文字コード

・この本からのメモ
・文字コードについての非常にわかりやすい解説があるため以下にメモ

ワイド文字列とマルチバイト文字列

・ワイド文字列とマルチバイト文字列は C 言語における用語

■マルチバイト文字列 (multibyte character)

・C 言語の char には 1 バイトしか格納できないため、Shift-JIS なり EUC なり UTF-8 と呼ばれる エンコーディング を利用して複数バイトで日本語の一文字を格納する。

問題点

・この方式だと、char 配列を最初から見ていかないとどこが文字の切れ目か判定できないという問題がある。
・文字列の検索を単なるバイト列の検索とすると、Shift_JIS で「\」を検索すると「表」の 2 バイト目にマッチしたり、EUC-JP で「海」を検索すると「ここ」にマッチしたりする

■ワイド文字列 (wide character)

・マルチバイトの問題点を踏まえ、日本語を含む十分なサイズの型で 1 文字を表現すればよいという発想
・C 言語での wchar_t 型であり、この配列で文字列を表現したのが ワイド文字列

C 言語でのリテラル

・ワイド文字

```
L'a'
```

・ワイド文字列

```
L"abc"
```

マルチバイト文字 / ワイド文字変換関数

■メリットデメリット

文字列	メリット	デメリット
ワイド	文字の切れ目を走査して判定する必要あり	データサイズは小さくてよい
マルチバイト	1 文字のサイズが固定のため、プログラムから扱いやすい	1 バイトで表現できる文字にも余分なサイズを消費してしまう

・ファイルに書き込むときには、サイズの小さいワイド文字列で格納し、プログラム上 (た

とえばエディタ) ではマルチバイト文字列を使用すると都合がよいため、さまざまなマルチバイト、ワイド文字列変換関数が存在する。

Unicode

・現状では、Linux でも Windows でも、ワイド文字として Unicode を利用することが多い

■歴史

・Xerox が提唱し Unicod Consortium により制定
・当初、世界のの文字を 16 ビットで表現するというコンセプトだったが、結局 16 ビットでは不足 (現状 21 ビット)

UCS2

・16 ビットで収まる範囲の文字一式は UCS2(Universal Coded-Character Set の 2 バイト版)
・Java で利用される Unicode はこれ

USC4

・USC2 に収まりきらなかった文字を含め、すべてを 4 バイトで表現

■文字集合 (character set)

・コンピュータで文字を扱うために、対象とする文字をを決め、番号を振た文字の集合
・Unicode の場合、文字にそれぞれ振られている番号をコードポイントと呼ぶ

■エンコーディング (character encoding scheme)

・文字コード (Unicode ではコードポイント) をメモリやディスク上にどのように表現するかは文字コードとは別の話でこの論理的な値を、バイトやビットにどう表現するのかを定めたものをエンコーディングと呼ぶ
・Shift-JIS と EUC は対象とする文字集合はどちらもほぼ同じ (JIS X0208) だが、メモリ上の表現形式が異なるため、異なるエンコーディングと位置づける

■ Unicode のエンコーディング

UTF-16

・1 文字当たり 2 バイト割り当てる
・バイトオーダーにより、UTF-16BE と UTF-16LE の 2 種類

UTF-8

・UTF-16 では、アルファベットを表現するにも 2 バイト消費してしまう上、既存の ASCII コードとの互換性もないため、考えられた。
・ASCII と同じ部分は 1 バイト (0x00 〜 0x7F)、その他の部分を 2〜6 バイトで符号化
  ・http://ja.wikipedia.org/wiki/UTF-8

機種依存文字

・コンピュータによる情報の表現と処理
・JIS X 0208 と 0213 と機種依存文字

■ Cp932 整理

・ウィキペディア Microsoft コードページ 932 に詳しい。

CP932

名称	内容
Windows-31J	Windows 3.1(J) のリリースに合わせて、マイクロソフトが JIS X 0208 に 機種依存文字（NEC 特殊文字、NEC 選定 IBM 拡張文字、IBM 拡張文字）を統合して作られた文字コード
MS932	Java で、「IBM のコードページ 932」と「Windows-31J」を区別する
CP932	MS-DOS と Windows における日本語コードページを表す。「Windows-31J」が制定されるまでは、OEM ベンダによって文字集合が違う。
MS 漢字コード	「CP932」とほぼ同じ意味。
OEM コードページ 932	Windows 3.1 日本語版の発売以前における、OEM ベンダ各自の拡張を許した仕様の文字セット

公的機関からも認められた文字符号化方式

名称	内容
シフト JIS	JIS X 0208 符号化文字集合を一定の規則に従ってシフトした文字符号化方式。具体的な内容は JIS X 0208:1997 に「シフト符号化表現」として記載がある。
Shift_JIS	「シフト JIS」の IANA 登録名
SJIS	Shift_JIS の短縮形。Java では Shift_JIS と同義語

■ CP932 の誕生と発展

・CP932 が、現在の「Windows-31J」の形として完成に至るまでには複雑な経緯がある。
・1982 年（JIS X 0208-1983 策定の前年）、JIS C 6226 を複雑にシフトさせた文字符号化方式として Shift JIS が誕生。
・Shift JIS はマイクロソフトにより、MS-DOS における標準日本語コードとして採用「コードページ 932(CP932)」という管理番号を与えられた。
・マイクロソフトは MS-DOS における唯一の日本語用コードページである「CP932」を OEM メーカーの自由に任せていた。
・NEC の PC-9800 シリーズ、IBM の PS/55 シリーズ、富士通の FMR シリーズなどは全て MS-DOS を搭載しており文字符号化方式も Shift_JIS を採用しているが、登録されている文

字集合がバラバラ

OEM コードページの統合

- マイクロソフトは 1993 年、Windows3.1 の日本語版を出すにあたり、「CP932」の仕様を OEM メーカーの自由に任せるという方針を撤回。
- 日本のパーソナルコンピュータ市場で、特に大きなシェアを持つ IBM、NEC2 社の統合コードを Windows における日本語標準コードとした
- これを IANA に「Windows-31J」という名で登録

統合の概要

- ベースとなる符号化文字集合として JIS X 0208-1990
- NEC が 9 - 13 区に登録していた特殊文字の内、13 区のものだけを継承。この 13 区登録の文字のことを「NEC 特殊文字」と命名。
- NEC が 89 - 92 区に登録していた漢字と非漢字は全て継承。このエリアの 374 文字のことを「NEC 選定 IBM 拡張文字」と命名。
- IBM が 115 - 119 区に登録していた漢字と非漢字も全て継承。このエリアの 388 文字のことを「IBM 拡張文字」と命名。

Windows-31J に重複登録されたコード

- 統合の過程で重複する文字が登録されてしまっている。
- NEC 選定 IBM 拡張文字と IBM 拡張文字については、まるごと重複
- 「¬」「∵」については三重複

文字コード変換時の重複文字の影響

- 文字コード変換を行う際に、別の文字コードから、「Windows-31J」に変換する場合に、重複するどちらの文字へと変換するべきかが問題

■ Java での文字コードの扱い

- Java での文字コードの扱い