

# 株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月19日 投稿者: SATOXITS

## 続GShell 0.1.5 - そうだ、IMEを作ろう

開発：さて、適当に作って放置していた端末からのコマンド入力用のラインエディタですが。

社長：今はCの外部コマンドとして仮設ですね。カーソルでヒストリが選べないのが実用には問題です。

開発：それで、Goで書き直そうと思います。いや、書き直すというほどの量では無いのですが。

開発：それでせっかく書いたら、エスケープシーケンスだけでなく、ひらがな入力なんかもできると良いかなと思うんです。自力で。大げさに言えばIME。

社長：オートマトンで作るんですか。

開発：いえ、べつに普通に関数の階層的な呼び出しで作れば良いと思うんです。ローマ字かな変換とか、今更カスタマイズなんて必要無いと思いますし、もし必要ならプログラムを書き換えれば良いだけです。書き換えたら再起動してプラグインとして取り込み直せば良い。所要3ミリ秒です。プログラムで書けば、正規表現的な状態遷移に束縛されることもありません。環境情報だって適宜取り入れられます。タイミングとかも。

社長：タイムアウトして確定なんていうのもあって良いですね。

社長：それはそうと、3指モールス符号IMEを是非作って下さい。JとKとLだけで入力できるやつ。

\* \* \*

開発：さてそれでローマ字かな変換ですが。そもそもモードというものが必要かということから。ふつう日本語を入力する状態になったら連続して何十文字も日本語を入力しま

す。もし一文字ごととか単語ごとに、その文字列がカナとして解釈されるべきものだというのを指示するとしたら、面倒です。また、カナに変換することを前提にしてサポートできることもある。例えば子音字の連続による促音への変換みたいなやつですね。だからモードは必要と思います。

社長：CapsLockみたいなものですね。

開発：それと内部表現と外部表現。内部表現的には実際に入力されたキーの列を保持するのが簡単だし、完全にバックトラック可能という意味で面白いですが、外部に表示したり、描画されたデータを取り消すには面倒。ですがこの点は、コンピュータ的には大した処理では無いので、毎回計算し直せばよいかも知れません。

開発：それと実際の端末への表示制御。例えば CR と BS と再描画だけで全部やっちゃうという手もありえますね、行エディタなら。termcap と curses かいらない。ANSIシーケンスとか使わない。とてもシンプル。いっそ入力も矢印キーでなく ^P と ^N、^F と ^B でやる。

社長：うーん、それはちょっと嫌かな。

開発：でも ^K とか ^I とか ^O は抵抗なく使うんですよね？

基盤：そのへんの文字端末コントロールはGoのパッケージにないんでしょうか？

社長：というか、入力モードを表現するのに、色やカーソル形状を有効利用すると良いと思うのですが。

開発：まあそれはいつでも付け足せることですので、とにかくミニマムな実装でやる、というのが出発点として面白いと思います。実はそれで実用のものできちゃったりする可能性もあります。なにせローカル端末なら、1秒に100万文字とか書けちゃうのかも知れませんが、途中でどう書いたかとかユーザには見えないと思います。私は組み込み系で115200ボー、つまりわずか14.4KBの出力でも、これ式の単純実装で実用してました。これは1秒書くのに0.07ミリ秒、100文字書くのに7ミリ秒ですから、ほぼ人間の目には認識できないです。

社長：まあ、ネタとしては面白いですね。プロトタイプだし。

開発：入力モードですが、たとえばカーソルの入力位置からひとつ開けて\*印でも出しておく。

社長：昔 onew ではカーソル近くに[ar]を出すモードがあったような気がします。

基盤：カーソルの色か形状が変わるのがいいけどなあ・・・

開発：ともかく、1文字毎に1行まるごと再描画、というのをまずやります。これの再描画は、プロンプトの左部分に時刻とか表示するのにも必要な機能だからです。どのみち ^L でやるべき処理ですし。

\* \* \*

社長：そもそも GShell なんて無理やりカナに変換しようとするのが変ですよ。

開発：オートマトンの近視眼だからですかね。

社長：辞書の階層に行く前に字句レベルで気づくべきです。あ、どっちにしても字句レベル以下ですね。

開発：プログラミング言語の言語処理系だと、字句レベルスキャナ、構文レベルパーサ、意味レベルということになるわけですが、普通のIMEは基本的に字句レベル止まりですね。

社長：探索っていう意味では、大昔流行った Prolog とかを IME に持ち込めたら面白いんですけど。処理系は比較的簡単に作れるんですよ。私もちっちゃいのは作りました。

基盤：GoでProlog処理系が作られてますね。

社長：いずれ試してみたいです。

開発：きっと誰かLisp処理系も作ってるでしょうね…

社長：PrologでBNFみたいなもので、LL1パーサとして使える気がするんです。！でカットしまる必要はあるかも知れませんが。

\* \* \*

開発：ああそれで、今日のお題のIMEなんですが。入力のinsertモードとoverwrite モードの切り替えもしないといけないですね。

基盤：あれ、InsertキーがDelete や Back space キーの近くにあるもので、しょっちゅう間違えて押してしまって本当に不便ですね。

社長：overwrite にしますよ、いいんですか？本当にいいですね？くらい聞いてきて欲しいですよ。

開発：邪魔なキーの典型として CapsLock と双壁ですよな。

基盤：間違ってoverwriteしてしまった部分を取り戻す undo 機能が必要だと思います。

開発：undo といえば ^Zだと思うんですが、これがコマンドインタプリタだとプロセスをストップさせるキーとかぶっちゃう。かと思うとmacOSのアプリだとコマンドキー+Zだったりするので、これもまた混乱します。

\* \* \*

開発：それで、テキストの内部表現と端末へ出力する文字コードですが、やはりUTF-8かなと思います。

社長：昔OnewとかDeleGateの時代に自作したのはもう腐っちゃってるというか… JISコードとUnicodeの間のマッピングテーブルとか。失われた技術。あまり掘り返したくない気分でなくもありません。

開発：一時期自動判別に血道をあげてましたよね。

基盤：UTF-8関係のサポートパッケージはGoにありますけど、中身の文字コードとかサポートされてますかね…

開発：単にエンコード・デコードだけだったりするかもですね。

社長：UTF-8 と聞くと敬愛するケン・トンプソンを思い出します。

開発：とりあえず[あr]を出したいんで… echo あ | od -c。そうですか。じゃあとりあえず fprintf(stderr, "[\343\201\202r]");

```
iMac% getline  
[あr]!0! *
```

社長：おー、なんかそれっぽいですね。

基盤：「あr」って何ですか？

開発：あ、はローマ字からひらがなへの変換モードにあることを示します。「Ar」はカタカナへの変換。他の事は記憶にありません。

社長：たぶん、Wnn起源の表示じゃないですかね。うーん… どうせやるなら Onew と互換のキーバインディングにしますかね…

開発：それはともかく、とりあえず「Hello, 世界!」を出しましょう。まずは辞書を作りました。

```
// MyIME/0.0.1 辞書 (2020-0819a)
sekai 世界
ko こ
nn ん
ni に
chi ち
ti ち
ha は
se せ
ka か
i い
```

基盤：えらく刹那的な辞書ですねw

開発：とりあえず文字列を左からスキャンして最長マッチを表示して行くという戦略です。

社長：内部表現はあくまでも“konnnichiwa, sekai!” なわけですね。

開発：そもそも、そういう外部表現もあって良いかなという気もしまして。で、この辞書を読み込むと… ああ、なんだかGoで書きたくなってきましたが、とりあえずCで。

開発：ああもう、身体が半分 Go に侵されており、行末のセミコロン忘れまくりですw。で、できました。まずは、Hello, 世界!

```
iMac% getline
--MyIME/0.0.1-- loaded romkana.txt [11]
// MyIME/0.0.1 辞書 (2020-0819a)
sekai 世界
ko こ
nn ん
ni に
chi ち
ti ち
ha は
se せ
ka か
i い
[あr]!0! Hello, 世界!
gotline: Hello, 世界!
```

基盤：要は表示モードを切り替えてるだけですね。

開発：そうです。オリジナルの入力列で記憶しているので、どちらでも表示可能。間違っていたら、ローマ字表示モードにしてローマ字表記として直せば良いというスグレモノw

```
!0! konnnichha, sekai!
```

```
[あr]!0! こんにちは, 世界!
```

```
!0! konnnichha, sekai!
```

社長：新しい世界が見えた気がした(^-^)/

開発：まじめな話、かな文字とか英数字とかASCIIにある記号とかは、一意に決まるものはこれ式で外部表現もできると良いと思うんです。Wikipediaの非欧米文字のページのURLとか悲惨ですよ。

社長：内部表現と外部記憶用表現は一致させて、表示する時に変換するという事ですね。

開発：日本語には正式なかなのローマ字表記があります。これは各国同じようなものではないかと思うんです。

社長：しかし表示の時点でかな漢字変換するとすると一意性が確保できなさそうですが。

開発：まあ漢字とかはUnicodeの文字番号でやるんでしょうね…

基盤：それなら、少なくともHTML中でなら表示できますね。

社長：なんとかローマ字表記の単語を漢字の単語に一意にマップする方法は無いでしょうかね…

開発：辞書を同梱すれば良いかもですね。同じ読みで複数の単語を使う場合には、辞書の中でのIDを添字的にするとか。

基盤：シンプルに、ローマ字表記と漢字混じり表記のUTF-8を併記して、表示する時に選択できれば良いようにも思いますが。HTMLのimageのALTテキスト的な。

開発：うーむ確かに。データのサイズとか気にする時代じゃないですしね。

\* \* \*

開発：これでとりあえず一段落と。で gsh から使ってみる。

```
iMac% gsh
[あr]!1! echo Hello, 世界!
Hello, Ā=Ā, ÂĀŜĀĀ!
```

開発：あれ？

社長：ありがちな試練ですねw

開発：どこで壊れているのかなと・・・

開発：ああ、自分で壊してるんですね。変数やらを置換する関数でマルチバイト対応に処理をしてない。とりあえずそこをスキップして。こんな感じ。

```
iMac% gsh
[あ r]!! -pr Hello, 世界!
Hello, 世界!
```

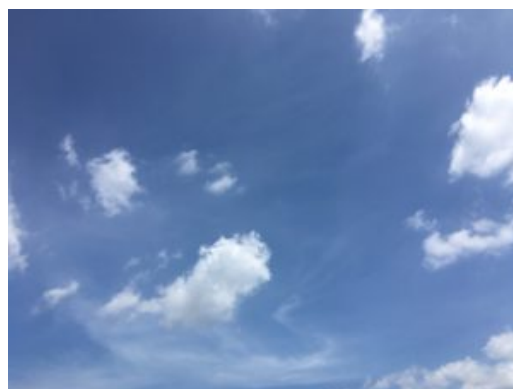
基盤：出ました。

開発：基盤にある処理系を信頼できるというのは楽で良いですね。昔は上から下まで自作だったので、問題が出ると全ての階層を疑ったものです（笑）

社長：ひととおりの見通しも付きましたので食事に行きましょう。今日はしらす丼かな。

\* \* \*

社長：いやはや今日も外は容赦なく夏でした。





社長：でも写真をとりながら思ったんですが、空に雲がなかったらどんなにつまらないだろうって。

開発：よくぞ地球に生まれけりですね。

基盤：エアコンのおかげで室内は30度代に踏みとどまっています。

経理：おかげでdenkoメータが高値安定状態ですが。

社長：節電は生命を犠牲にしてまでする必要はありませんよ。

\* \* \*

社長：それでJKモルス入力はどうになりましたかね。

開発：あー、ちょっと真面目な話にのめってまして… コマンドの実行結果をどう演算するかということなのですが。やっぱり我々はスタックが好きですね。名前を付けなくて良い変数。確実に開放される領域。シンプルな記法。キューもまた良いですね。コマンド間の通信にも使えちゃう。内蔵パイプです。OSを呼ばなくて良いし、Goルーチンさえ使わないので、内蔵コマンドのパイプライン処理を超高速にできそうです。

社長：問題はコマンドの実行結果をどう標準的に表現するかですね。

開発：基本的には可変長文字列の可変長配列を、可変長配列にすればいいんじゃないかと思うんです。要するにコマンドヒストリと同じ構造です。ただ、これもやはりヒストリと同じ構造ですが、各エントリには時刻とリソース使用量をデフォルトの属性として付ける。あと、それが実行されたワーキングディレクトリ。つまり、いつどこで何がされた結果であるかということです。これはファイルの属性と名前のセットを表すのにもちょうど良いです。

社長：サブジェクト側とオブジェクト側が同じ構造で表現されるというわけですね。



開発：で、少なくとも内蔵コマンドは全てこの標準結果出力的な構造のデータを残すことにします。そうすると、実行結果の間のいろんな演算が、共通の標準的なデータ処理コマンドで処理できるわけです。まあまずは集計とかですが。いずれ、実行結果を見て条件判断して実行するなんていうこともできるかなと思います。

社長：それでJKモールスの方は…

開発：まあ辞書を作ればよいだけですが… 我社のIMEは最長一致で変換表示するという、ある意味無敵のIMEなんで、辞書さえあれば… えーと、日本語モールス符号表… みなさんイロハ順でつらい。Googleさんのを見て手入力です。あー、あ、い、う、え、… やれやれ、入力終了です。

```
//ver JA-morse.txt (2020-0819) (^-^)/SatoxITS
.---- .-... .- 世界
--.--- あ
.- い
..- う
-.- え
.-.... お
.-... か
-.-... き
...- く
-.-... け
---- こ
- - - さ
```

開発：でツーをj、トンをk、文字の終端をlということにして、降順ソート。ああ、終端まで完全一致を見るからソートも必要ないですけどね。

```
//ver JKL-JA-morse.txt (2020-0819) (^-^)/SatoxITS
kjjjklkjkkklkj 世界
kkkkkl め
kkkjkl く
kkkl ら
kkjkkkl と
kkjkjkl み
kkjkl ち
kkjjkl °
kkjjkl の
kkjkl う
kkkl *
llllllll ち
```

基盤：これは… 「せかい 世界」って定義できないと、辞書を作るのが辛いですね。

開発：それはまあやればできることです。はい、できました。

社長：どれどれ、Hello, kjjjklkjkkklkj!

```
iMac% gsh
!!
zsh: suspended  gsh
iMac% gsh
!! echo Hello, kjjjklkjkkklj!
Hello, kjjjklkjkkklj!
[あ r]!2! echo Hello, 世界!
Hello, 世界!
```

社長：できた！ううう、長年の夢が叶いました。。。あとはビープ音とか、読み上げが出るとうれしいな。

開発：それはまたおいおい。ところでこれ、終端まで見ずに現在の最長一致を表示させると、それはそれで面白いですね。

社長：とうたすかかかのモールス版みたいですね。

基盤：これは、3指の固定位置でブラインドタッチ入力できるのがメリットということですね。スマホのアプリを作る土地勘が無いのですが、最近買った7インチのディスプレイはタッチパネルなんです。あれで試してはどうかと。

開発：まあそれもおいおいですね。そもそも、モールス符号で定義されてない文字をどうするのかという、技術的では無い難しい問題があると思います。そこで結局ブラインドタッチできなくなっちゃうと、有用性ががっかり落ちてしまいますね。

社長：3指ブラインドを想定した新しい符号体系を考えると良いかもです。4回のタッチを考えれば、3の4乗で81文字を4タッチでいけますからね。

開発：モードを導入すれば、アルファベットも数字も記号も3タッチでイケますよ。かなが難関ですけど。

基盤：複数のキーを同時に押せばいいんじゃないですかね。7×7で49文字を2タッチでイケます。

開発：それは考えた事ありますが、多分タッチが難しい。連続入力には苦しいと思います。まあ、ファンクションキー的には良いかもですが。全押しで1文字消去とか、長押ししたら全消去とか。

社長：21世紀のモールスをめざしちゃうかな（笑）

— 2020-0819 SatoxITS

