1 Mule system, Windows System

Mule は多言語処理のできる GNU Emacs System です。したがって、(ごく当たり前のことですが)、Mule を Windows 上に実装することを考えるときにおおまかに 2 つに分けて考えることができます。

- GNU Emacs と Windows の関係
- Mule(Emacs/Mule の多言語処理部分) と Windows の関係

今回のお話しの中では、主に後者のことについてお話しします。前者については、私がかかわった中で興味深いものや Windows のプログラミングをするうえで役に立つことについて、お話しします。

さて、Windows 上に Mule を実装するには (つまり、後者のこと) Windows 上での国際化 (Intenationalization) のことを考えなくてはいけません。選択肢は 2 つあります。すなわち、

- Windows 上の国際化の手法を使って Mule を実現する
- Windows 上の国際化の手段にはまったく頼らずに Mule を実現する

結論を先に言ってしまうと、現在の Mule for Windows では、前者の方法しかサポートしておりません。したがって、ある言語を使うためにはその言語用の Windows が必要になります。

しかし、現在開発中の Emacs20/Mule for Windows は (Meadow と名付けられる予定) Windows 上の国際化手法には基本的には依存せずに作られる予定です。したがって、US Windows 上で多言語処理がすべてできるようになります。

今回はその手法について駆け足で紹介します。順番は以下のようになっています。

1. Windows Internationalization(Windows の国際化とは?)
2. What does Mule contribute to Windows? (Mule が Windows をどう変えていくか)
3. Font Probrem (Font の問題)
4. Mule Input to Windows? Windows Input to Mule?(Input Method) (IME の問題)

1.1 Windows Internationalization

Windows のアプリケーションがある国の言葉を使用するとき、アプリケーション側からは、各国語用のコード系で操作することが普通になっています。通常、日本語を扱うアプリケーションは日本語 Windows95/NT で動作させていたわけです。

しかし、Windows 自身は各国語のバージョンで異なるということは WindowsNT から原則としてなくなりました。NT では各国語用の情報を Code Page の形で持っていて、その Code Page が使える状態であれば、OS レベルでそのコード系が使用可能となります。[1]核の部分は Unicode で固められていて、あとは各国語用のコード系を扱えるようなシステムが周りに控えているという感じで大体あたっているでしょう。

しかし、この Unicode をほぼすべての API でサポートしているのは WindowsNT に限られます。Windows95 では、非常に限定された範囲でしか Unicode が使えません。具体的には、TextOutW など、本当に基本的な画面表示関数と、Unicode ↔ 各国語用のコード系の変換関数の類しかありません。[2]

しかし、これだけでもそれなりのことはできます。現在、Microsoft が出している国際化アプリケーションというと Internet Explorer Version3 があげられます。このシステムは基本的には英語版 Windows で多国語表示を可能とします。これは、基本的には TrueType Font + 各国語用の Code Page 情報でセットになって提供されます。たとえば、この Japanese Version をインストールすると、CodePage932 が使用可能となり、ShiftJIS(正確には Windows31J のようである:-) ↔ Unicode への変換が OS レベルでサポートされることになります。(参照、図 1)

しかし、この方式には大きな欠点があります。

- 現状では Internet Exproler をインストールしていないと、使用できない
- 各言語ごとに使用できる文字コードが、Unicode と用意された Code Page の示すコード系しか使えない
- 使用可能なフォントを取得することが簡単ではない

これらの問題点は無視できるものではなく、むしろ不自由さを感じさせるだけです。[3]

したがって、次のバージョンの Mule for Windows(Meadow) では根本的にこの問題に対処する予定です。変更点は

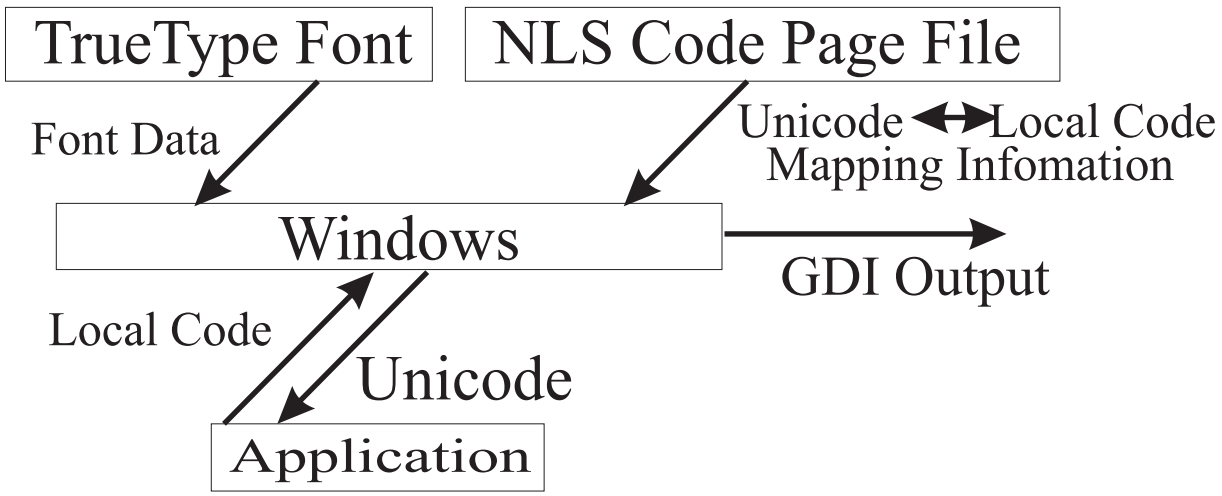
- OS から提供を受ける文字コードは、原則として Unicode(まがい?) とする

[1] 実際 NT4 から、Language Pack なるものが付属するようになっているため、IME の問題や、Localize の問題を除けば (それが大きいとはいえるのですが。) WindowsNT はかなり国際化の能力は持っているといえるでしょう。

[2] したがって、メニュー、ダイアログなどで、Unicode が使えない。

[3] Unicode 自身に欠点は数多く指摘されている、という点に目をつぶるとしてもです。

図 1: Windows(95?) Unicode Support



- 受け取ったコードは直ちに Mule Intenal Code に変換し、あとは Mule が操作をする
- Mule が扱うことのできる文字セットのフォントすべてが扱えるようにする

Windows によって、提供されるフォントは基本的には Unicode で扱うものなので、このさい、次の Mule for Windows では、ISO-10646 の制限的サポートを考えていますが、どのようになるのかは、はっきりとまだ決まっていません。[4]

[4] ボトムアップで構築していくために、ISO-10646-BMP (Unicode のことと考えてください) のみを考えた設計にはならないとは思います。

1.2 What does Mule contribute to Windows?

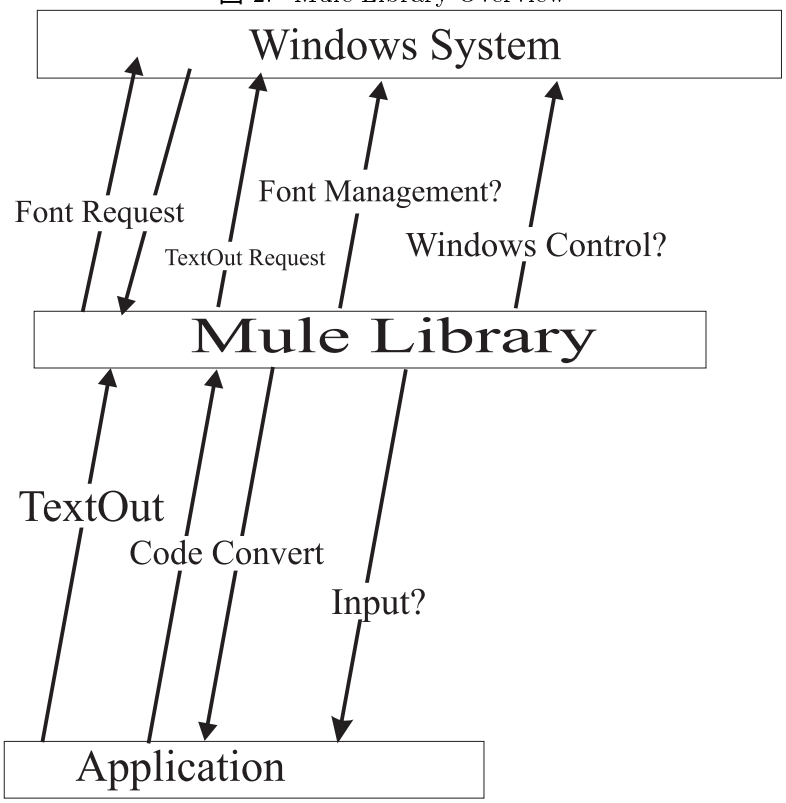
Mule が Windows にのることによって、強力な多国語処理能力を得ることができます。まず、コード系が Unicode に縛られることがなくなります。Mule の持つすべてのコード系を自由に扱うことができるようになり、おまけに多国語処理をすることができるようになります。

現在、Mule の多国語処理部分だけを Emacs から分離して独立したライブラリにできないか検討中ですが、もし、これが可能になった場合、多国語アプリケーションを書くための労力が大幅に軽減されます。(図 2 参照)

まず、US Windows 上で Mule のサポートしている好きなコード系を用いて、アプリケーションを書くことが可能となります。Unicode しか選択肢がなかった状態と比べると相当大きな進歩です。さらに、好きなフォントが使用可能となります。詳しくは後述しますが、Unicode 文字セット以外のフォントを使うことができるようになります。さらに、すくなくとも Mule 上では、多国語の入力が可能となります。Mule の多国語入力システムがそのまま使用可能となるのです。

Mule は長い間、多国語処理の実験場を提供してきました。新しい文字セットにも柔軟に対応する能力を持っています。この能力が Windows 上でも使用できるようになるのでしょうか?

図 2: Mule Library Overview



1.3 Font Probrem

Font の問題はもっとも厄介なものの一つです。Mule では Font を Fontset というもので管理します。これは、Mule Internal Code が LC(Leading Character) で各文字セットを管理していることに起因します。つまり、各 LC に対して 1 つづつ表示する Font を指定することになります。[5]

さて、Windows 上で、たとえば、JISX0208 の Font を使用しようと思ったら、どうしたらよいのでしょうか。一つは、ShiftJIS などの、Windows の用意しているコード系で Font を指定するという方法があります (現在の Mule for Windows はこの方式を取っています)。しかし、この方法では、各文字セットに対して Font を指定することがこのままではできません。Mule はこのために CCL という仕組みを用意して、各文字セットのコードポイントを自由に変換することができます。

たしかに、この方法は一つの解決策です。しかし、US Windows 上で独自にフォントを用意する必要性があるので、この方法だけでは不十分です。したがって、現在用意されている BDF 形式のフォントを Windows のフォント形式に変換して使用することができるようにする予定です。このようにすることによって、現在フリーで配布されているたくさんのフォントを Windows 上でも使用することができるようになります。

さらに、Mule が Unicode をサポートしたときには、Windows の Unicode Encoding 形式のたくさんのフォントをそのまま使用することができるようになるでしょう。

1.4 Mule input strings to Windows? or Windows input strings to Mule?

多国語を入力するという問題も厄介な問題です。Windows にはよく知られているとおり、IME(Input Method Editor) というシステムが存在しており、このシステムの助けを借りることで、多国語の入力が可能となります。しかし、IME を動かすためのシステムは Far East Asia Version にしか付属していないため、US Windows で IME を動かすことは、通常は不可能です。

現実問題としては、これでいいともいえます。なぜなら、US Windows 上で多国語を扱えるシステムはほとんどありませんので、事実上 Mule で入力ができればいいということになります。Mule での多国語入力システムは、すでに多くの物が開発されているので、ここで取りたてて述べることもないでしょう。

また、IME をもっている Windows 上で、それを用いて入力するのであれば、terminal-coding-system を設定すればできるようになるでしょう。

欲を言えば、Mule を入力手段の一つとして使用することができれば、なかなか面白いのですが、これに関しては、また、機会があるときに.....

[5] LC は現在は Leading Code になった。概念的には Charset になりました。

2 WindowsとEmacs

ここからは、Mule for Windowsを作成していった過程で、興味深いことをいくつか紹介します。ちょっとしたことですが、参考になれば幸いです。

2.1 dumpとWindows PE format

これを一番始めに持ってきたのは、これが一番苦労させられたため... です。さて、Mule for WindowsのPE formatの概略は図3に示したとおりです。Emacsの場合にどんな変わったことをしているか、いかに恐ろしいことになりうるかを説明しましょう。

2.2 Message & Emacs event....

おそらく、東田さんも苦労させられた、Emacsの EventとWindowsのMessageとの実装の問題です。これを完璧に解決する手段はMulti Threadしか、W32ではおそらくありません。今回は簡単にしか説明しません。ある意味で東田さんによって、語り尽くされた問題だからです。

2.3 Process Input & Output

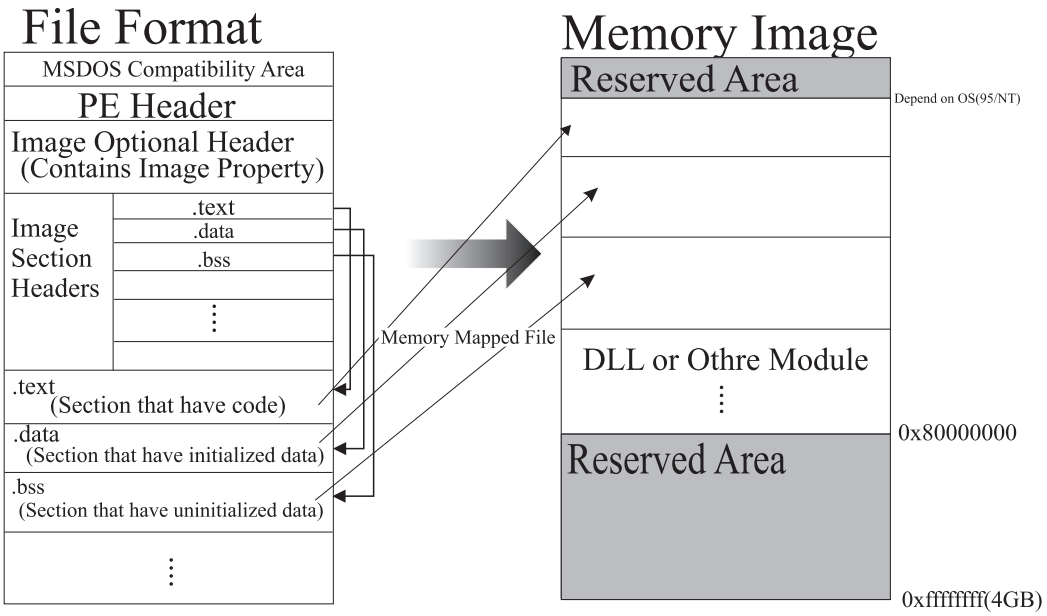
Windowsでは、UNIXのPipeと同じように、Anonymous Pipeが用意されています。しかし、これはちょっと (あまりにも?) 変わっていて、かなり頭の痛い問題です。さらに、(なぜだかまったくわからないが)Anonymous Pipeでは、Overlapped I/Oが使用できません。さらに、Named PipeはWindows95ではサーバーになれません。ある意味で、かなりおかしな仕様と言えます。

EmacsNTの開発チームがとった解決手段はMulti Threadでした.....。[6]

さて、Pipeが非同期で読めないとどういう状況が発生するのでしょうか.....

[6] Windows95では、よく、このThreadが応答しなくなりますが.... 原因をつかみかねているので、もう一度書き起こそうかと考えています。

図 3: W32 Portable Executable Format



In the case of Emacs.....

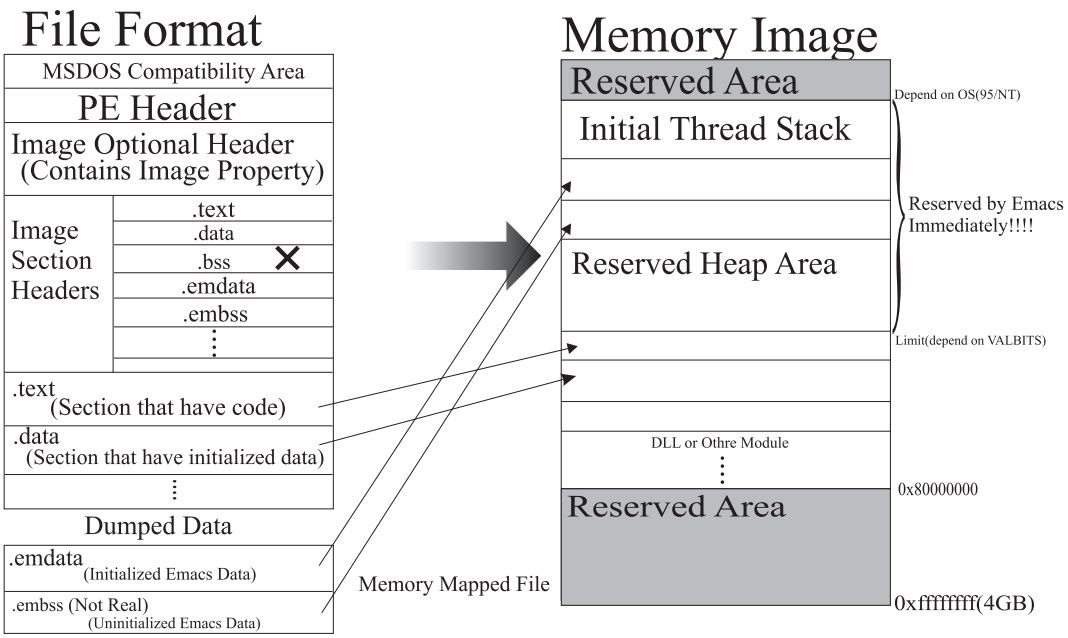
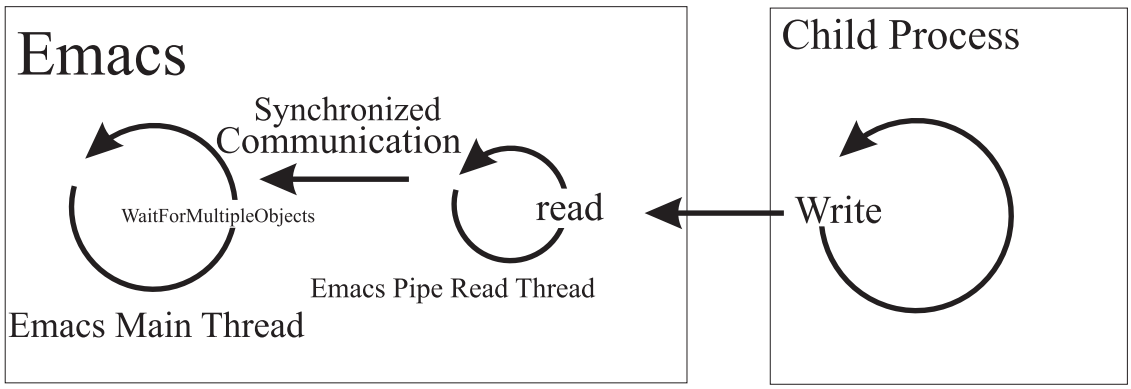



図 4: W32 Anonymous Pipe