

本日の授業のテーマ

本日の授業のテーマは、以下のとおりです。

- (1) 文字を表現するコード

本日の授業のゴールは、以下のとおり。

- 文字を表すために種々のコードがあることが理解できる。
- 制御コードの役割が理解できる。

1 コードとは

一般に、情報を記号によって表現することをコード化(coding)と呼びます。表現したものをコード(code)または符号といいます。

ここで、記号と言っていますが、コンピューターの世界では数字です。例えば、

- 文字を数字で表す。
- 音を数字で表す。
- 絵を数字で表す。

などです。

2 文字を表現するコード

2.1 文字コード

コンピューター(電子計算機)がここまで、一般に使われるようになったのは、数字のみならず、文字や絵、音が扱えるようになったからです。ここでは、文字の取り扱いに関して簡単に説明します。

今までの授業では、数字の取り扱いの基礎について学習しました。次は文字です。コンピューターが取り扱うデータは、"0"と"1"で表現されるというのは、数字も文字も同じです。絵や音も全て同じです。

文字を表すためには、どうすればよいでしょうか?。考えてください。文字の場合、数のように、10進数を2進数に変換して、"0"と"1"で表現することはできません。そこで、全ての文字に番号を付けるのです。その番号に従い、2進数の整数で表現するのです。コンピューター内部では、この番号に従い、対応する文字を探し出して変換しています。この番号と文字の対応表をコード表と言います。

文字と番号の対応が間違っていると、ぜんぜん違う文字を表示することになります。例えば、Internet Explorer で表示のエンコードを変えてみてください。コードを変化させるとができ、とんでもない表示(文字化け)を起こします。図1の場合が間違ったコードで数字を文字に戻した場合です。

文字と数字の対応は、それこそいくらでも考えられます。いろいろな規格があります。例えば、

ASCII
JIS 8 単位コード
EBCDIC
ISO
EUC
シフト JIS windows はこれを使っている。SJIS とも言う。
JIS コード
日本語 EUC

です。世界中を考えると、文字の種類だけ、コードがあります。となると、異なる文字を web に表示させるとなると、それだけのコードを用意して、そしてどのコードが使われているか見分ける必要があります。これは困った問題です。いずれは解決されると思いますが、今のところ、世界のすべての文字を表現できるコードはありませんので、しばらくは不便が続くと思います。最近、unicode という国際標準文字コード体系の導入が進められています。

幸いなことに、アルファベットと数字、制御コード(あとで学習)はほとんどのコ

ードで同一なので、間違ったコードを指定しても、これらは正しく表示される場合が多いです。

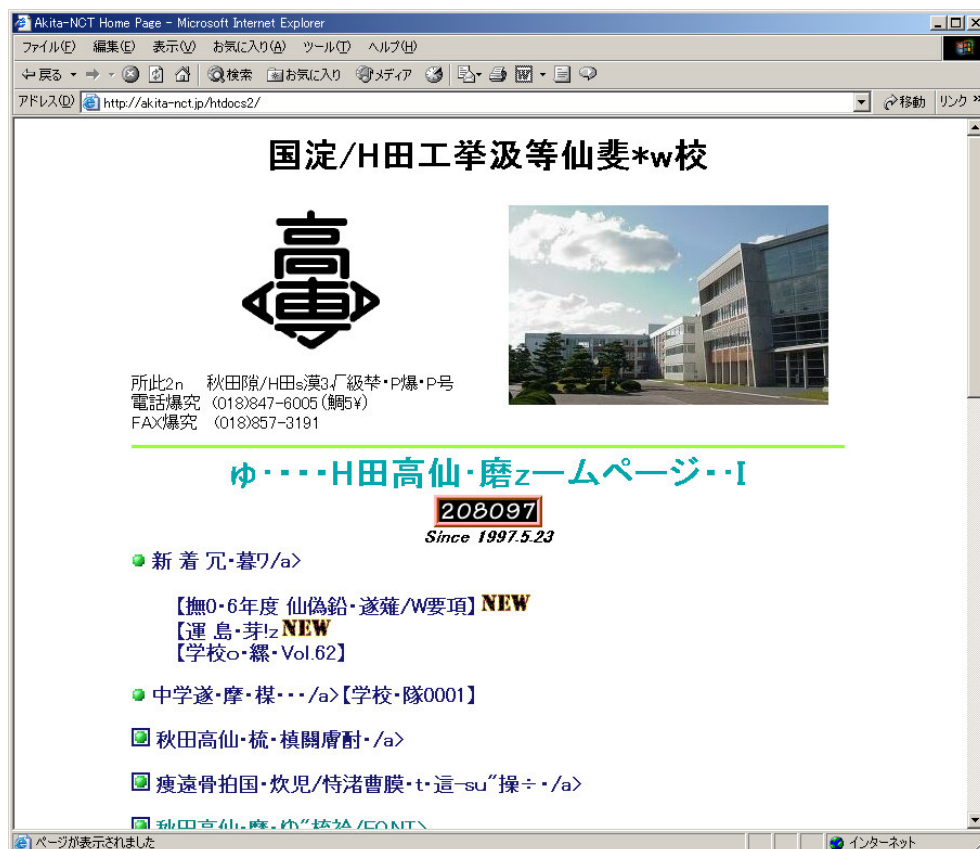


図 1 Extended UNIX Code で表示した場合、文字化けしている。

2.2 制御コード

文字を数字で表すのは、先に述べた通りです。数字で表すことができるから、コンピュータで取り扱うことができます。では、実際に文字を数字で表します。漢字は大変なので、アルファベットにしましょう。それでは、例として

Akita

を JIS8 単位コード(教科書 P179)で表してみましよう。すると、

416B697461

なります。実際にこのようになっているか調べてみましょう。調べる方法は、次のようになります。

- ① テキストエディタ、例えばワードパッドで、Akita という文字を書き込みテキスト形式で保存します。
- ② 保存したテキストファイルをバイナリーエディタで開きます。すると、16進数でファイルの中身を見ることができます。

それでは、実際に Akita という文字を打ち込んで、それをバイナリエディターで開いたものを図 2 に示します。ちゃんと文字がコード化されているのが分かります。

16 進数の 1 文字は 4 ビットです。したがって、1 文字は 8 ビット=1 バイトであることが分かります。このデータを保存すると、5 文字分、即ち 5 バイトの記憶容量を使うことになります。

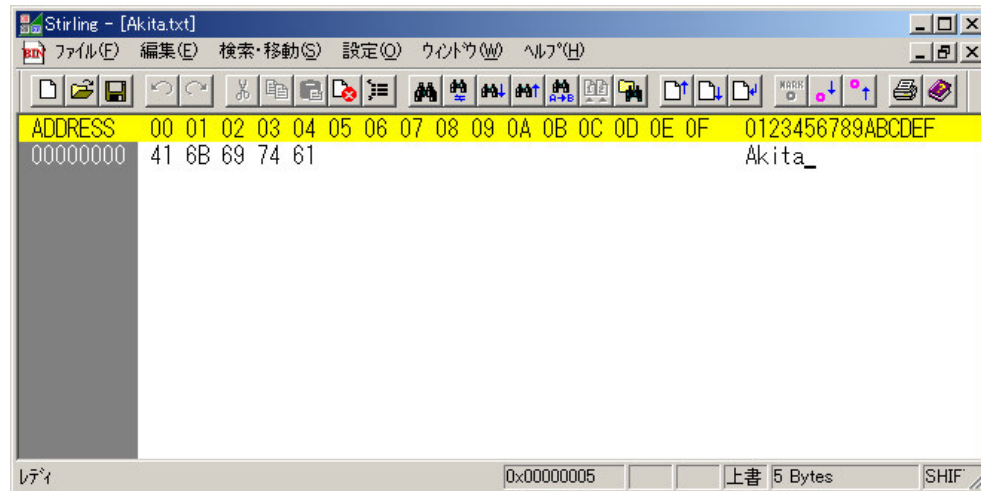


図 2 Akita と書いたテキストファイルをバイナリエディターで開いた。

これで、すべての文字がコード化できる。そうすると、これですべての文章がコード化できることになるだろうか?。文字とは異なり、文章では、次のような場合がある。例えば

```
Akita
Kosen
```

である。Akita のあとに改行されている。文章では、この改行がある。これをどう表すかが問題である。そこで、制御コードというものが考えられた。それでは、実際にテキストファイルを作成して、バイナリエディターで見てみよう。図 3 のようになります。Akita の a の後、Kosen の K のまえに 0D 0A というコードが入っています。これをコード表で見ると CR と LF です。これが、改行の制御コードです。

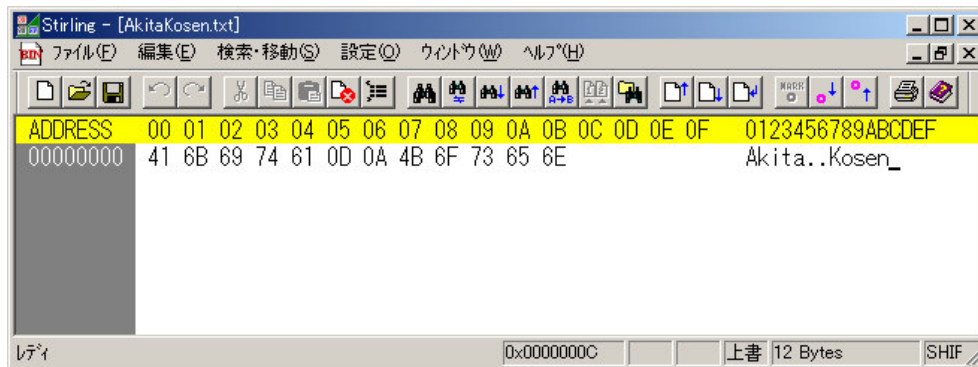


図 2 Akita+改行+Kosen と書いたテキストファイルをバイナリーエディターで開いた。

制御コードを表 1 にまとめておきます。これらの制御コードは、文章を書くためのテキスト制御と通信を行うための制御コードがあります。詳細は、私も知りません。皆さんは必要になったときに調べてください。

表 1 制御コード

下位 4 ビット	上位 4 ビット 0	上位 4 ビット 1
0	NULL:空白	DEL :Data Link Escape
1	SOH :Start of Heading	DC1 :Device Control 1
2	STX :Start of Text	DC2 : 2
3	ETX :End of Text	DC3 : 3
4	EOT :End of Transmission	DC4 : 4
5	ENQ :Enquiry	NAK :Negative Acknowledge
6	ACK :Acknowledge	SYN :Synchronous Idle
7	BEL :Bell	ETB :End of Trans. Block
8	BS :Backspace	CAN :Cancel
9	HT :Horizontal Tabulation	EM :End of Medium
A	LF :Line Feed	SUB :Substitute Character
B	VT :Vertical Tabulation	ESC :Escape
C	FF :Form Feed	FS :File Separator
D	CR :Carriage Return	GS :Group Separator
E	SO :Shift Out	RS :Record Separator
F	SI :Shift In	US :Unit Separator

まとめ

- コードには文字コードと制御コードがある。
- 制御コードの役割は、通信制御やテキスト制御その他
- 制御コードは表示されないが、どこかに書き込まれている。