Linux用の日本語入力環境Hekeとその周辺

田畑 悠介 (京大マイコンクラブ)

平成13年8月3日

概 要

現在 Linux 上で使用されている日本語入力システムのうち、フリーなものは1990年前後に国産の Unix ワークステーションのために作成されたものをそのまま使用しており、その開発は1990年代後半においてはほぼ停止した状態である。それに対し Linux 自体は性能や使用されている技術も向上し、アプリケーションにおいてもブラウザの mozilla やオフィススィーツの OpenOffice、デスクトップ環境の Gnome や KDE などといったすばらしいソフトウェアがオープンソースで登場しつつある。

このような現在の状況においてアプリケーションと日本語入力システムのギャップは憂慮すべき状態である。また、既存の日本語入力システムは性能の問題のほかに、セキュリティやプライバシーの面で重要な問題を数多くかかえている。本論文ではこの問題と解決策について考察する。最後に、セキュリティやプライバシーの問題、及びフリーで十分にメンテナンスされている日本語入力システムの不在といった問題に対処すべく仮名漢字変換エンジンやグラフィカルな IME などの作成を行っている Project Heke についてその経過と技術的な特徴について述べる。

1 はじめに

我々が現在使用できるオープンソースの日本語入力システム (インプットメソッド) は1990年前後に国内の Unix ワークステーションを作成していたベンダによって作られたものである。これらのベンダは現在のようにオープンソースが話題になっていなかった当時にネットワーク上のさまざまな人達と協力することによってソフトウェアを発展させてきたことがソース中のコメントやドキュメントなどからうかがえる。

日本語入力システムは日本人が Linux を使う上では必須のものであるにもかかわらずそれらのベンダがメンテナンスを停止して以来あまり注意を払われてこなかった。これは言語の処理を行うプログラムを作成するのに非常に高度な技術とセンスと多大な時間を要すると思われてきたことによると思われる。このため、Linux の日本語入力システムは90年代前半のレベルのままで停止しており、Linux をデスクトップ用途で使用することの妨げになっている。

以前のように、Linux がマニアのためのものであればこのような状況でも問題はなかったが、近年のように Linux のさまざまな用途への使用が検討されている状況において日本語入力システムがさまざまな問題を抱えており、解決を試みる動きがほとんど無い状態であることは憂慮すべきである。

本論文では日本語入力システムを構成するさまざまな要素について考察し新たなシステムを構成しようとする試みを容易にすることを目指している。

1.1 注意

- Linux で使用される日本語入力システムの中には多く使われている連文節変換方式の他に、SKK や T-code のようなシンプルかつ一部で高い人気を誇るものが存在するが、本論文中ではそれらについては考察しない。
- 入力システムの国際化は重要なテーマであるが、この問題は日本人でなくても解決することができると考えているので、本論文中では特に考察しない。

2 インプットメソッドの位置付け

アプリケーションからインプットメソッドを利用するためにはアプリケーションとインプットメソッドを接続するためのインターフェースが必要となる。このインターフェースは Windows などで使用されるライブラリ形式と Unix 系の OS でよく使用されるクライアントサーバ形式がある。この二つの形式についていくつかの視点から説明する。

2.1 アプリケーションからの対応

Linux で使用されている Canna や Wnn などのクライアントサーバ形式の入力システムは主にサーバと通信するためのライブラリを用意することによってアプリケーションからの利用を実現している。このライブラリは変換システムごとに仕様が異なるため新たにクライアントとなるアプリケーションを作成する際の負担が大きい。このためアプリケーションレベルで入力システムが実装されているのは emacs や vi などの人気のあるテキストエディタなどに限られており、そのような実装はメンテナンスが困難であるなどの問題がある。

Windows において、アプリケーションからのインターフェースが統一されているのは Windows の API がマイクロソフト一社によって作られたということ及び、Windows のユーザインターフェースが Windows のウィンドウシステムの一種類しかないことが原因だと思われる。それに対して Linux においては端末や X Window System のほかに X Window System 上の各種のツールキットを用いることがあるためライブラリモデルで統一したインターフェースを実現す

るのは困難である。

X Window System を利用する際には XIM プロトコルを利用することになる。この場合クライアントとなるアプリケーションは Xlib の API を用いて XIM サーバとの通信を行うことによって入力を行うことができる。

2.1.1 セキュリティについて

近年、パスワードなどに限らずユーザが入力する内容はすべてプライバシーが保護されるべきものと考えられるようになっており、ネットワーク経由のログインや X Server との接続をする際は LAN の中であっても ssh の使用が推奨されている。

現在のクライアント・サーバ形式の日本語入力システムは暗号化を行わずに通信を行うようになっているのため、ssh のトンネルなどを使わずにリモートホストの変換エンジンを使用することは危険であるので避けるべきである。FreeWnn と Canna は UnixDomain によるローカル接続をサポートしているので、一部のディストリビューションでは、TCP の接続の機能を削除することによりローカルからの使用のみを許すようにすることによってこの問題に対処している。

通信の内容の保護の他にもサーバクライアント型の入力システムの場合、接続時にユーザの認証を行うことが必要である。認証を行わないとユーザ名の詐称を許してしまうという問題がある。それに対して、ライブラリ方式のシステムの場合、通信は発生しないのでこのような問題は発生しない。

2.1.2 個人情報について

日本語入力システムは学習機能によって蓄積される情報やプライベートな辞書などの情報を保持する。この情報はメールの内容などと同様に他のユーザから保護される必要がある。

ユーザの識別を Unix のアカウントを用いて行うのであれば Unix 的なファイルのアクセスコントロールによって各ユーザの情報を保護することができるが、既存のシステムではこの方式は取られていない場合が多い。これは変換サーバが同時に複数のユーザのデータを同時に扱う構成になっていることが原因である。

クライアントサーバ形式一つのメリットとしてユーザの個人情報をネットワーク上のサーバが動作するホスト上に集中させることができるということがあるが、ライブラリ形式を用いる場合はデータを共有ファイルシステム上に保存することによってデータを集中して管理することができる。

3 各種のフレームワーク

この章では入力システムの各種のフレームワークについて説明する。

3.1 XIM

XIM は現在の X Window System の標準の入力システムである。アプリケーションは XIM プロトコルによって XIM サーバと入力のイベントやプリエディット (入力途中の文字列) などを通信する。

XIM の実装は認証や暗号化を含まないが、通信路を X のイベント UnixDomain、TCP のいずれかから選択できるため、ssh や Xauthority などの他のメカニズムを用いることによってそれらの機能を使用することもできる。

3.2 LI18NUX-IM

1999 年、Linux の国際化 API の標準化と共通のフレームワークの作成のために Linux i18n initiative Li18nux.org が結成された。その一部門として入力システムなどの改良を行う li18nux-im が存在する。

li18nux-im では Solaris で使用されている国際化されたインプットメソッドのフレームワークである IIIMF (Internet/Intranet Input Method Framework) をオープンソースにしてメンテナンスを行っている。

IIIMF は IIIMP(IIIM Protocol) で通信を行うサーバの IIIMSF とクライアントの IIIMCF で構成されている。

サーバの IIIMSF はクライアントから要求された言語に対応する language engine と呼ばれるシェアードオブジェクトをロードすることによって各種の言語の入力に対応する。

クライアントの側には

- IIIMECF(IIIM Emacs Client Framework)
  emacs lisp で書かれた emacs 用のクライアント
- IIIMXCF(IIIM X Client Framework)
  Xlib の XIM API を上書きして XIM プロトコルではなく IIIM プロトコルで通信を行うライブラリ
- IIIMJCF(IIIM Java Client Framework)
  Java で書かれたクライアント

が存在する。

プロトコル自体はウィンドウシステムや OS と独立であるため TCP の接続さえ実現していれば Java や emacs のようなクライアントからの使用することができるという利点があるが認証や暗号化の実現が困難になるという問題がある。

3.3 GTK+-IMMODULE

ウィンドウシステム非依存な GUI のためのツールキットである gtk+の現在の開発バージョンである 1.3.x は各種の入力メソッドをシェアードライブラリとして読み込む機能を実装することによって、様々なインプットメソッドを動的に切り替えることができる。

4 Project Heke

我々は Project Heke(Hyper Enhanced Kanji Environment) を行っており以下のソフトウェアの作成を行っている。

4.1 かな漢字変換エンジン Anthy

かな漢字変換エンジン Anthy は既存の変換エンジンへの反省をもとに次のような特徴を持っている。

- ライブラリとしての実装
  変換エンジンのコアをライブラリとして実装することによって各種の実装形態に対応できることを目指している。Anthy を使用したソフトウェアは現時点では IIIMP サーバの Utena と XIM サーバの jmode が存在していて、将来的には gtk+の immodule を作成することを検討している。
- ソースコードの形態
  ソースコードは理解が容易になるように可能なかぎりフラットな階層を取るようにしている。各モジュールのソースにはデータの受け渡しをするモジュールとの関係についてコメントを書くことによって構造を容易に理解できることを目指している。
- アルゴリズムの選択
  よく知られたアルゴリズムを使用するようにしている。アルゴリズムの名前はソースコード中に明記しているため適切な教科書などを参考にすることによって容易にソースコードを理解できることが期待できる。
- シンプルな API
  変換システムと辞書の管理のそれぞれに可能な限りコンパクトな API を提供しており、ユーザとのやりとりを分離して扱うようにしている。
- シンプルな辞書管理コマンド
  個人用の辞書の操作のための手段は辞書の内部表現と外部表現を変換するためのコマンドのみを提供するにとどめている。
- Unix のユーザ管理とは独立した独立したユーザ管理
  変換エンジンはライブラリとして実装しているのでライブラリを使用するプロセスを実行したユーザの権限で動作する。学習履歴などのデータを複数同時に持つことを実現しているが、それらはそのユーザの権限で異なるファイルに保存される。
- ドキュメント
  ユーザとして使うためのドキュメントの他に、使用しているアルゴリズムやデータ構造について可能な限りドキュメントを充実させようとしている。これによってプロジェクトの外部の人がコードを理解しやすくなり、メンテナンスを引き継ぐことも容易になると思われる。

4.2 XIM サーバ jmode

直感的なユーザインターフェースをもった XIM サーバを gtk+ を用いて作成している。Gnome の applet として動作する機能や設定を変更するためのグラフィカルなインターフェースなどを実装しつつある。

jmode は変換エンジンの部分をシェアードオブジェクトとして分離する機能を持っており、現時点では Anthy の他に SKK を使用することができる。

5 まとめ

いままでフリーソフトの世界はコンパイラから OS まで自分たちが必要なものをユーザ自ら作成してきた歴史がある。本論文が、現時点では停滞している日本語入力システムの領域で我々のプロジェクトの他にもプロジェクトが発生して活発に開発を行うことの助けとなることを期待している。