

# 株式会社 ITS MORE

2020年4月設立



ITS more

2020年8月24日 投稿者: SATOXITS

## GShell 0.2.0 - 第3フェーズ

開発：内蔵ビューアのCSS+JavaScriptが面白くて、GoでGShellの本題を忘れてしま  
いそうです。

社長：これはソフトウェアの読みやすさ書きやすさ使いやすさを支配する支援機能です  
から、本体のGoコードと両輪ですよ。仕事しにくい環境は一桁も生産性を下げてし  
まいます。

開発：ただ、Goで書いたコードと、JavaScriptで書いたコードが物理的には同じファ  
イルに同居しているけれど、論理的には別世界にいるのがつまらないなど。特に、Go  
のランタイムからは、自分を表現しているHTML、CSS、JavaScriptが見えない。

基盤：ビューアとしてだけでなく、実際に編集して実行できると良いですね。textarea  
にあるコードは編集できるわけです。

開発：うーん。GoのWebAssemblyでextension作れば可能かもね。

基盤：Goの部分はさておき、HTML部分は編集して表示してみることはできますよ  
ね。

開発：・・・それ、面白いかも。

社長：HTMLの使い方のガイドとかでは良くやっていますよね。あれ式が良いと思いま  
す。

開発：いや、そもそもGShellをHTTPサーバとして実行しておけば、そこにブラウザで編集したコードを送って、run するなり build するなりして実行するという手もあると思うんです。ブラウザ自身で実行というのは敷居が高いです。

社長：それは面白そう。それやってみましょう。

\* \* \*

開発：まずユースケースシナリオを考えます。ユーザはブラウザでGoのソースコードを編集してサーバに送ります。サーバはそれを受け取って実行して結果を返す。

開発：この時送り込むコードはGShellのコードそのものとして、そもそもHTTPサーバとしての機能を含むGShellをその場でビルドして実行する。つまり、自分が送り込んだHTTPサーバと通信できるようなのが良いと思います。

社長：プラグインとして生成して動的リンクするくらいが穏便なのは。

開発：それでも良いですね。自在です。それで、必要な要素は以下。

- 要件
  - ① HTTPのPOSTでコードを送り込める
  - ② GShellコマンドと応答がやり取りできる
    - IME機能、インスタントな履歴検索？

開発：①はありふれた機能ですが、わたしはこれをGoで実装した経験はありません。②も普通にCGIでやってきた式でやれば良いと思います。すでに試行はしています。課題は、②のコマンド入力でGShellのIME機能を使えるようにするか？と思います。

基盤：いまどきならAjaxとか使うんでしょうけど、250msのレイテンシーの世界では、まともに使えないでしょうね。

開発：なのでこの部分は、やろうと思ったら extension としてブラウザローカルに組み込むか、JavaScriptでIMEを書くか、という選択肢しか無いと思います。

社長：うちのIMEはその場で変換でなくても良いので、半二重の世界でも使えると思います。

開発：そういえば、こんにちは世界辞書をGShellのビルトイン辞書にして置きたいと思  
いますw

社長：全部のひらがな入力と矢印記号あたりもビルトインしておくともよいのでは。あ  
と、JKモールスも 😊

開発：あーそれ、忘れないうちにやっとうと思います… data URI 大活躍ですね。

基盤：またコードが数百行膨らむんでしょうか？

開発：BASE64に詰めちゃうので、せいぜい数十行ですよ。あ、それで思い出したんで  
すが、せっかくなのでGoとJavaScriptとの間でなにか実行時情報を共有したいという  
件。

開発：ずっと面倒くさいと思っていたんですが。GShellのバージョン番号と日付を、  
Go用とHTML用に置いているので、更新するのが面倒なわけです。なので、Goで定義  
した変数のコンテンツを、JavaScriptで検索して、JavaScriptの変数としても使えるよ  
うにしたい。

社長：JavaScriptで自分のテキストをなめて探し出すんですかね。

開発：まあいずれそうしなくてはと思いますが、とりあえず刹那的に、Goの邪魔にな  
らない形でHTMLのspan かなにかで定義して、それを innerHTML で読み込めば良い  
のではないかと。

社長：atoi とか scanf 相当の関数が JavaScript にあればできそうですね。あるでし  
ょうけど。ただ、いずれ全体をスキャンする機能を作るなら、局所的な寄り道な感じは  
しますが。

基盤：せっかくなら、JavaScriptとしてevalしたりその結果を使えたりすると良いんで  
すね。

開発：む、JavaScriptとしてeval… 少なくともウィンドウを生成して中身をwriteして  
closeするというのは、それに相当しますね。でもやりたいのは自分の手元でエヴァッ  
て、自分の環境に作用させたり、結果を利用できことですが。

基盤：JavaScriptでエヴァる… ああ、document.evaluate() という関数があるようです。これじゃないですかね？現状、全ブラウザがサポートしている模様。MSIEを除いてw

社長：つまりこれは、JavaScriptを部分的に使う言語を自作できるということですか。

開発：おそらく。ただ、同じDOMの上で生活している必要はあると思いますが。結局ブラウザのextensionでことでは。

基盤：オフラインでは Node.js を使うとか。

開発：いや、Mozilla か Micromium を GShell の動的ライブラリと言うかGoプラグインとして使えば良いのだと思いますよ。Mozillaに描画させといてDOMを吐かせるとか、Goから突っ込むとか覗くとか。

社長：それは、Mozilla を HTTPサーバ化する計画とどっちが実現性が高いですかね。

開発：サーバ化するほうが簡単だとは思いますが、効率とやり取りできる情報の質・量は動的ライブラリ式が圧倒的ですよね。

社長：でも、普通にユーザが使ってる生きてる状態のブラウザを外のプログラムから利用できる、認証関係の話とかが楽になるでしょうね。これは、ユーザが実際に使っているGShell自体がGShellサーバになって、他のGShell仲間とかにサービスするのと同じ構造です。そもそもどっちがサーバでどっちがクライアントという関係でもなくて、対等もしくは一体です。この構造は、一般的に有益だと思います。

基盤：ユーザが使っている状態のアプリをサーバにすると、セキュリティ、ガーって言う人が居るでしょうね。

社長：いやクライアント・サーバ方式は、ソフトウェアのモジュール間を繋ぐ一形式に過ぎないです。外部コマンドにするか、動的リンク・プラグインにするのと対等な選択肢。一つのアプリが一つのホストの上で動くか、複数のホストにまたがって動くかだけの違いです。繋ぐときには必要な認証はします。でもそもそも、ホスト内でのコマンドプログラムの実行とか動的リンクなんて、相手の認証はすごくユルイですよ。本当に本気なら、サーバに相当するプログラムだって実行するたびに、誰が作ったものかとかコードが改ざんされてないかとか確認する必要がある。クライアントに相当する実行者

の権限をユーザIDとかざっくりした認証するのも怪しいです。だから、性能の問題がなければ、全部クライアント・サーバにしちゃえばいいとも思うんです。システムコールとかライブラリ関数なんかもみーんな。実際、たとえばRPCでできると思います。

開発：システムコールは実際、一種のIPCですよな。

基盤：ホスト内のプログラムとかライブラリとかファイルとかプロセスは、ホスト外からは直接アクセスできないので、ユルインでしょうな。

開発：まあ、値渡ししか出来ないと、流石に今のコンピュータの能力でも厳しいでしょうけど。でもいろんな抽象レベルで参照でも渡せますか。オブジェクトのハンドルみたいな。

社長：分散オブジェクト指向っていうのは、そういう世界なんだと思います。

基盤：というかそもそも、Mozillaにもなにかそういう基盤技術がありますよね。Xなんとか言う… あの、起動するときになんとか怒ってくるやつ。XOOPS？

社長：結局、値渡しだけのクライアント・サーバモデルだと、特にHTTPみたいにセッションレスだと、インターフェイスが単純にならざるをえない制約がかかって、結果的に共通部品化が成立するってことなのかなとは思いますが。

開発：大好きな scanf 専用サーバなんてあったら面白いかもですねw

社長：なんにしてもこういう話は、研究レベルでは何十年も前にやられて終わってる話なんだと思います。今はああいう話が、実用になる基盤がそろった。

\* \* \*

基盤：洗濯機が大好きなんですけど、いつもバカバカしいと思うのが、あの洗濯物の量を測って適切な洗剤の量を伝えてくるという儀式です。ほとんどの場合、キャップ0.5杯に決まってるし、そんなもん手作業で正確にも測れるわけでもないです。

開発：0.3杯とか0.7杯とか言ってきたらびっくりしますよねw

基盤：そもそも大さじ小さじなら規格がありますが、キャップなんて標準化されてるん

ですかね。だいたい洗剤によって洗浄能力も違うし。まあそもそもいちいちキャップに入れたりしないで直接注いじゃいますけど。だからスタートボタンを押したらとっとと始めると。始めながら洗濯物の量を測って、洗濯時間とか決めればいいんです。

開発：というか、もう少し高いランクのだと、自動でタンクから適切な洗剤注入もしてくれませんか。洗う時間だって水の汚れ具合で判断したり、乾燥だって湿度で判断する。

社長：自分で洗濯アルゴリズムを書けると良いのですがね。必要なセンサーがついてて値がもらえれば。

基盤：ご飯を炊く時のお米の量と水の関係はもっと真剣ですよ。もっとも、加熱する前に十分に浸しておくかどうかは成否には支配的ですけど。

社長：我が国の誇る鍛造厚釜技術は20年前には完成してましたよね。わたしはあの時点で完全に満足してました。かれこれ10年くらい炊いた記憶が無いですが。その後何か進化したんでしょうか？

開発：米びつと水道をつないどくと、勝手に洗って炊いてくれるとよいですね。というか、無洗米があるから洗わなくてもよいですか。

基盤：業務用ならあるんでしょうけど、家庭用にリーズナブルの価格のがありますかね？

開発：白ひきのミルの全自動コーヒーマーカーってありますよね。2万円くらいから。昔にあれがあったら絶対購入してましたね。そのうち、焙煎から全自動なんていうのが出たりして。

社長：今はもう20円パックのコーヒーで十分。違いがわからない男。

\* \* \*

社長：ああそれで、GShellでファイル名にあたる部分を指定する型式は、URL互換にしたいですね。data URI なんかも使えちゃう。

開発：それなんです、悩ましいのはリモートのファイル名です。たぶん rcp の時代か

ら、host:path という型式が確立していて染み付いている。pathの頭に / が付いて host:/path ならそれはパス名の一部で、絶対パスということになる。一方URLは [[scheme:][//host][:port]][/]path。URL式に行くと、サーバの指定とパスの指定の間の / はただの区切り記号なんで、省略系を考えると、 host:/path と書いてローカルと区別するのが妥当。

社長：まあこの /path の件は、FTPのURLの解釈ではだいぶ混乱させてましたね。絶対パスにしたかったら ftp://server/%2Fpathですよって。

開発：要するにURL的には : の後はポート番号であると解釈するしか無い。なので、gcp では [host]:[port:]path と解釈しています。

基盤：そうしないと、標準ポートでない場合にかつたるいです。-P port host:path とか面倒くさい。環境変数で与えられればまだ良いのですが。

社長：ファイルのパス名の中に : が含まれる場合との区別が問題ですね。まあ Windows とかでは禁止されてたと思いますが。

基盤：Unixでは普通に使えますね。

社長：実際のところ、リモートの絶対パスを指定したい状況とかあまり無いようにも思いますね。簡単に書けると危険でも有る。

開発：@を目印にするという選択もあり得ると思うのですが、GShellでも普通リモートも自分だし、OSのユーザ名で認証するとも限らないから、かかない可能性が高い。

基盤：わたしは scp と間違っって cp file host:path なんてやっちゃってしまっただけで事がよくあります。そういう意味でこの host:path 記法は危険だと思います。

host:/path なら弾いてくれるんですが。host: なんていう名前のディレクトリはまず無いですから。

社長：rcp r:a b、rcp b r:a という形に倣うのにこだわることも無いと思いますけどね。これは、コピーコマンドがshellの外にあって、リモートホスト r の部分が何であるかという状態をshellが知らないと言うか、補完して上られないのが問題なんだと思います。コピーコマンドがshellのビルトインなら、ftp コマンドみたいに、相手が現在つながっている先、最近つないだ先に決まれば、切り替える時以外は指定する必要が無

い。 `put local [remote]`、`get remote [local]`。このほうが使いやすく安全だと思います。リモートのカレントディレクトリの状態も保持しているから、ほとんどの場合リモート側も相対パスで済みますし。

開発：要するに、状態レス、文脈自由は汎用で処理も実装も簡単ですが、ユーザには負担を強いるということかなと思います。shell とコマンドとの間で引数とかに関してもっと綿密な知識共有が出来ると良い。ftpコマンドが簡単なのは、コマンドがインタプリタと一体で状態と意味を共有してるからですね。当然コマンドの実行結果についてもインタプリタは理解していてその先に利用する。

社長：うーん、何でしたっけ。MSDOSだったと思うんですが、違うドライブにドライブ名:で移動した時に、そのドライブのルートじゃなくて、自分のカレントディレクトリみたいな所に移動してびっくりしたような記憶があります。

開発：それでは、ファイルのやりとりは `gput`、`gget` コマンドとして、ftpコマンドの `put`、`get` コマンドの型式に倣う、「カレントリモートホスト」を相手にする限り、相手サーバを明示する必要は無い。同じファイル名にコピーするならコピー先の名前も省略して良い。という感じですかね。

社長：あとは、リモート側での `chdir` や `ls` と、ローカルホストでのそれをどう区別するかですね。ftpではローカル側については `lcd` なわけですが、GShellではユーザは基本ローカルホストに座しているので、リモートの場合を特殊扱いする。`rcd` とか `rls` とか。

開発：本来はリモートファイルがローカルファイルと同じ名前空間にあるように見えるのがカッコいいとは思いますが、リモートのカレントディレクトリっていう考えが魅力的ですね。

基盤：リモートでなくてもいいんじゃないですかね。カレントディレクトリ以外に、どこかもうひとつのワーキングディレクトリという状態を維持できて、それを `rcd` だか `gcd` だかで移動させる。で、`gput` と `gget` は、そのディレクトリからの相対アドレスということにする。

開発：・・・それがいいかも。

社長：ならば欲張って、「相手のディレクトリ」というのを複数持てるとよいと思いますね。ただ `chdir` という作用を受けると状態を `chdir` と同様に代える変数、みたいな。

開発：実際、「ワーキングディレクトリを同時に複数持つ」というのは、自然だし魅力的だと思います。この線で考えます。

社長：実際に `gwd` にプロセスを貼り付けて、そいつに OS レベルの `chdir` をさせれば良いかもですね。Go ルーチンだとワーキングディレクトリは一緒に動いちゃうからたぶんだめ。いや、ユーザレベルでカレントディレクトリを作って、そこから相対パス名解釈をやれば良いのかな？これも `open` とか `stat` のシステムコールをラッピングしてやればできるはずです。

開発：でもまあ簡単にできる所から… 簡単に考えれば、たとえば `GWD` という変数を作って、`gchdir xxx` は `GWD` の値をそれに従って書き換える、`gput`、`gget` でリモート相当部分の相対パス名は `GWD` からの相対パスとする、`gpwd` や `gls` も同様。他のコマンドでも `$GWD/path` のようにしてそれを参照したり、`GWD=...` というふうに直接書き換えることもできる。そういう感じかと思います。

社長：それで行きましょう。ただ、`$GWD/` という表記は、もっと簡単に書けると良いですね。

\* \* \*

開発：今日の夜はボウリングの試合ですね。

社長：そろそろ出かける準備をしないと。

開発：とりあえず簡単なモックアップを作りました。こういう感じですよ。

```
iMac% gsh
gsh/0.2.0 (2020-08-24) SatoxITS(^-^)/
!1! tpwd
/Users/ysato/gsh
!2! tcd /usr/local
--I-- TWD=/usr/local
!3! tput a b
--I-- tget from=a to=/usr/local/b
!4! tget x y
--I-- tget from=/usr/local/x to=y
```

社長：そうですね。

開発：この tput のメッセージは tput from … ですよね。

社長：でも、gcdじゃなくて tcd にするんですか。

開発：仮置です。g は GNU 系コマンドというイメージがあり、ちょっと違和感を感じます。r は rsh などのリモート系、x は X Window コマンド系。いずれも左手キーです。

開発：t は terget とか twin のイメージなのですが、これもまたどうかなと。まずちょっと t のキーは打ちにくい。理想的には、右手のホームポジション近傍が良いと思います。

基盤：さすがにそれほど頻繁には使わないのでは？

社長：コマンド名に愛着が持てることは大事です。

開発：それでもう、これは一丁目一番地ですが、j はどうかなと jcd, jls, jput, jget, … Join とか Joint とかの J。Joy の J。Joke と Junk の J :-p。従来の shell は fork 方向に対しする支援が主でしたが、GShell では join 方向に力を入れるという意思表示。

社長：Japan の J でも良いかもですね。最近、国籍色のあるソフトっていいなって思ってるんです。Vivaldi なんかもそう。GShell 改め、JShell にしようかな・・・

開発：JavaScriptで書いた Shell みたいな… でも、JavaScript を理解する shell はあって良いでしょうね。

基盤：GJShell とか。GJで。

社長：そういえば以前 jQueen の時、gjqueen というドメイン名を取ったような…

開発：あの頃はブラウザ周りに我々の入り口があると思ってましたけどね。まさかshell が入り口になるとは思いませでした。

\* \* \*

基盤：あれ？ボウリングには行かなかったんですか？

社長：ちょっと気合が入らず、出かけるのが面倒くさくなった… いやいや、そもそも先週の大不調が続いているとすると、休んだ時にもらえるデフォのポイントを下回ってしまう可能性がある。チームに迷惑がかかるじゃないですか。合理的な判断です。

開発：ところで、思ったのですが、リモートシェルとして考えると、やはりリモートサーバ上のファイルをリモートサーバ上のプログラムで処理する事が多いと思うわけです。なので、カレントワーキングディレクトリはリモートにある。そういう時に、時々スポット的にローカルホストのファイルとかコマンドを付いたくなる。となると、cd とか ls がどちらで実行されるかは状況による。ローカルでやる事を明示するために lcd とか llc とかが有る、というのも良いかなと。

社長：ふむ… となると、cd は仮想cd になるということですか。

基盤：ISOフォーマットのファイルみたいなやつですね。

経理：部屋の照明が全開になっているようですが。

社長：節電ブームは去りました。わたしはもともと照明が暗いのが苦手なんです。

基盤：部屋の温度も28度で安定してしまいました。30度と戦っていた日々が遠く感じられます。

開発：外気の温度が下がったのかも知れないですが、最近ちゃんとエアコンで冷えるようになりましたね。

基盤：といたしますか、エアコンがすごいガラガラ言ったのが、ずいぶん静かに回るようになりました。回しているうちにバリだかサビがとれたって感じですかねw

— 2020-0824 SatoxITS

```
/*
                                GShell version 0.2.0 // 2020-08-24 // SatoxITS
≡GShell   ≡GShell   ≡GS
```

## GShell // a General purpose Shell built on the top of Golang

It is a shell for myself, by myself, of myself. -SatoxITS(^-^)

| [NewWindow](#) | [Unfold](#) | [Fold](#) | [Stop](#) | [Close](#) | \*/ /\*

▶ Total Source of GShell

\*/ /\*

▶ Overview

\*/ /\*

▶ Go Source Code Index

\*/ //

▶ Go Source Code

//

▶ Consideration

/\*

▶ References



-> \*/ //

