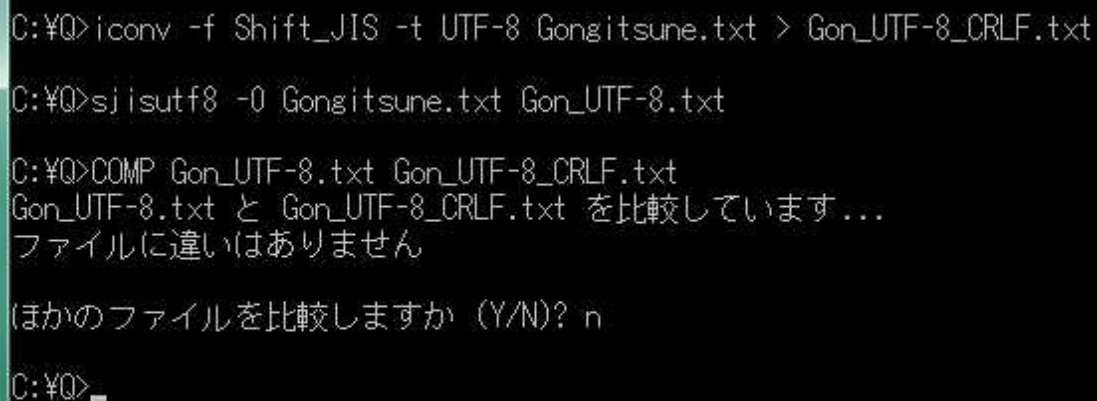
JIS, UTF-8 などのテキストファイルは、00h が含まれていませんが、UTF-16 のファイルは、00h が含まれている場合があります。

テキストエディタ

Windows 用のテキストエディタとしては、EmEditor, K2, Mery, MIFES, TeraPad, WZ, サクラエディタ、秀丸、メモ帳などがあります。インターネットからダウンロードした HTML のテキストファイルを開覧するには、0Ah のみで改行されたテキストも正しく表示できるテキストエディタを選択してください。

リダイレクトでファイルを作成

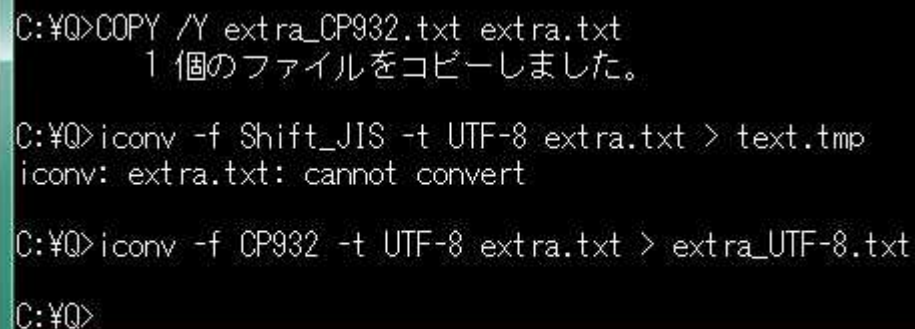
シフト JIS から UTF-8 へ変換するように設定して、iconv コマンドを実行しました。リダイレクトして UTF-8 のテキストファイルを作成しました。このファイルは、どの行も 0Dh および 0Ah で改行されています。



```
C:¥Q>iconv -f Shift_JIS -t UTF-8 Gongitsune.txt > Gon_UTF-8_CRLF.txt
C:¥Q>sjisutf8 -0 Gongitsune.txt Gon_UTF-8.txt
C:¥Q>COMP Gon_UTF-8.txt Gon_UTF-8_CRLF.txt
Gon_UTF-8.txt と Gon_UTF-8_CRLF.txt を比較しています...
ファイルに違いはありません
(ほかのファイルを比較しますか (Y/N)? n
C:¥Q>
```

Shift_JIS でエラーが発生

Shift_JIS から UTF-8 に変換する設定で、エラーが発生して変換が中止された例を下図に示します。テキストファイルに機種依存文字が含まれているおそれがある場合、シフト JIS から UTF-8 へ変換するには、iconv コマンドのパラメータとして、Shift_JIS ではなく CP932 を入力してください。



```
C:¥Q>COPY /Y extra_CP932.txt extra.txt
1 個のファイルをコピーしました。
C:¥Q>iconv -f Shift_JIS -t UTF-8 extra.txt > text.tmp
iconv: extra.txt: cannot convert
C:¥Q>iconv -f CP932 -t UTF-8 extra.txt > extra_UTF-8.txt
C:¥Q>
```

中止したい合図

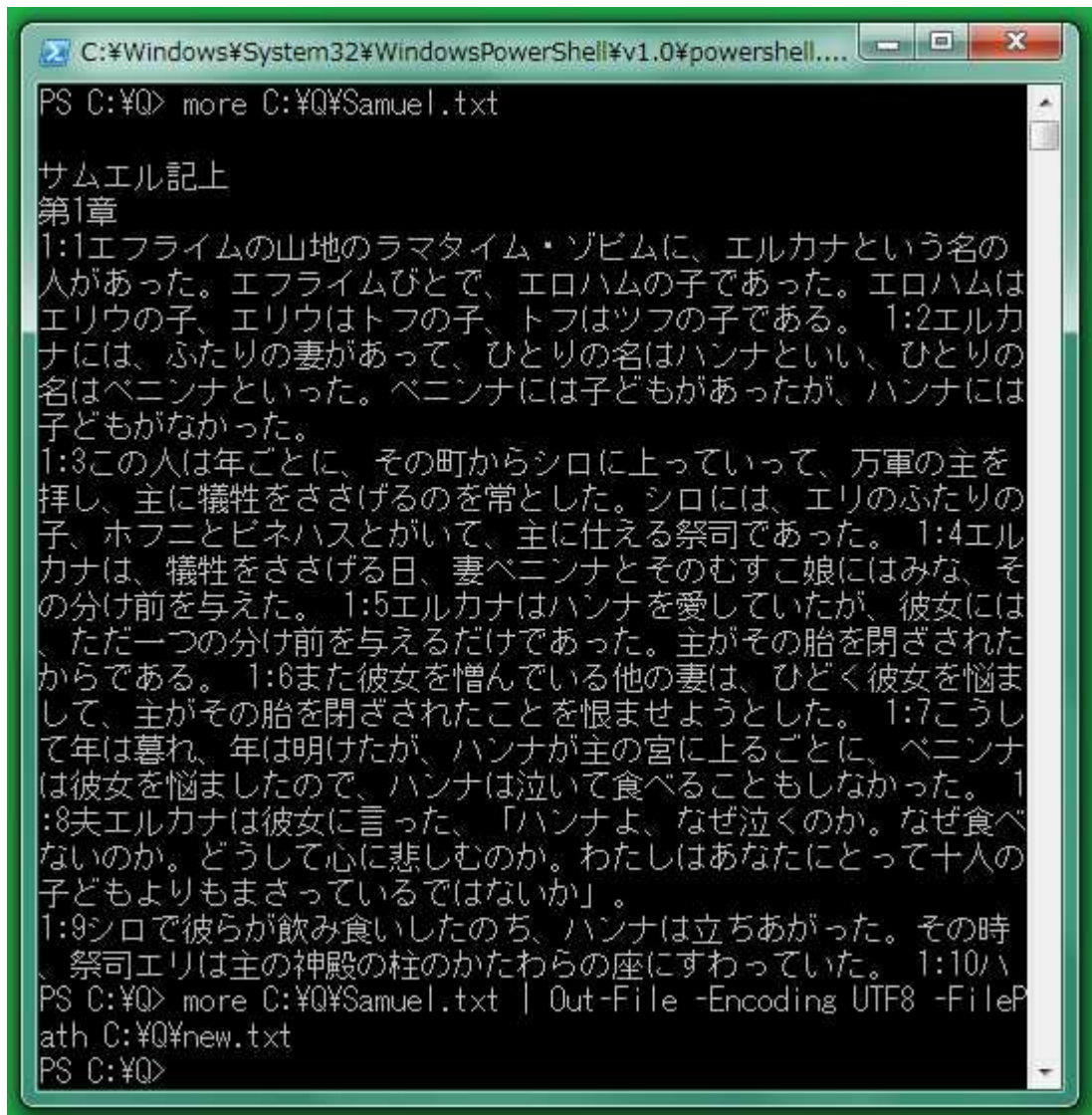
コマンドプロンプトに、プログラムの実行を中止したい合図を入力するには、Ctrl を押しながらかを押してください。Ctrl + C は、この操作を意味します。

PowerShell の Out-File

PowerShell ウィンドウを開くには、まず Windows ロゴキー(田キー)を押しながら、R を押してください。[ファイル名を指定して実行]ウィンドウが出現します。名前欄に powershell を入力して Enter を押してください。さもないと、OK ボタンを押してください。

文字コードを指定してリダイレクト

Samuel.txt は、シフト JIS のテキストファイルです。more コマンドおよび Out-File コマンドレットのパイプによって、新しい UTF8 ファイルを作成した例を下図に示します。



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
PS C:\> more C:\>Samuel.txt

サムエル記上
第1章
1:1エフライムの山地のラマタイム・ゾピムに、エルカナという名の
人があった。エフライムひとりで、エロハムの子であった。エロハムは
エリウの子、エリウはトフの子、トフはツフの子である。 1:2エルカ
ナには、ふたりの妻があつて、ひとりの名はハンナといい、ひとりの
名はペニンナといった。ペニンナには子どもがあつたが、ハンナには
子どもがなかった。
1:3この人は年ごとに、その町からシロに上って行って、万軍の主を
拝し、主に犠牲をささげるのを常とした。シロには、エリのふたりの
子、ホフニとピネハスとがいて、主に仕える祭司であつた。 1:4エル
カナは、犠牲をささげる日、妻ペニンナとそのむすこ娘にはみな、そ
の分け前を与えた。 1:5エルカナはハンナを愛していたが、彼女には
、ただ一つの分け前を与えるだけであつた。主がその胎を閉ざされた
からである。 1:6また彼女を憎んでいる他の妻は、ひどく彼女を悩ま
して、主がその胎を閉ざされたことを恨ませようとした。 1:7こうし
て年は暮れ、年は明けたが、ハンナが主の宮に上るごとに、ペニンナ
は彼女を悩ましたので、ハンナは泣いて食べることもしなかった。 1
:8夫エルカナは彼女に言った、「ハンナよ、なぜ泣くのか。なぜ食べ
ないのか。どうして心に悲しむのか。わたしはあなたにとって十人の子
どもよりもまさっているではないか」。
1:9シロで彼らが飲み食いしたのち、ハンナは立ちあがった。その時
、祭司エリは主の神殿の柱のかたわらの座にすわっていた。 1:10ハ
PS C:\> more C:\>Samuel.txt | Out-File -Encoding UTF8 -FileP
ath C:\>new.txt
PS C:\>
```

コマンド	パラメータ	説明
more	C:¥Q¥Samuel_UTF-8.txt	PowerShell ウィンドウにシフト JIS のテキストファイルを表示できる。more コマンドを終了するには、Q を押してください。
Out-File	-Encoding UTF8	UTF-16 ではなく UTF-8 を指定できる。
	-FilePath C:¥Q¥new.txt	new.txt にリダイレクトする。

標準入力

利用者が読み込みたいファイルを入力しなかった場合、sjisutf8 コマンドは、標準入力から読み込もうとします。

```
C:¥Q>DEL sjisutf8.txt

C:¥Q>TYPE short.txt
短い

C:¥Q>TYPE short.txt | sjisutf8

C:¥Q>DIR sjisutf8.txt | FINDSTR "sjisutf8.txt"
2020/06/17 20:16          8 sjisutf8.txt

C:¥Q>
```

パラメータ無しの sjisutf8 コマンド

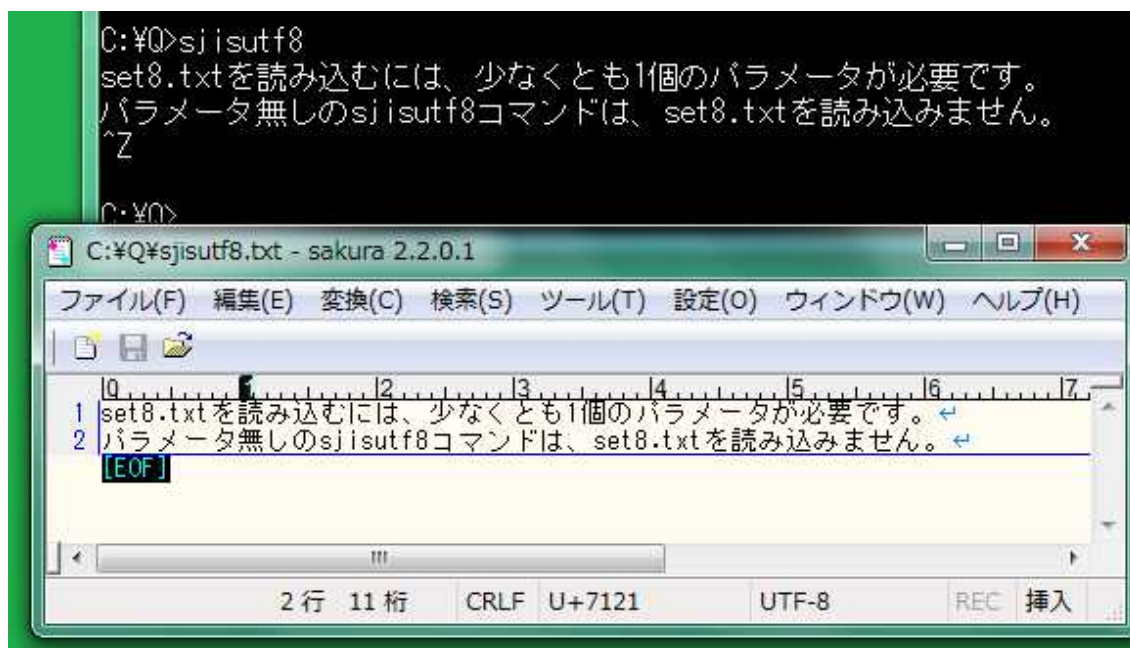
パラメータが 1 個も入力されなかった場合、sjisutf8 コマンドは、下記の設定で sjisutf8.txt を書き出します。設定ファイルは、読み込まれません。

設定	説明
C:¥Ssjisutf8¥MS932~UTF-8.bin	set8.txt の第 1 行に記入されたとみなす。
-d-	変換表を探索する。
-a-	set8.txt の第 1 行に記入されたパス名を優先しない。
-w-	どの変換表を使用するか表示しない。

ファイルの終わり

キーボードから入力したのち、このモードから脱出するには、Ctrl を押しながら Z を押してください。Ctrl + Z は、この操作を意味します。^Z が表示される場合があります。

す。



無効にしたい

たとえば、`-L-`は、`-L`を無効にします。初期段階で`-L`が設定されますが、利用者は、`set8.txt` に`-L-`を記入できます。

何回でも指定できる

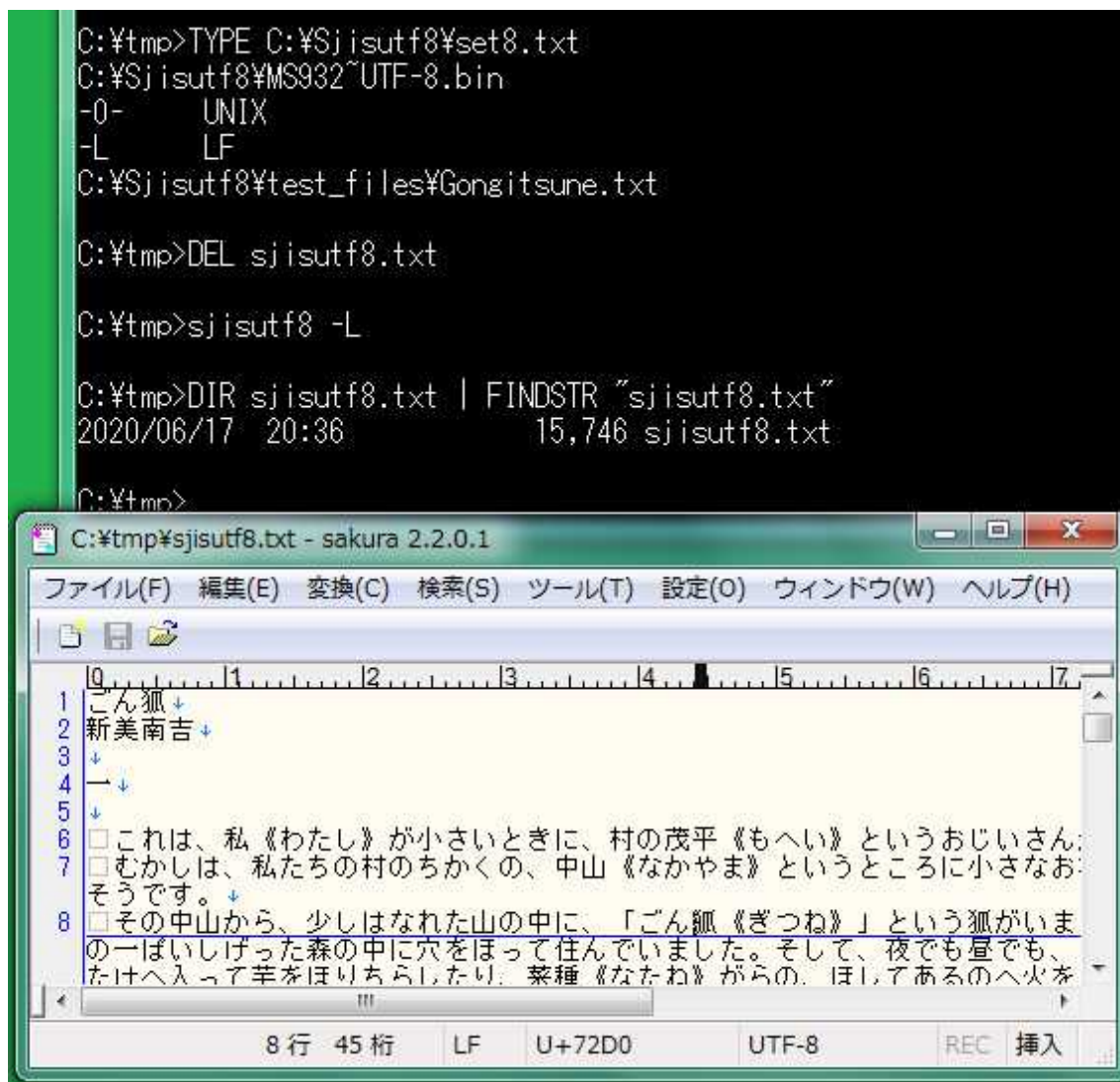
`set8.txt` に`-L-`が記入された場合でも、`sjisutf8` コマンドのパラメータとして、`-L`を付加できます。コマンドプロンプトに入力されたパラメータが設定ファイルより優先されます。ただし、`-L`があろうがなかろうが、`-0`が優先されるため、`-L`を有効にするには、`-0-`が必要です。

少なくとも1個のパラメータを付加してください

`set8.txt`を読み込むには、少なくとも1個のパラメータが必要です。たとえば、すでに`set8.txt` に`-L`が記入されている場合、`sjisutf8` コマンドのパラメータとして再度`-L`を入力しても悪い影響は生じません。

第1のファイルのパス名

`sjisutf8` コマンドは、第1のファイルを読み込んで、変換して、第2のファイルを書き出します。`sjisutf8`のパラメータとして第1のファイルの名前を入力すると同様に、利用者は、読み込みたいファイルの名前を`set8.txt`に記入できます。パス名を記入した例を下图に示します。

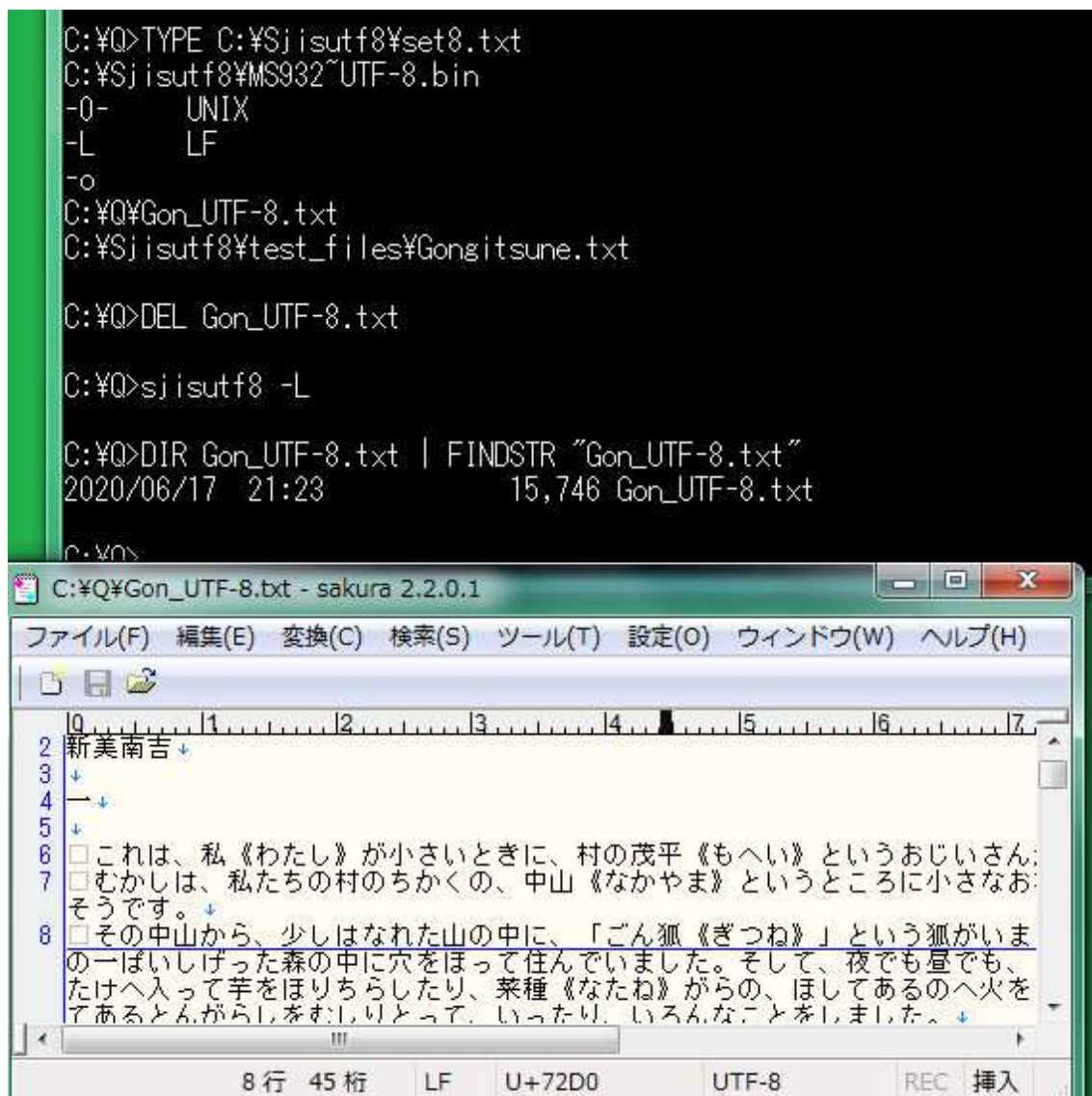


第2のファイルのパス名

sjisutf8 コマンドは、第1のファイルを読み込んで、変換して、第2のファイルを書き出します。sjisutf8 のパラメータとして第2のファイルの名前を入力するのと同様に、利用者は、書き出したいファイルの名前を set8.txt に記入できます。

順序を変更できる

第1のファイルの名前より下に、第2のファイルの名前を記入してください。先に第2のファイルの名前を記入するには、ある行に-○を記入して、その次の行に第2のファイルの名前を記入してください。パス名を記入した例を下図に示します。



第 1 のファイルが指定されなかった

上図の例のとおり、書き出したいファイルを -o で先に指定できますが、利用者が読み込みたいファイルが set8.txt にも sjisutf8 コマンドのパラメータにも結局は指定されなかった場合、sjisutf8 コマンドは、標準入力から読み込もうとします。


DIR コマンド

どの Windows も最初から DIR コマンドを用意しています。ファイルのサイズに関して、昇順に行を並べ替えるには、DIR コマンドのパラメータとして、/O:S を付加してください。日本語は、CP932 で表示されます。DIR コマンドおよび sjisutf8 コマンドのパイプにより、UTF-8 のテキストファイルを作成できます。下記の例では、set8.txt にも sjisutf8 コマンドのパラメータにも結局は第 1 のファイルの名前が指定されなかったため、sjisutf8 コマンドは、標準入力から読み込みました。


```
C:\>DIR /O:S *.csv | sjisutf8 -o DIR_O_S.txt

C:\>TYPE C:\>sjisutf8\set8.txt
C:\>sjisutf8\MS932~UTF-8.bin
-O-      UNIX
-L       LF

C:\>DIR /O:S *.csv | sjisutf8 -o DIR_O_S.txt
```



計測

PowerShell でコマンドの実行時間を計測できます。

PowerShell ウィンドウを開く

PowerShell ウィンドウを開くには、まず Windows ロゴキー（田キー）を押しながら、R を押してください。[ファイル名を指定して実行]ウィンドウが出現します。名前欄に powershell を入力して Enter を押してください。さもないと、OK ボタンを押してください。

設定ファイルを読み込むコマンドは計測しやすい

sjisutf8 コマンドは、設定ファイルにパラメータを記入できます。iconv コマンドと異なり、

sjisutf8 コマンドは、リダイレクトしないで、UTF-8 のテキストファイルを作成できます。Out-File コマンドレットと異なり、sjisutf8 コマンドは、パイプを構成しないで、シフト JIS のテキストファイルを読み込むことができます。

```
C:¥Q>TYPE C:¥Sjisutf8¥set8.txt
C:¥Sjisutf8¥MS932~UTF-8.bin
-O-      UNIX
-L      LF
C:¥Q>
```

-B 付きの sjisutf8 コマンド

-b 付きの sjisutf8 コマンドは、**1 行のみ読み込んで変換して 1 行を書き出す工程**を反復しますが、これを発展させたモードとして、-B 付きの sjisutf8 コマンドは、各行のバッファが満杯かどうか点検します。添付の `too_long.txt` は、第 4 行、第 5 行、第 6 行が 8191 バイトを超える長さを有しています。

8191 バイトしか読み込めない

`too_long.txt` の第 4 行に関して、-B 付きの sjisutf8 コマンドは、まず 8191 バイトぶんを読み込んで変換して書き出して、第 5 行として残りを読み込んで変換して書き出します。すなわち、新しいファイルの第 5 行は、`too_long.txt` の第 4 行の一部です。バッファが満杯になるごとに行番号がずれていきます。このとき、-B 付きの sjisutf8 コマンドは、標準エラー出力に警告および新しい行番号を表示します

ファイルの行をカウント

ファイルの行数を表示するには、sjisutf8 コマンドのパラメータとして -l を付加してください。l は、L の小文字です。【メモ】-l 付きの sjisutf8 コマンドは、1 行ずつ読み込んだ回数を表示するのではなく、0Ah の個数を表示します。

```
C:¥Q>sjisutf8 -l too_long.txt
6 lines

C:¥Q>sjisutf8 -B too_long.txt text.tmp
Warning: full buffer ( Line-4 )
Warning: full buffer ( Line-6 )
Warning: full buffer ( Line-8 )

C:¥Q>sjisutf8 -l text.tmp
9 lines

C:¥Q>
```

何も警告されない

-B 付きの sjisutf8 コマンドと異なり、-b 付きの sjisutf8 コマンドは、各行のバッファが満杯かどうか点検しません。行数が増加したことから、利用者は、各行のバッファが満杯になったか気づくことができます。

```
C:¥Q>sjisutf8 -l too_long.txt
6 lines

C:¥Q>sjisutf8 -b too_long.txt text.tmp

C:¥Q>sjisutf8 -l text.tmp
9 lines

C:¥Q>_
```

小さいファイルなら-b は不要だ

-B も-b も付加されていない sjisutf8 コマンドは、ファイルの全体をメモリーにコピーして、ファイルを閉じて、変換して、メモリーからファイルにデータを書き出します。行数は、変化しません。-B も-b も付加されていない sjisutf8 コマンドに関しては、利用者は、行の長さではなくファイルのサイズに注意してください。

```
C:¥Q>sjisutf8 -l too_long.txt
6 lines

C:¥Q>sjisutf8 too_long.txt text.tmp

C:¥Q>sjisutf8 -l text.tmp
6 lines

C:¥Q>_
```

-B の有無で比較

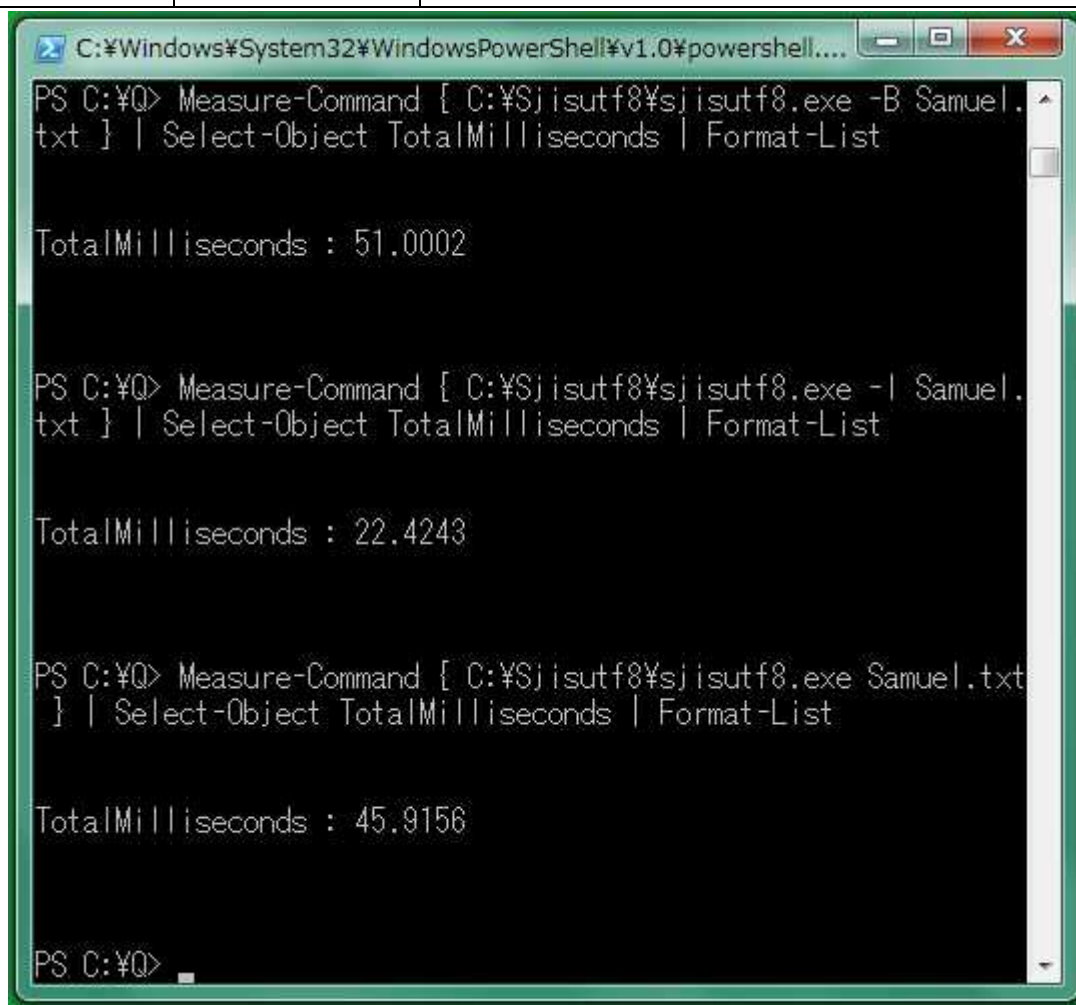
-B 付きの sjisutf8 コマンドは、**1 行のみ読み込んで変換して 1 行を書き出す工程**を反復しますし、各行のバッファが満杯かどうか点検します。-B 無しの sjisutf8 コマンドは、ファイルの全体をメモリーにコピーして、ファイルを閉じて、変換して、メモリーからファイルにデータを書き出します。-l 付きの sjisutf8 コマンドは、バイナリファイルとして各レコードを読み込んで 0Ah をカウントして個数を表示します。

Measure-Command コマンドレット

PowerShell において、Measure-Command コマンドレット、Select-Object コマンドレット、

Format-List コマンドレットのパイプにより、ミリ秒単位で、sjisutf8 コマンドの実行時間を表示してみました。

パラメータ	実行時間の例	説明
-B 付き	51 ミリ秒	1 行のみ読み込んで行の長さを点検しながら書き出す工程を反復した。
-l 付き	22 ミリ秒	ファイルの行数をカウントした。
-B 無し	46 ミリ秒	ファイル全体を読み込んで、ファイルを閉じて、メモリーからファイルに書き出した。



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell....
PS C:\> Measure-Command { C:\Sjisutf8\Sjisutf8.exe -B Samuel.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 51.0002

PS C:\> Measure-Command { C:\Sjisutf8\Sjisutf8.exe -l Samuel.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 22.4243

PS C:\> Measure-Command { C:\Sjisutf8\Sjisutf8.exe Samuel.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 45.9156

PS C:\>
```

Out-File コマンドレットと比較

-Encoding UTF8 が付加された Out-File コマンドレットは、標準入力から読み込んで、変換して、BOM 付き UTF-8 テキストファイルを作成できます。どの行も 0Dh および 0Ah で改行されています。

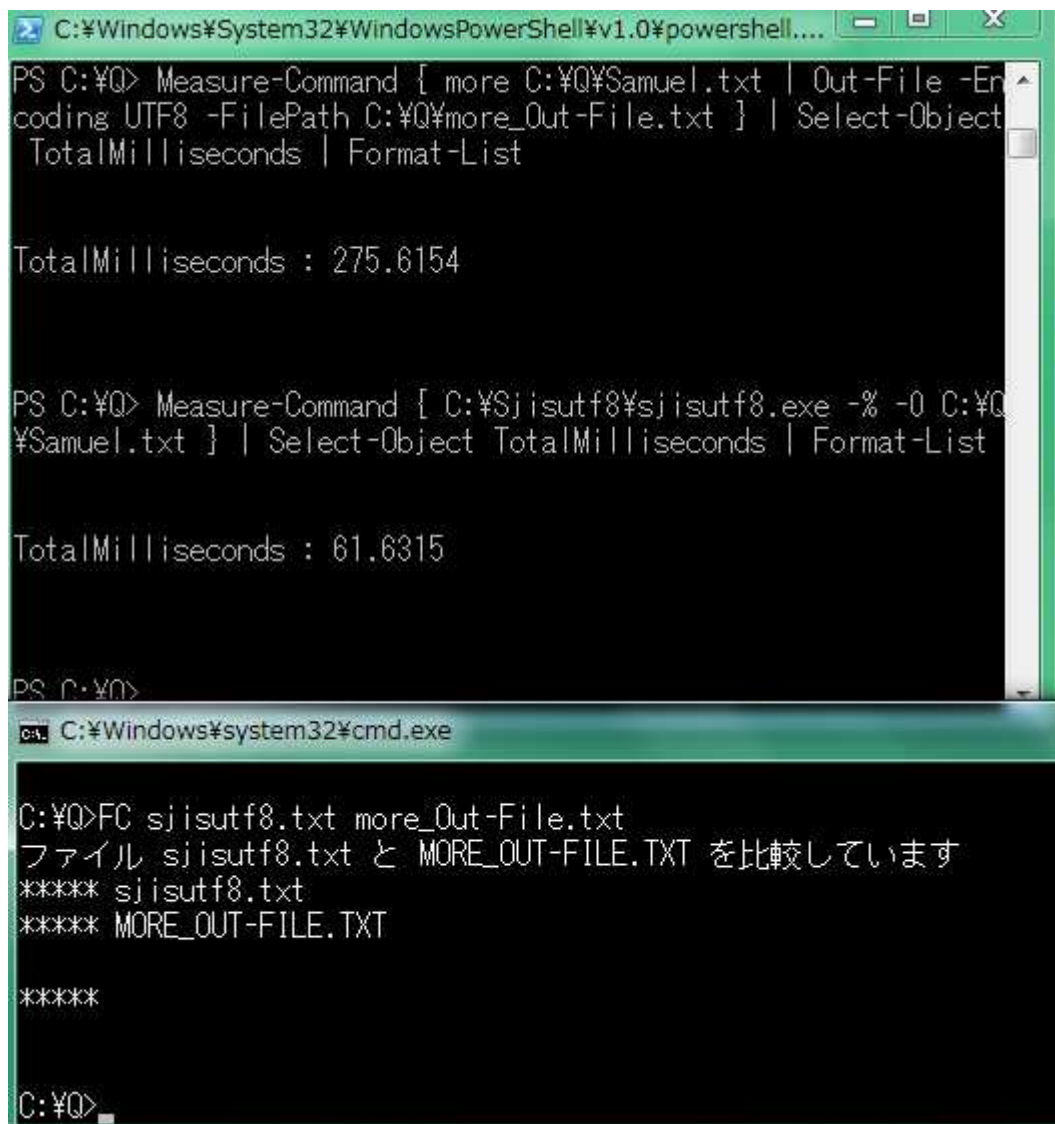
BOM 付きの UTF-8 テキストファイル

EFh, BBh, BFh が UTF-8 テキストファイルの最初の部分にあるかもしれません。EFh,

BBh, BFh は、UTF-8 テキストファイルの BOM です。BOM は、不要ですが、BOM 付きの UTF-8 テキストファイルを作成するには、sjisutf8 コマンドのパラメータとして-%(マイナスパーセント)を付加してください。

0Dh および 0Ah で改行

0Dh および 0Ah で改行するには、sjisutf8 コマンドのパラメータとして-0(マイナスゼロ)を付加してください。



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell...
PS C:\> Measure-Command { more C:\>Samuel.txt | Out-File -Encoding UTF8 -FilePath C:\>more_Out-File.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 275.6154

PS C:\> Measure-Command { C:\>sjisutf8\sjisutf8.exe -% -0 C:\>Samuel.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 61.6315

PS C:\>

C:\Windows\system32\cmd.exe
C:\>FC sjisutf8.txt more_Out-File.txt
ファイル sjisutf8.txt と MORE_OUT-FILE.TXT を比較しています
***** sjisutf8.txt
***** MORE_OUT-FILE.TXT

*****

C:\>
```

Out-File コマンドレットはファイルの終わりで改行した

-% -0 付きの sjisutf8 コマンドは、上図に示したとおり、Out-File コマンドレットとほぼ同じ UTF-8 テキストファイルを作成できました。相違点として、ファイルの終わりで Out-File コマンドレットは、余分に改行しました。

FC コマンドではなく COMP コマンドで比較してみる

ファイルの終わりで Out-File コマンドレットが追加した改行をテキストエディタで削除し

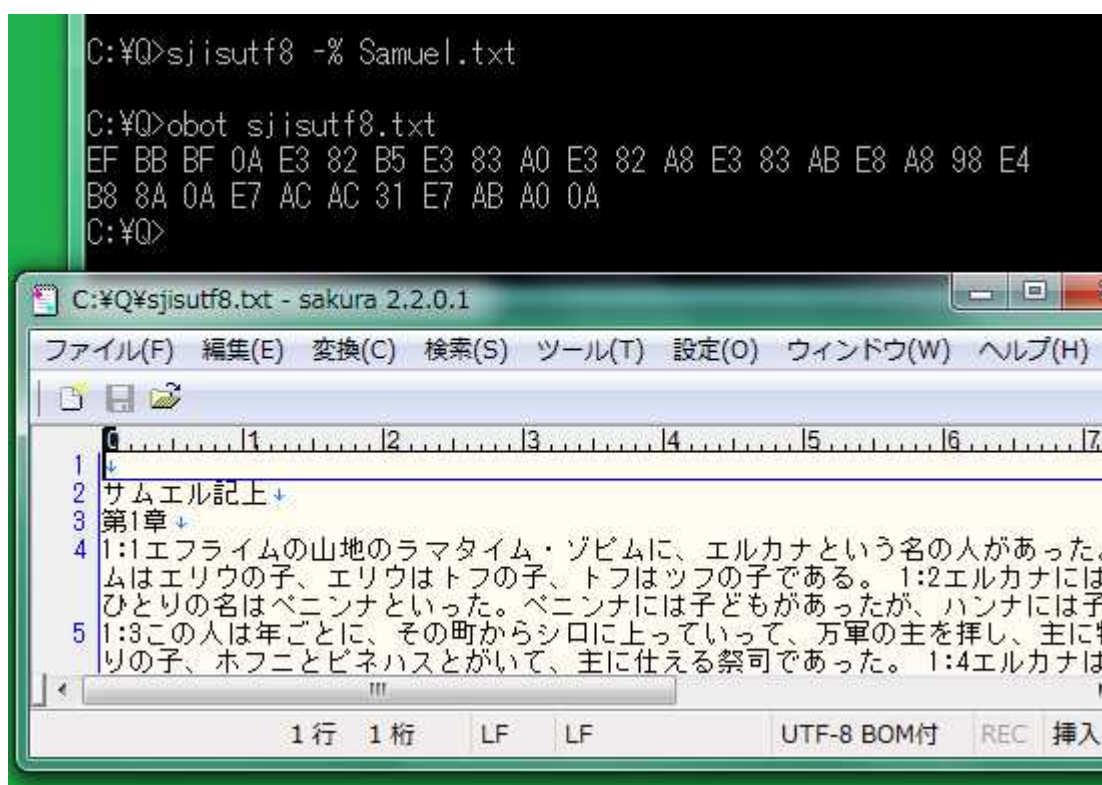
ました。ファイルの終わりにある改行が唯一の相違点であったことを示すために、COMP コマンドでも比較してみました。サイズも内容も一致しました。

```
C:\¥Q>COMP sjisutf8.txt more_Out-File.txt
sjisutf8.txt と more_Out-File.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n
C:\¥Q>
```

obot コマンド

添付の obot コマンドは、ファイルの最初の部分を十六進数で表示できます。3 バイトの BOM として、EF BB BF が sjisutf8 コマンドから出力されたことを obot コマンドで確認できました。サクラエディタのステータスバーに **UTF-8 BOM 付**と表示されました。



FormFeed.txt

添付の FormFeed.txt は、改ページの制御文字(0Ch)が含まれています。-s 付きの sjisutf8 コマンドで 0Ch が削除された例を下図に示します。


```
C:\¥Q>obot FormFeed.txt
50 61 67 65 2D 31 0C 50 61 67 65 2D 32 0C 45 4F 46
C:\¥Q>sjisutf8 -s FormFeed.txt FormFeed_UTF-8.txt

C:\¥Q>obot FormFeed_UTF-8.txt
50 61 67 65 2D 31 50 61 67 65 2D 32 45 4F 46
C:\¥Q>
```

最初の部分を標準出力に書き出す

標準入力から入手したデータの最初の部分を標準出力に書き出すには、obot コマンドのパラメータとして、`-K 数 -p` を付加してください。obot コマンドは、多数の行を読み込んで、1 個のブロックを書き出します。最初の部分は、十六進数ではなくそのまま表示されます。たとえば、ファイルの最初の 1024 バイト未満を表示するには、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。1024 バイト未満のブロックが書き出されます。

TYPE Samuel.txt | obot -K1024 -p

```
C:\¥Q>TYPE Samuel.txt | obot -K1024 -p

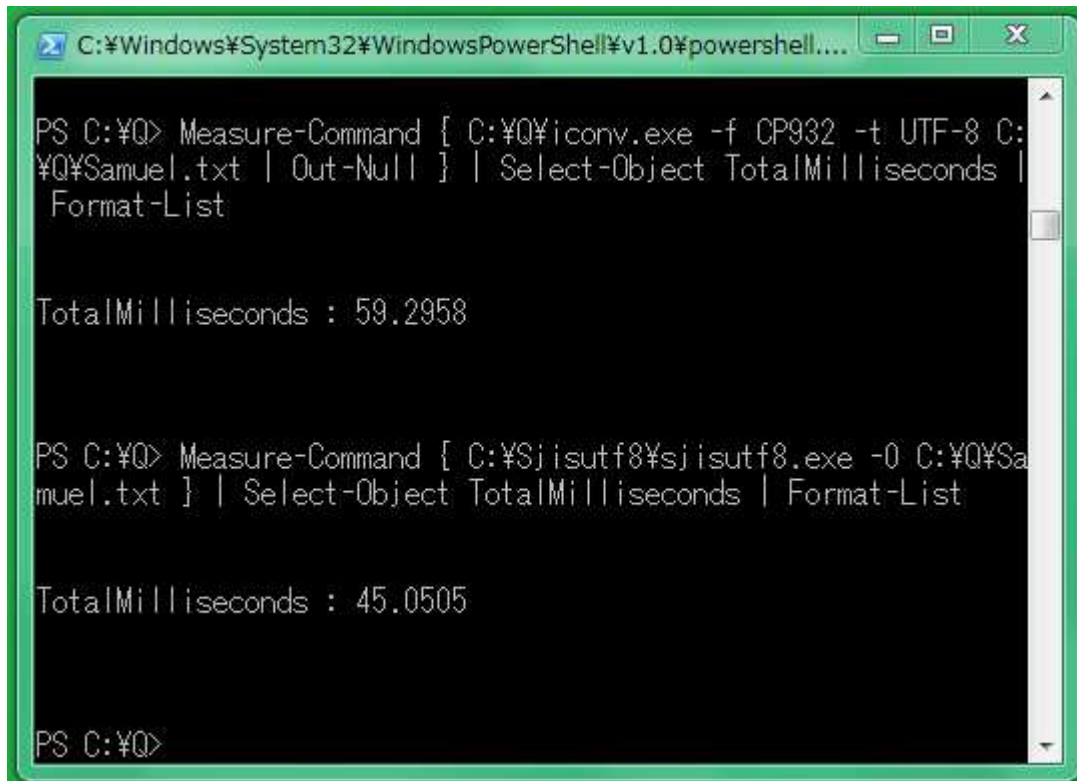
サムエル記上
第1章
1:1 エフライムの山地のラマタイム・ゾビムに、エルカナという名の人があった。エフライムひとりで、エロハムの子であった。エロハムはエリウの子、エリウはトフの子、トフはツフの子である。 1:2 エルカナには、ふたりの妻があって、ひとりの名はハンナといい、ひとりの名はペニンナといった。ペニンナには子どもがあったが、ハンナには子どもがなかった。
C:\¥Q>
```

iconv および Out-Null のパイプ

iconv コマンドおよび Out-Null コマンドレットのパイプを実行して、経過時間を計測してみました。Out-Null のパイプは、コマンドプロンプトで NUL にリダイレクトすることに似ています。more コマンドと異なり、iconv コマンドは、PowerShell に適合していないおそれがあります。

Out-Null コマンドレット

sjisutf8 コマンドは、期待通りの `sjisutf8.txt` を作成しましたが、Out-File と異なり Out-Null は、何もファイルを作成しません。残念ながら、iconv コマンドは、sjisutf8 コマンドと同じ結果を生じたか、この計測法では確認できません。



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell....

PS C:\> Measure-Command { C:\>iconv.exe -f CP932 -t UTF-8 C:
>Samuel.txt | Out-Null } | Select-Object TotalMilliseconds |
Format-List

TotalMilliseconds : 59.2958

PS C:\> Measure-Command { C:\>sjisutf8.exe -O C:\>Sa
>muel.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 45.0505

PS C:\>
```

UTF-8 から UTF-16 に変換

添付の utf8to16 コマンドは、UTF-8 のテキストファイルを UTF-16 のファイルに変換します。BOM 付きの UTF-16 ファイルが FFh, FEh で開始することを obot コマンドで確認できます。下記の UTF-16 のファイルは、00h が含まれています。

BOM 無しの UTF-16 ファイル

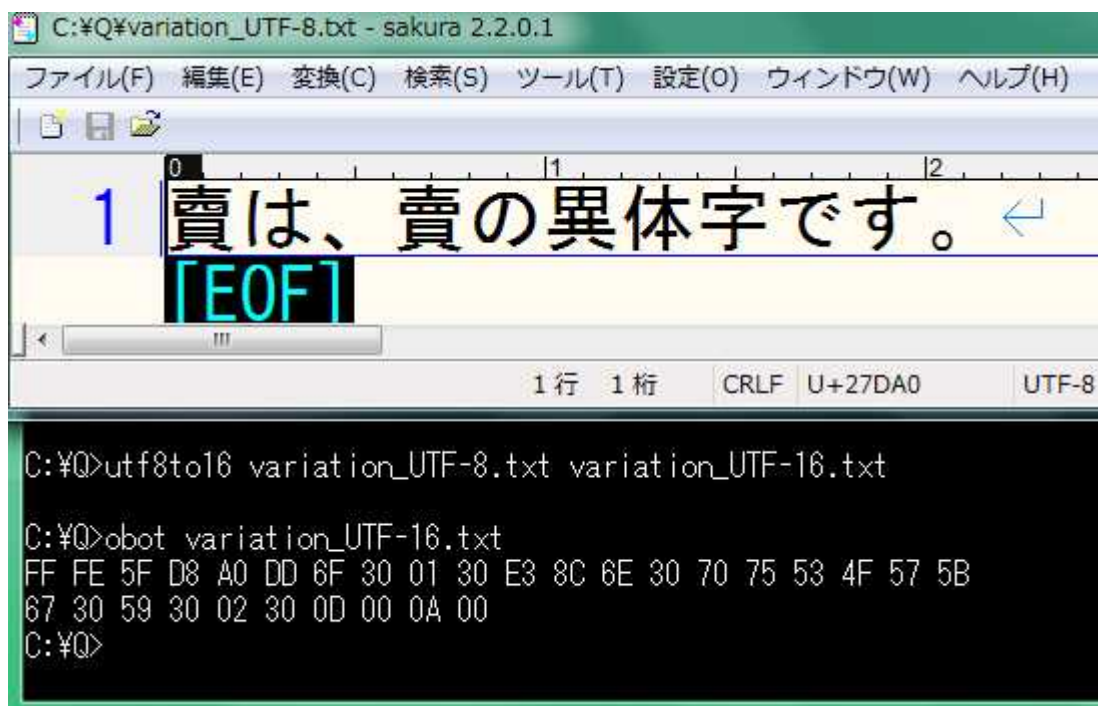
ファイルの最初に BOM を付加しないように命令するには、utf8to16 コマンドのパラメータとして、-m-を付加してください。

変換表が不要である

UTF-8 も UTF-16 も Unicode の一種であるため、utf8to16 コマンドは、まず符号位置を算出して、次に符号位置から計算して UTF-16 の形式で出力します。変換表を使用しません。

サロゲートペアをサポート

utf8to16 コマンドは、サロゲートペアをサポートします。サクラエディタのフォント設定で、サイズを 36 に設定して、27DA0 の符号位置にある漢字を表示した例を示します。サロゲートペアは、D85F,DDA0 です。なお、賣は、8CE3 の符号位置にあります。



BOM

Windows パソコンは、メモリ、ディスクに 16 ビットの整数を記憶するとき、下位 8 ビット、上位 8 ビットの順に並べて記憶します。すなわち、2 バイトの整数を記憶するとき、下位バイト、上位バイトの順に並べて記憶します。これをリトルエンディアンのアーキテクチャーと言います。Intel Core i5 は、リトルエンディアンのプロセッサです。FFh, FEh のバイト順マーク (BOM, byte order mark) を UTF-16 ファイルの先頭に付与することで、リトルエンディアンのアーキテクチャーに適していることを明示できます。

ビッグエンディアン

FEh, FFh のバイト順マークが付与されていれば、2 バイトずつ記憶されたデータは、ビッグエンディアンのアーキテクチャーに適しています。たとえば、Solaris 10 という OS の Sun SPARC は、ビッグエンディアンのプロセッサです。

iconv および Out-File のパイプ

iconv コマンドおよび Out-File コマンドレットのパイプを実行して、経過時間を計測してみました。

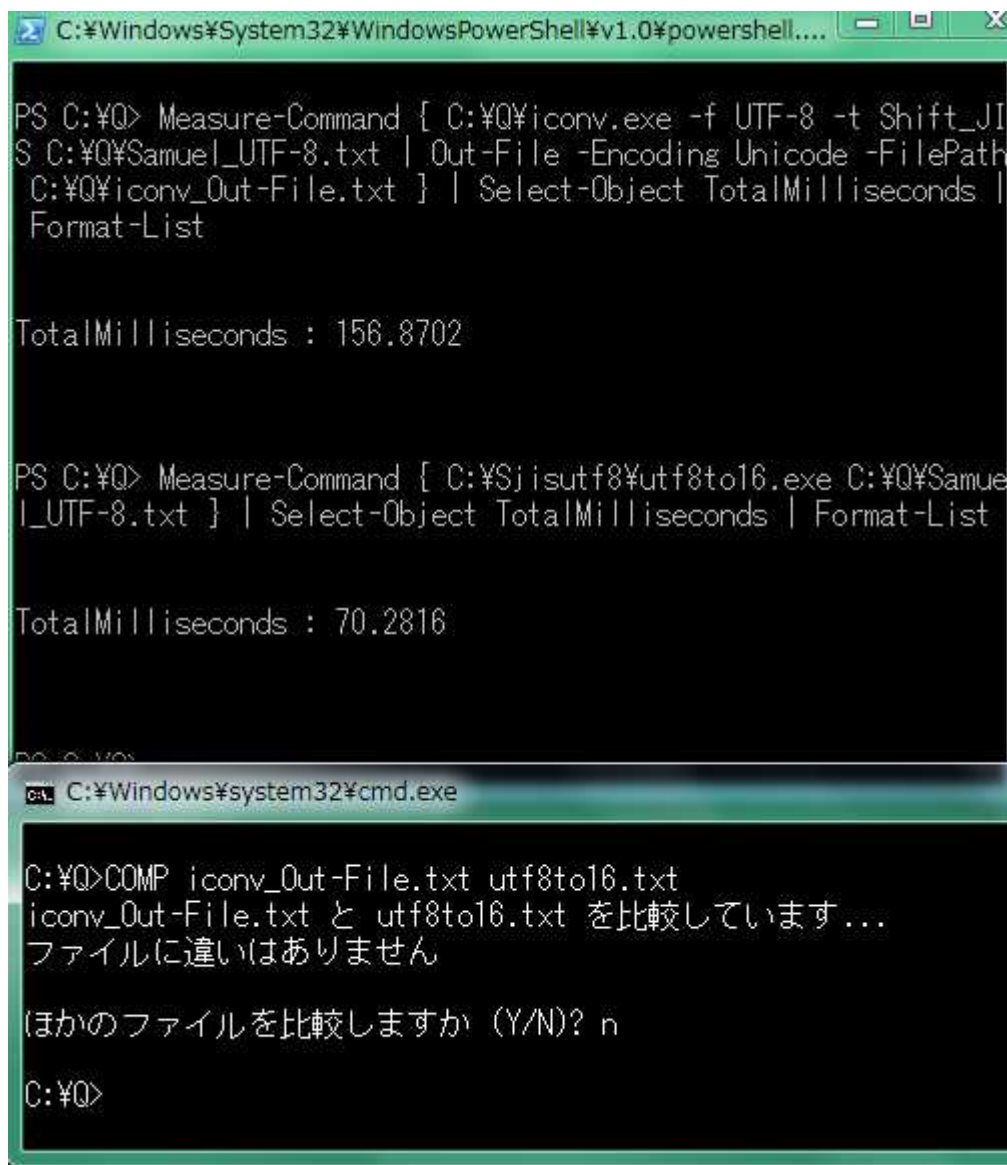
UTF-8 からシフト JIS に変換

iconv コマンドは、UTF-8 のテキストファイルを変換して、シフト JIS のファイルを作成しました。

Unicode のファイルを作成

Out-File コマンドレットのパラメータとして、-Encoding Unicode を付加しました。

iconv コマンドおよび Out-File コマンドレットのパイプは、UTF-8 のテキストファイルを変換して、UTF-16 のファイルを作成しました。



```
PS C:\> Measure-Command { C:\>iconv.exe -f UTF-8 -t Shift_JIS C:\>Samuel_UTF-8.txt | Out-File -Encoding Unicode -FilePath C:\>iconv_Out-File.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 156.8702

PS C:\> Measure-Command { C:\>sjisutf8\utf8to16.exe C:\>Samuel_UTF-8.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 70.2816

C:\> COMP iconv_Out-File.txt utf8to16.txt
iconv_Out-File.txt と utf8to16.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:\>
```

sjisutf16 コマンド

添付の sjisutf16 コマンドは、シフト JIS のテキストファイルを UTF-16 のファイルに変換します。

設定ファイル

sjisutf16 コマンドの設定ファイルは、シフト JIS のファイルです。インストールディレクトリ(C:\>Sjisutf8)またはカレントディレクトリに、利用者は、設定ファイルとして set_sjisutf16.txt を用意できます。set_sjisutf16.txt の第 1 行に変換表のパス名を記入できます。第 2 行、第 3 行、第 4 行、…に sjisutf16 コマンドのパラ

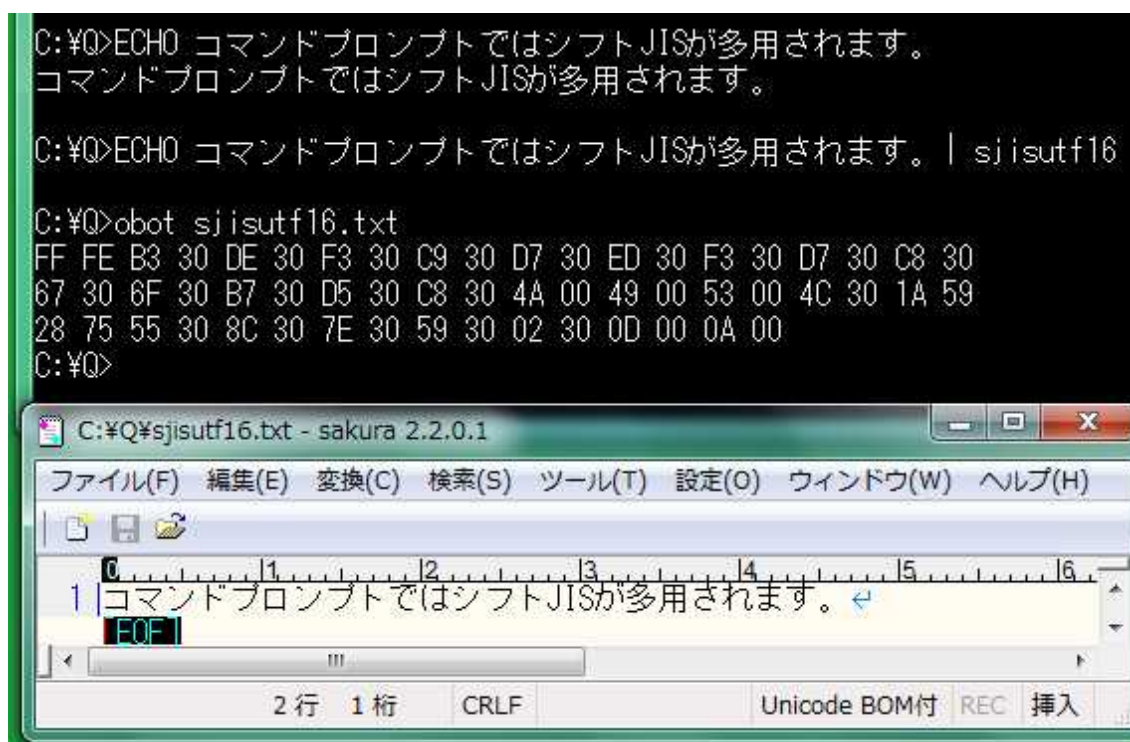
メータを記入できます。

ECHO コマンド

どの Windows も最初から ECHO コマンドを用意しています。ECHO コマンドのパラメータとして入力した内容は、コマンドプロンプトにシフト JIS のテキストとして表示されます。改行がシフト JIS のテキストに付加されます。

パイプ

ECHO コマンドおよび sjisutf16 コマンドで構成したパイプにより、sjisutf16.txt を作成した例を下図に示します。添付の obot コマンドは、ファイルの最初の部分を十六進数で表現できます。sjisutf16.txt は、FF FE で開始することを obot コマンドで確認できます。すなわち、sjisutf16.txt は、BOM 付きの UTF-16LE のファイルです。0D 00 0A 00 で改行されていることも確認できます。このユーザズガイドでは、UTF-16 は、BOM 付きの UTF-16LE を意味します。



The screenshot shows a Windows command prompt window with the following commands and output:

```
C:\>ECHO コマンドプロンプトではシフトJISが多用されます。
コマンドプロンプトではシフトJISが多用されます。

C:\>ECHO コマンドプロンプトではシフトJISが多用されます。 | sjisutf16

C:\>obot sjisutf16.txt
FF FE B3 30 DE 30 F3 30 C9 30 D7 30 ED 30 F3 30 D7 30 C8 30
67 30 6F 30 B7 30 D5 30 C8 30 4A 00 49 00 53 00 4C 30 1A 59
28 75 55 30 8C 30 7E 30 59 30 02 30 0D 00 0A 00
C:\>
```

Below the command prompt is a text editor window titled "C:\>sjisutf16.txt - sakura 2.2.0.1". The editor shows the text "コマンドプロンプトではシフトJISが多用されます。" followed by a blue arrow cursor. The status bar at the bottom indicates "2行 1桁 CRLF Unicode BOM付 REC 挿入".

COMP コマンドで比較してみる

more コマンドおよび Out-File コマンドレットのパイプと、sjisutf16 コマンドを比較しました。下記の例では、ファイルの終わりで Out-File コマンドレットが追加した改行をテキストエディタで削除しました。ファイルの終わりにある改行が唯一の相違点であったことを示すために、COMP コマンドで比較してみました。サイズも内容も一致しました。


```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell....

PS C:\> Measure-Command { more C:\>Samuel.txt | Out-File -En
coding Unicode -FilePath C:\>more_Out-File.txt } | Select-Obj
ect TotalMilliseconds | Format-List

TotalMilliseconds : 314.1765

PS C:\> Measure-Command { C:\>Sjisutf8\Sjisutf16.exe C:\>Samu
el.txt } | Select-Object TotalMilliseconds | Format-List

TotalMilliseconds : 35.6945

PS C:\>

C:\Windows\system32\cmd.exe

C:\>COMP more_Out-File.txt sjisutf16.txt
more_Out-File.txt と sjisutf16.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:\>
```