UTF-16 は、Unicode の符号化方式のひとつです。16 ビットを 1 単位として、ひとつの Unicode 符号位置を 16 ビットまたは 32 ビットで表します。16 ビット固定長の Unicode の元々の形式に基づいて拡張を施したものです。

## 由来

当初、Unicode は 16 ビット固定長で 1 文字を表すのが売り文句の文字コードでした。しかし、16 ビットの符号空間では最大 65,536 文字しか扱えず、拡張が必要となりました。

他方、Unicode と同等の文字コード規格 ISO/IEC 10646 では UCS-4 という 4 バイトコードが定義されており、これを使うと 65,536 符号位置からなる面をいくつも扱うことが可能でした。Unicode の 16 ビットの符号空間は面 00, Basic Multilingual Plane (BMP) という位置付けです。

従って、UCS-4 を使えば BMP 外への拡張の問題は解決するのですが、一方で既存の Unicode の 16 ビットコード (2 バイトコードすなわち UCS-2) を扱うプログラムとの互換性も求められました。

そこで、BMP の中の文字の割り当てられていない符号位置を使用し、2 つの符号位置の組み合わせで BMP 外の 1 つの符号位置を表現する方法が採用されました。これが UTF-16 です。例えば、BMP 外の漢字 U+29E3D を表すのに、上位サロゲート D867 と、下位サロゲート DE3D の組み合わせを用います。

## JIS X 0213 との関係

Unicode が JIS X 0213 の文字を全て含んでいるので、UTF-16 で JIS X 0213 の全ての文字を符号化できます。

ただし、一部の文字が BMP 以外の面にあるため、それらの文字については、UTF-16 ではサロゲートペアを使って、1 符号位置あたり 4 バイトで表されることになります。BMP 内の漢字、例えば「亜」(U+4E9C) は 16 ビットつまり 2 バイトですが、Plane 02 の U+29E3D にある「ホッケ」の漢字 ( 魚へんに花 ) は、上位サロゲート 2 バイト、下位サロゲート 2 バイトで合わせて 4 バイト必要です。

文字列を UTF-16 で扱うプログラミング言語 (Java など ) を使うときは、サロゲートの存在を考慮しないと、こうした BMP 外の文字で問題を生じ得ます。

## 関連項目

- Unicode
- ISO/IEC 10646
- UTF-8