文字コード概要

C言語の場合

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

int main()
{
    /* 2つの文字列を結合する：C言語*/
    const char* strA = "Hello, ";
    const char* strB = "world.";

    /* 領域確保 */
    char* strC = (char*)malloc(strlen(strA) + strlen(strB) + 1);

    strcpy(strC, strA); /* strAをstrCにコピー */
    strcat(strC, strB); /* strBをstrCに結合 */
    puts(strC);

    if (strcmp(strC, "Hello, world.") == 0) /* 比較 */
    {
        puts("ok.");
    }
    free(strC); /* 領域解放 */
}
```

C++の場合

```
#include <iostream>
#include <string>

int main()
{
    // 2つの文字列を結合する：C++
    std::string strA = "Hello, ";
    std::string strB = "world.";

    std::string strC = strA + strB; // 結合
    std::cout << strC << std::endl;

    if (strC == "Hello, world.") // 比較
    {
        std::cout << "ok." << std::endl;
    }
    // 確保された領域は自動的に解放される
}
```

- std::stringクラスの実体は`std::basic_string<charT , traits,Allocator>`というテンプレートクラス

文字コード

- 文字は数値、文字列は数値の並びとして扱われる。
- 入出力の際に文字と数字との対応表を文字コードという

昔の文字コード

> ASCIIコード

- 7bit(0〜127 : 0x00〜0x7F)で表現できる数値のそれぞれに文字を割り当てたもの
- 昔は英数字、数字、いくつかの記号、改行やタブなどの制御記号しか扱っていなかった

今流行りの拡張

MBCS : Multi Byte Character Set

- 複数のcharの組に1文字を割り当てる。
  - 半角文字は1バイトだが、全角文字は2バイトとか3バイトとか
- 以前から使ってきた文字列(char*,char[])をそのまま使えるが、char配列の文字列の長さと文字列の長さが一致しなくなる。

C 日本語文字列 - yonewiki

- UTF-8、Shift-JISなど。
- **std::stringはMBCSを扱っている。**

WSC : Wide Character Set

マルチバイト文字とワイド文字

- charより大きなサイズの型「wchar_t」を用意してwchar_tの列で文字列を表す
- **ワイド文字列**とも呼ばれる
- 常に16ビットなので、英数字しか使わない場合はMBCSより文字列を確保するのに必要な領域が増える
- Unicode(UTF-16)
- **std::wstringがWSCを扱っている**

Revision #1
★Created 8 June 2025 05:30:25 by 西川和樹
Updated 2 June 2026 18:29:34 by 西川和樹