

株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月29日 投稿者: SATOXITS

GShell 0.2.5 - 電子署名付きHTML

開発：さて今日は、昨日やりかけて終わった、JavaScript で Go のプログラムを生成する件で行こうかと思いますが。

社長：緊急動議。HTMLとかPNGに電子署名する話が着手されてないですね。GShellでやってみると良いのでは。この機能は、あればすぐに使いたいものです。gsh.go.htmlを公開する上でも。

開発：確かに電子署名は創立以来の本丸でした。

基盤：緊急動議。GShellでレンタルサーバにHTTPの口からログインしたいです。SSHがかかったるいし。やりたいのはGShellで既に出来ている遠隔exec、実際単なるtail -fだからです。

開発：リクエストと応答をHTTPにくるむ必要がありますが、そもそも現状、コマンドごとにコネクションを張る実装になってますし、GShellサーバをCGIとしても動かせばよいかなと思います。

社長：一回、我社のバックログを洗い出すというか、着手していることも含めて、整理する必要がありそうです。ちょっとたばこ飲み物を仕入れてきます。

* * *



社長：ひーは一… 帰りました。

基盤：遅かったですね。

社長：車で出て、大好評のとうもろこしとスイカを仕入れて来ました。

開発：カット版ですね。

社長：それが、小玉スイカの12分割みたいのが590円とかして、あんぐりしまして。しゃぐ。

開発：何がおきてるんでしょうね。しゃぐ。甘さイマイチ。

基盤：やはりスイカはでっかく切ってかぶりつかないとですね。しゃぐ。

社長：それに引き換えこのとうもろこし、結構太くて一本128円です。4本で512円。

基盤：ずいぶんキリの良い値段ですね。

開発：さっそくチンしましょう。90度で5分30秒くらいかな… レンジでGo!

レンジ：ゴー…

社長：そういえば、タバコに着火するのに使い捨てライターはもったいないと思うようになりました。携帯するわけではないし、台所のガス台の前で吸ってるわけですから。

基盤：ベランダで吸えばいいんじゃないですかね。気分もいいでしょうし。うるさい換気扇を回す必要もないですから。

開発：うーん、とうもろこしの甘い香りがして来ました。

レンジ：ぱびぺ、ぱびぺ、ぴぴぽぽぱー

開発：出来ました。どれどれ… あちっ、あちち。これしばらく触れないんですよ… やれ剥けました。

基盤：みっしり詰まってますね。どれどれ。はぐ。もぐもぐ。うん美味しい。

社長：もぐもぐ。前のよりスイート感が低い感じですが味が濃い気がしますね。色も濃い目。もぐもぐ。

開発：加熱時間も関係するかもですね。もぐもぐ。あー美味しかった。

基盤：これで128円はおトクですよ。

社長：100円ちょっとで売ってる低価格のカップ麺の凄さというのも改めて感じます。

* * *

開発：ふああ。。よく寝ました。

基盤：今日はまだ、とうもろこしを食べただけですね。

社長：手を動かすのだけが仕事ではありません。わたしは寝てる間に面白い事を思いつきました。

基盤：夢のお告げというやつですね。

開発：どのような。

社長：イベントレコーダーです。発生したイベントを時系列で記憶しておく。再生もできる。まあとりあえずキーロガーが良いです。たとえばGShell IMEでの入力操作を再生できたら見た目に面白いんじゃないかと思います。

開発：script コマンドは欲しいと思ってましたが、その特殊系のようなものですね。

基盤：キー入力レベルのヒストリというか。

社長：そうそれで、記録したものを外部表現できて伝達できて別の場所で再生できると良いと思うんです。できるだけコンパクトな表現で。

開発：文字列の中に `¥b` みたいな機能コードを表現できますが、あれに「指定時間遅延をする」という抽象的なコードを入れると良さそうですね。

社長：それか、ANSIIのエスケープシーケンスの拡張ですかね。

開発：コンパクトということにこだわらなければ、イベントを生成するコマンド列であっても良いですよ。タイミング情報付きのスクリプトというか。

基盤：タイミングは `usleep` コマンドで作ればいいんじゃないですかね。

社長：それもありですね。いろんな表現形式があって良い。

開発：実例をやってみせて学習させる式のプログラミングみたいのがありましたよね。なんちゃら `by example` とか `by demonstration` とか。

社長：GShell にもそういう感じの機能があると面白いですね。

開発：GShell+unfs3 で Automatorみたいなのは作りたいですね。

* * *

基盤：ああ、洗濯が終わってますね。おー、糸くずフィルターがーっ！て警告が。

開発：糸くずはまだしも、乾燥フィルターのほうがすごくないですか。座布団みたいに積もってる。

経理：このぶん、衣類が痩せてしまったかと思うと悲しい気分ですね。

基盤：掃除掃除。パカッと開いて捨てるだけ。よく出来てますねー。これって、前の機種だと結構面倒だったんです。

社長：内部にどっさりたまってしまって、針金で掻き出したり掃除機で吸い出したのを思い出しますね。

開発：しかしこれだけフィルターに堆積していても、乾燥が機能してたというのは大したものですね。

基盤：以前は毎回到近く乾燥フィルタはクリアしてたんですけどね。なんで忘れてたのかな…

経理：効率が下がって余計に電気を消費してたのでは無いでしょうか。

基盤：しかし衣類が痩せるとか電気を食うとか「洗濯物を干す」という作業のコストを考えればネグリジブルだと思いますね。季節によっては生乾きの匂いに嫌な思いをしていた時代が信じられません。

社長：そもそも「洗濯をする」こと自体が手作業で、大変な労力をかけてた時代もあったわけです。人類史的にはずっと。つい数十年前まで。

基盤：家電的にはテレビはもう見ないし、冷蔵庫もコンビニのおかげで無しでも過ごせるようになりましたが、洗濯機だけは無くってはならないですね。

社長：学生時代は共用の洗濯機かコインランドリーでしたけどね。これが非常に壊れやすく稼働率が低かった。一人の専有時間も長くて空きを待つのが大変。全自動洗濯機はまさに夢の家電でした。夢がかなった今も。

開発：槽内に糸くずが舞ってますね。

基盤：そういえばメンテナンスコマンドを2ヶ月ばかりやってませんでした。では自動クリーニングのフルコースで。うーん、洗濯機って面白いなー。

* * *

経理：7月ぶんの年貢を納めようと思って、PayB でピッとしようとしてスマホのカメラを向けたら、バーコードが無い事に気が付きました。

基盤：国税はPay-easyですね。

開発：なんでバーコードを付けないですかねー。

社長：出力フォームの変更に大幅なシステムの作り直しが必要だとか。ハイリスクなのにお役所にはノーリターン。

経理：それで、銀行のペイジーの画面で納付番号を入力したのですが、2回間違えてしまいました。12桁の数字なので、4桁ごとに区切りらないとわかりにくいです。

開発：ああ、あの「区切りのハイフンはいりません」というやつ。親切心なんですかね
W

基盤：別にあなたのためではなく自分のために入れたいだけなんですかって。

開発：システムを設計した人がたまたまブレインダメージだったんですかね。誰しもそういう日はあります。

基盤：逆に任意の場所に空白とかハイフンを入れても認識してくれるのもありますよね。

経理：英数字の全角だ半角だってのも困ります。

開発：いずれそもそも人間が入力しなくても良くなるでしょうけど。とりあえずバーコード、いずれ電子証明書があれば済むことです。

社長：しょぼい機械に人間が合わせなければならなかった時代。それを引きずった非道なUIで人が苦しめられていた時代。そんな時代もあったねと、笑い話に時が変えるよ、もうすぐね、みたいな。

基盤：そういえば社内でのログインにフェリカを使おうって買ったきりになってますね。

開発：あれは、リーダライタのドライバがmacOSに対応してないことがわかったので、計画が頓挫しました。

基盤：Parallelsが全く問題ないことがわかりましたから、いずれWindowsで読ませてMacに伝達すれば良いと思います。

社長：パスワード入力を求められたらフェリカに聞きに行くようなインフラができると良いですかね。

基盤：そしたらもうパスワードは全部、最大長のランダムな文字列にしてやりたいです。

開発：パスワードのリポジトリという意味では、別にフェリカでなくても良いようには思っています。物理的な接続・切断が簡単にできるメディアであれば。

基盤：USBメモリの抜き差しは結構面倒ですね。

開発：そこはスイッチ付きのUSBハブで。

社長：ラズパイをパスワードサーバにして、Etherを落とすというのもありかと。ああ、無線だと簡単じゃないかな。

開発：サーバのバイナリの中に暗号化して焼き込むのも良いかもですね。GShellをサーバにして。

基盤：ssh-agent 的な感じですかね。

開発：起動するのにパスワードを要求される自分専用のshellというか、実行ファイルってあって良いと思うんです。自分専用のbinに置く。OSでのユーザ認証とかに頼らない。

社長：ああそれは DeleGate でもやってました。GShellの場合、ブラウザからJavaScript でgsh.goを生成する時に埋め込めると便利だと思います。でバイナリをbuildしてソースは消す。

開発：ほとんどの場合パスワードはブラウザからの入力ですから、やはりクライアント側としてはブラウザに手を突っ込むのが一番手っ取り早いかなとは思っています。

基盤：Acrobatという難関がありますけどね。

* * *

社長：ということで今日は、計画を変更してイベントレコーダをやりたいです。

開発：起動するのにパスワードが必要なGShellっていうのも面白いとは思いますが。

社長：まあそれは後日。

基盤：クリーニング中で洗濯機の槽内が泡だらけになってるのが面白いですが、槽内の照明のLEDが8秒くらいで切れてしまうのが本当にムカつきます。せめてスイッチを押してる間は点灯してるとかすれば良いのに…

開発：それにしてもこれ、イベントをつかまえるというなら、ttyのreadじゃなくてキーのイベントを捕まえたいですね。そうすればもっと分解した単位で捕まえられると思うのですが。stty -echo -icanonなんて外部コマンドでやっているのも内心忸怩たる思いもあり。

社長：イベントを飛ばしてくれるTerminalアプリを作ると良いかもですね。

基盤：Windows版のGoでは使えなそう。Cygwinでイケるんですけどっけ？

社長：まあそれも後日。とりあえずこっちを先に進めましょう。

開発：えーと、キー入力の時刻とキーをイベントの可変長配列に追加していくと。こういうところがGoはCより簡単で便利なんですよね。安全だし。性能的にはどうか知りませんが。で、イベントをダンプする。time.Timeからナノ秒を取り出すには… … ああ、まんまNanosecond() っていうのがありますね。

開発：ということで、とりあえずこんな感じに。

```
gsh/0.2.5 (2020-08-29) SatoxITS(^-^)/
[あr]!1! echo Hello, 世界
Hello, sekai
[あr]!1! echo Hello, 世界
Hello, 世界
[あr]!3! hいstory
!1 echo Hello, sekai
!2 echo Hello, 世界
[あr]!4! ev
2  !1  [Aug 29 19:04:12.909987] 101 65 e
3  !1  [Aug 29 19:04:13.125962]  99 63 c
4  !1  [Aug 29 19:04:13.238198] 104 68 h
5  !1  [Aug 29 19:04:13.301372] 111 6F o
6  !1  [Aug 29 19:04:13.429710]  32 20 \s
8  !1  [Aug 29 19:04:14.925675]  72 48 H
9  !1  [Aug 29 19:04:15.126113] 101 65 e
10 !1  [Aug 29 19:04:15.829531] 108 6C l
11 !1  [Aug 29 19:04:16.008574] 108 6C l
12 !1  [Aug 29 19:04:16.261750] 111 6F o
13 !1  [Aug 29 19:04:16.693844]  44 2C ,
14 !1  [Aug 29 19:04:16.989987]  32 20 \s
15 !1  [Aug 29 19:04:17.181886] 115 73 s
16 !1  [Aug 29 19:04:17.261014] 101 65 e
17 !1  [Aug 29 19:04:17.366083] 107 6B k
18 !1  [Aug 29 19:04:17.525781]  97 61 a
20 !1  [Aug 29 19:04:17.596968] 105 69 i
22 !1  [Aug 29 19:04:18.861253]  92 5C \
23 !1  [Aug 29 19:04:20.141860] 106 6A j
25 !1  [Aug 29 19:04:21.460979]  10 0A \n
27 !2  [Aug 29 19:04:23.236546]  27 1B ESC
28 !2  [Aug 29 19:04:23.236552]  91 5B [
29 !2  [Aug 29 19:04:23.236556]  65 41 A
31 !2  [Aug 29 19:04:25.029416]  92 5C \
32 !2  [Aug 29 19:04:25.822087] 105 69 i
34 !2  [Aug 29 19:04:26.853554]  10 0A \n
41 !3  [Aug 29 19:04:33.214712] 104 68 h
42 !3  [Aug 29 19:04:33.278014] 105 69 i
43 !3  [Aug 29 19:04:33.614445] 115 73 s
44 !3  [Aug 29 19:04:33.942518] 116 74 t
45 !3  [Aug 29 19:04:34.054297] 111 6F o
46 !3  [Aug 29 19:04:34.134303] 114 72 r
47 !3  [Aug 29 19:04:34.278330] 121 79 y
48 !3  [Aug 29 19:04:34.853574]  10 0A \n
50 !4  [Aug 29 19:04:36.078134] 101 65 e
51 !4  [Aug 29 19:04:36.326477] 118 76 v
52 !4  [Aug 29 19:04:36.741658]  10 0A \n
[あr]!5! █
```

基盤：おー、キーボードだ。

社長：再生は！コマンドと同じようにやると良いかもですね。!1-2 とかすると、!1か

ら!2までの実行のタイミングどおりにキー入力を再生する。

開発：！コマンドはコマンドライン入力された後の文字列を直接再現実行しちゃいますが、キーの入力のタイミングまで再現するコマンドを作れば良いのでしょうか。うーん、つまりキー入力のイベント列を入力のカューに入れて、そこにあればttyじゃなくてそこから読む、みたいな実装が簡単かな。replay 1-2 みたいなコマンドで。

社長：何もなく時間が経過した、っていうのも一種のイベントとして入れとくと、実装が簡単そうですね。

開発：レベルかエッジかってとこですかね… じゃま、WAITという仮想的なキーが押されていたとします。

基盤：WAITというかIDLEじゃないですかね。

開発：じゃIDLEで。IDOLにしようかな。バイバイバイわたしのアイドル〜♪

社長：この、イベントにヒストリの番号とか、高次のコンテキストをセットにして記憶するのは何かに見えそうな気がします。

開発：それはコマンドヒストリ自体もそうだと思うんです。そのコマンドをどのワーキングディレクトリで実行したかとか。

基盤：以前と同じディレクトリで同じコマンドを実行したいって普通にありますよね。

* * *

社長：うー、無性にチューリップが聴きたくなりました。

基盤：iPhoneのiTunesに全部入ってますよね。

社長：めんどくさい。昔は手持ちのCDを全部 iMacのiTunesに読ませて、簡単にプレイリストくりくりで聞けて便利だったんですが。

開発：そういうえ何年前かにYouTubeでチューリップにハマってましたよね。残ってるのでは。

社長：久しぶりGoogleにログイン…高く評価した動画・・・ありました。だいぶ消されちゃってますね。河口湖のライブのパックがすごく良かったんですが…

開発：しかしひどい画質ですね。Retinaが泣いている。

基盤：でもiMacスピーカーは大活躍。ちゃんとどっしりしっとりした音がでますね。アンプの関係も？

社長：フィリップス君のスピーカーは悲惨でした。

開発：まあオーディオは当時より今のほうが退化してるかもですね。

社長：YouTubeで色々探すと、姫野さんの風貌の変遷が面白いんですw

開発：あー、山本潤子幸せになるための続いでましたね。しみ入るナ〜…

基盤：プリキュア夢は未来への道来ましたw

社長：くじけない決して逃げない〜♪ か…そう言えばつらい時代もあったよね。

開発：これは前に聴いてたのと詰合せが全然違いますね。

社長：まあどうせフローラくらいしか覚えてないですけど。でもこの作詞者天才かなと思います。

基盤：これBGMとしてプレイリストローテーションしときませんか。なんか元気出ますし。

社長：いろんなのを高く評価しちゃってますからねー。プレイリストって複数作れるんですかね？

開発：自前で作ればいいんじゃないですかね。URLを開くコマンドを外からブラウザに送れば良いと思いますが…あるいはJavaScript にリストをわたしてwindowを順次開かせるとか。プレイの終了をどうやって検出するか… まあ window のURLが変わったのを見ればいいんですかね。

基盤：別のアカウントでやれば簡単なのでは。Tube-Uでやりましょうか。

社長：いや、sato@its でいいです。これは正式な福利厚生の一環です。

基盤：前に聴いてたプリキュアの詰め合わせってこれですよね。

社長：ああそれです。

基盤：ではとりあえず、チューリップと山本潤子と北川理恵ローテーション、セットしました。

開発：竹田の子守唄もしみるナー…

基盤：続けて聴いてるとあっという間に空気になっちゃうでしょうけどね…

開発：久石さんのStand Alone と幸せになるためにはクルマで6年間に1000回以上聴いたと思いますが、飽きませんでしたね。

社長：iTunesでなくて良いので、手持ちのCDを読み込んでプレイリスト作って再生できると良いのですが。

基盤：また300枚読み込むんですか…

社長：青春の思い出ですからね。あれだけは引っ越しの時に捨てなくてよかったと思います。

開発：あとこれ、広告の自動スキップextension作りたいですね。

社長：画面をパターンマッチしてマウスクリックイベント出すだけですよね。ブラウザ内蔵でなくても良いのでは。汎用の「広告スキップロボット」で。

基盤：ここにもReCAPTCHA技術が採用されてるかもですがw

* * *

開発：では仕事を再開します。

社長：間に挟まってるカブトムシ♪がなかなか良いですね。

開発：BGMのある生活って、久しぶりですね。それで、IDLE時間を表示するようにしたら面白い事が分かりました。こんな感じです。

```
iMac% gsh
gsh/0.2.5 (2020-08-29) SatoxITS(^-^)/
[あ r]!1! echo 世界
sekai
[あ r]!2! ev
0  !1  [Aug 29 22:00:17.252409] 101 65 e      0.000ms
1  !1  [Aug 29 22:00:17.252409] 207 CF IDLE 279.751ms
2  !1  [Aug 29 22:00:17.532205]  99 63 c      0.000ms
3  !1  [Aug 29 22:00:17.532205] 207 CF IDLE  95.436ms
4  !1  [Aug 29 22:00:17.627647] 104 68 h      0.000ms
5  !1  [Aug 29 22:00:17.627647] 207 CF IDLE  72.510ms
6  !1  [Aug 29 22:00:17.700160] 111 6F o      0.000ms
7  !1  [Aug 29 22:00:17.700160] 207 CF IDLE 136.186ms
8  !1  [Aug 29 22:00:17.836350]  32 20 \s     0.000ms
9  !1  [Aug 29 22:00:17.836350] 207 CF IDLE 463.087ms
10 !1  [Aug 29 22:00:18.299451] 115 73 s      0.000ms
11 !1  [Aug 29 22:00:18.299451] 207 CF IDLE  81.043ms
12 !1  [Aug 29 22:00:18.380497] 101 65 e      0.000ms
13 !1  [Aug 29 22:00:18.380497] 207 CF IDLE 159.958ms
14 !1  [Aug 29 22:00:18.540461] 107 6B k      0.000ms
15 !1  [Aug 29 22:00:18.540461] 207 CF IDLE 151.865ms
16 !1  [Aug 29 22:00:18.692332]  97 61 a      0.000ms
17 !1  [Aug 29 22:00:18.692332] 207 CF IDLE 128.307ms
18 !1  [Aug 29 22:00:18.820643] 105 69 i      0.000ms
19 !1  [Aug 29 22:00:18.820643] 207 CF IDLE 1159.218ms
20 !1  [Aug 29 22:00:19.979896]  92 5C \     0.000ms
21 !1  [Aug 29 22:00:19.979896] 207 CF IDLE  336.684ms
22 !1  [Aug 29 22:00:20.316591] 106 6A j      0.000ms
23 !1  [Aug 29 22:00:20.316591] 207 CF IDLE 1374.587ms
24 !1  [Aug 29 22:00:21.691219]  10 0A \n     0.000ms
25 !1  [Aug 29 22:00:21.691219] 207 CF IDLE 2841.187ms
26 !2  [Aug 29 22:00:24.532500] 101 65 e      0.000ms
27 !2  [Aug 29 22:00:24.532500] 207 CF IDLE  200.117ms
28 !2  [Aug 29 22:00:24.732623] 118 76 v      0.000ms
29 !2  [Aug 29 22:00:24.732623] 207 CF IDLE  159.358ms
30 !2  [Aug 29 22:00:24.891987]  10 0A \n     0.000ms
[あ r]!3! █
```

開発：ざっと見て、連続して入力している時のキーとキーの間隔は100ms程度です。つまり、わたしのキー入力は最速で1秒に10文字程度。ただし、ホームポジションから離れたキーを押す前とかは空走時間が長い。

社長：どのくらい再現性があるか調べると面白そうですね。昔、キー入力の癖で個人を

特定する研究をしてる人がいましたが、こういうのが基礎データなんでしょうね。

開発：で、このタイミング情報を何か、IMEにおける変換の推測とか、ユーザの意図の推測とかに使えると面白いなと思います。

社長：ユーザが意図的に、打鍵のリズムで別の単語を選ぶとか言うのもいいかもですね。

基盤：少し長押ししたら長音記号が出るとか便利だと思います。

開発：あと、エスケープシーケンスの生成速度とか。上向き矢印は ESC [A なのですが、一瞬で生成されてると思いますから文字間の時間間隔はナノ秒単位ではないかと。表示されている数マイクロ秒は、GShellによるキーの処理時間と思われます。

```
iMac% gsh
gsh/0.2.5 (2020-08-29) SatoxITS(^-^)/
!! date
Sat Aug 29 22:14:34 JST 2020
!! date
Sat Aug 29 22:14:36 JST 2020
!3! ev
0  !1  [Aug 29 22:14:34.213057] 100 64 d      0.000ms
1  !1  [Aug 29 22:14:34.213057] 207 CF IDLE  71.572ms
2  !1  [Aug 29 22:14:34.284638]  97 61 a      0.000ms
3  !1  [Aug 29 22:14:34.284638] 207 CF IDLE  176.078ms
4  !1  [Aug 29 22:14:34.460722] 116 74 t      0.000ms
5  !1  [Aug 29 22:14:34.460722] 207 CF IDLE  63.844ms
6  !1  [Aug 29 22:14:34.524569] 101 65 e      0.000ms
7  !1  [Aug 29 22:14:34.524569] 207 CF IDLE  471.178ms
8  !1  [Aug 29 22:14:34.995761]  10 0A \n      0.000ms
9  !1  [Aug 29 22:14:34.995761] 207 CF IDLE  688.420ms
10 !2  [Aug 29 22:14:35.684207]  27 1B ESC     0.000ms
11 !2  [Aug 29 22:14:35.684207] 207 CF IDLE   0.006ms
12 !2  [Aug 29 22:14:35.684213]  91 5B [      0.000ms
13 !2  [Aug 29 22:14:35.684213] 207 CF IDLE   0.003ms
14 !2  [Aug 29 22:14:35.684217]  65 41 A      0.000ms
15 !2  [Aug 29 22:14:35.684217] 207 CF IDLE  775.262ms
16 !2  [Aug 29 22:14:36.459503]  10 0A \n      0.000ms
17 !2  [Aug 29 22:14:36.459503] 207 CF IDLE 1649.238ms
18 !3  [Aug 29 22:14:38.108809] 101 65 e      0.000ms
19 !3  [Aug 29 22:14:38.108809] 207 CF IDLE  215.279ms
20 !3  [Aug 29 22:14:38.324095] 118 76 v      0.000ms
21 !3  [Aug 29 22:14:38.324095] 207 CF IDLE  119.873ms
22 !3  [Aug 29 22:14:38.443973]  10 0A \n      0.000ms
!4! █
```

社長：これはコンピュータの負荷状況をざっくり知るのに使えるかも知れないですね。

基盤：定期的に行っている処理があれば、その時間のズレとか。

開発：いや、それはあるんですよ。1秒毎にselectで入力をタイムアウトして、行を再表示してるんでw。うざいから表示を削りましたが、表示するとこんな感じ。

```
9  !1  [Aug 29 22:22:33.296643] 207 CF IDLE 1014.253ms
10 !2  [Aug 29 22:22:34.310948] 206 CE CE 0.000ms
11 !2  [Aug 29 22:22:34.310948] 207 CF IDLE 1006.818ms
12 !2  [Aug 29 22:22:35.317798] 206 CE CE 0.000ms
13 !2  [Aug 29 22:22:35.317798] 207 CF IDLE 1001.837ms
14 !2  [Aug 29 22:22:36.319672] 206 CE CE 0.000ms
15 !2  [Aug 29 22:22:36.319672] 207 CF IDLE 1007.586ms
16 !2  [Aug 29 22:22:37.327289] 206 CE CE 0.000ms
17 !2  [Aug 29 22:22:37.327289] 207 CF IDLE 1000.677ms
18 !2  [Aug 29 22:22:38.327997] 206 CE CE 0.000ms
```

開発：これは以前「そうだ、スケジューラを作ろう」の時にも測ったと思いますが、今回は暇人のiMacなので、スケジュールの時間のズレというか、selectがタイムアウトしてくる時間の遅延は少ないです。長くとも15ms程度。

基盤：さっきからYouTube流しているのだから、あれは結構スケジューリングを乱してるんじゃないでしょうか？止めてみましょうか。

```
31 !4  [Aug 29 22:28:50.328286] 207 CF IDLE 1004.591ms
32 !4  [Aug 29 22:28:51.332908] 206 CE CE 0.000ms
33 !4  [Aug 29 22:28:51.332908] 207 CF IDLE 1004.791ms
34 !4  [Aug 29 22:28:52.337729] 206 CE CE 0.000ms
35 !4  [Aug 29 22:28:52.337729] 207 CF IDLE 1002.947ms
36 !4  [Aug 29 22:28:53.340707] 206 CE CE 0.000ms
37 !4  [Aug 29 22:28:53.340707] 207 CF IDLE 1004.731ms
38 !4  [Aug 29 22:28:54.345470] 206 CE CE 0.000ms
39 !4  [Aug 29 22:28:54.345470] 207 CF IDLE 1004.392ms
40 !4  [Aug 29 22:28:55.349893] 206 CE CE 0.000ms
41 !4  [Aug 29 22:28:55.349893] 207 CF IDLE 1004.353ms
42 !4  [Aug 29 22:28:56.354277] 206 CE CE 0.000ms
43 !4  [Aug 29 22:28:56.354277] 207 CF IDLE 1001.648ms
44 !4  [Aug 29 22:28:57.355956] 206 CE CE 0.000ms
45 !4  [Aug 29 22:28:57.355956] 207 CF IDLE 1000.323ms
46 !4  [Aug 29 22:28:58.356310] 206 CE CE 0.000ms
47 !4  [Aug 29 22:28:58.356310] 207 CF IDLE 1004.322ms
48 !4  [Aug 29 22:28:59.360663] 206 CE CE 0.000ms
```

社長：良くなったのかもしれませんが、4msズレたら組み込み系のリアルタイム用途では使えないですね。40usでも厳しい。4usでもダメな場合もある。

開発：まあ macOS は組み込み用じゃないですからねw

基盤：そういえば Parallels も動いてますしね…

開発：ということで、イベントのログを取るのはとりあえずこんなところかと。

社長：あとはいよいよ、再生ですね。ワクワク。

開発：私は少し飽きてきてしまったんですが、プリキュアが流れるとなんだかシャキッとしますw

* * *

開発：さて実装の形として、伝統的な形式でコマンド行入力の下請けになっているために、作りがちょっといびつになります。入力を差し戻すという。GUIのようにイベント・ドリブンで出来れば、もっと素直にできるんですが。

社長：Goルーチンでキーボード入力のフリをするタスクを作ればいいんじゃないですかね。タイミング調整しながらキー入力をpipeに送る。IMEは、tty とpipeをselectする。

開発：それは美しいかも知れない…

基盤：自動入力に手動入力を介入させるのは面白そうです。

開発：確定しました。それでいきます。そもそもキー入力だけでなく、ファイルやソケットの入力に全部これを仕掛けます。すると、割り込み処理無しで、ポーリングだけで、単一タスクでイベント処理ができる。排他制御とかしなくて済むし、一つのコンテキストで実行できるのも良いと思います。コンテキストを分けたければその時は分ければ良い。即時処理が必要な処理なら、ここを通らずに普通にGoルーチンを分ければ良い。

* * *

開発：前進しました。

社長：完成ですか。

開発：一歩手前、寸止め状態みたいな。いや、Go の syscall が Poll を提供してなかったり、socketpair の仕様がドキュメントと違ったりでちょっと嫌な思いをしました。それはともかく、こういうことになりました。

```
iMac% gsh
gsh/0.2.5 (2020-08-29) SatoxITS(^-^)/
!1! ungets date
!2! date
Sat Aug 29 23:57:59 JST 2020
!3! ev
#0 !1 [Aug 29 23:57:56.338507] 117 75 u 0.000ms
#1 !1 [Aug 29 23:57:56.338507] 207 CF IDLE 223.979ms
#2 !1 [Aug 29 23:57:56.562512] 110 6E n 0.000ms
#3 !1 [Aug 29 23:57:56.562512] 207 CF IDLE 255.678ms
#4 !1 [Aug 29 23:57:56.818198] 103 67 g 0.000ms
#5 !1 [Aug 29 23:57:56.818198] 207 CF IDLE 248.346ms
#6 !1 [Aug 29 23:57:57.066552] 101 65 e 0.000ms
#7 !1 [Aug 29 23:57:57.066552] 207 CF IDLE 232.095ms
#8 !1 [Aug 29 23:57:57.298654] 116 74 t 0.000ms
#9 !1 [Aug 29 23:57:57.298654] 207 CF IDLE 272.093ms
#10 !1 [Aug 29 23:57:57.570756] 115 73 s 0.000ms
#11 !1 [Aug 29 23:57:57.570756] 207 CF IDLE 503.985ms
#12 !1 [Aug 29 23:57:58.074757] 32 20 \s 0.000ms
#13 !1 [Aug 29 23:57:58.074757] 207 CF IDLE 264.377ms
#14 !1 [Aug 29 23:57:58.339142] 100 64 d 0.000ms
#15 !1 [Aug 29 23:57:58.339142] 207 CF IDLE 95.934ms
#16 !1 [Aug 29 23:57:58.435080] 97 61 a 0.000ms
#17 !1 [Aug 29 23:57:58.435080] 207 CF IDLE 175.351ms
#18 !1 [Aug 29 23:57:58.610437] 116 74 t 0.000ms
#19 !1 [Aug 29 23:57:58.610437] 207 CF IDLE 72.271ms
#20 !1 [Aug 29 23:57:58.682711] 101 65 e 0.000ms
#21 !1 [Aug 29 23:57:58.682711] 207 CF IDLE 807.365ms
#22 !1 [Aug 29 23:57:59.490101] 10 0A \n 0.000ms
#23 !1 [Aug 29 23:57:59.490101] 207 CF IDLE 3.884ms
#24 !2 [Aug 29 23:57:59.493987] 100 64 d 0.000ms
#25 !2 [Aug 29 23:57:59.493987] 207 CF IDLE 0.008ms
#26 !2 [Aug 29 23:57:59.493995] 97 61 a 0.000ms
#27 !2 [Aug 29 23:57:59.493995] 207 CF IDLE 0.006ms
#28 !2 [Aug 29 23:57:59.494002] 116 74 t 0.000ms
#29 !2 [Aug 29 23:57:59.494002] 207 CF IDLE 0.008ms
#30 !2 [Aug 29 23:57:59.494010] 101 65 e 0.000ms
#31 !2 [Aug 29 23:57:59.494010] 207 CF IDLE 0.009ms
#32 !2 [Aug 29 23:57:59.494019] 10 0A \n 0.000ms
#33 !2 [Aug 29 23:57:59.494019] 207 CF IDLE 916.788ms
#34 !3 [Aug 29 23:58:00.410844] 101 65 e 0.000ms
#35 !3 [Aug 29 23:58:00.410844] 207 CF IDLE 247.986ms
#36 !3 [Aug 29 23:58:00.658837] 118 76 v 0.000ms
#37 !3 [Aug 29 23:58:00.658837] 207 CF IDLE 191.182ms
#38 !3 [Aug 29 23:58:00.850025] 10 0A \n 0.000ms
!4!
```

開発：ungets で date という文字列をキー入力として送っています。

基盤：ungetsじゃないのにw

開発：それで、#14 - #22 までの date の入力の手入力ですから、キーの間が100ms程度空いている。ところが、#24 - #32 でキー入力だと思って受け取ったキーは、数マイクロ間隔で届くというわけです。

基盤：驚速！

社長：驚速がGoogle IMEのデフォルトに無いのは悲しいですね。自分の辞書に追加っと。

開発：あれはあの時期、画期的なネーミングとインパクトのあるCMでしたよね。

社長：もう商品としては無いんですかね？

基盤：検索したら、ちゃんと販売してます。

開発：でも私らが言ってる驚速というのは、キーボード入力が速くなる、たしか練習ソフトだったような気がするんですよ。

* * *

開発：どうもブログでヨタ話を書いている時間が長くて、作業が進みませんねw

基盤：プリキュア再生してで気分を再生しましょうw

開発：ふう、出来ました。これでいいんじゃないですかね。遅延入りで再生です。

```
iMac% gsh
gsh/0.2.5 (2020-08-29) SatoxITS(^-^)/
!1! date
Sun Aug 30 01:01:07 JST 2020
!2! replay
--Ir-- Replay(!1)
  0.000ms #0  !1  [Aug 30 01:01:06.496044] 100 64 d      0.000ms
  0.111ms #1  !1  [Aug 30 01:01:06.496044] 207 CF IDLE   96.230ms
 99.975ms #2  !1  [Aug 30 01:01:06.592293]  97 61 a      0.000ms
100.020ms #3  !1  [Aug 30 01:01:06.592293] 207 CF IDLE  143.841ms
251.161ms #4  !1  [Aug 30 01:01:06.736148] 116 74 t      0.000ms
251.217ms #5  !1  [Aug 30 01:01:06.736148] 207 CF IDLE   63.998ms
324.263ms #6  !1  [Aug 30 01:01:06.800153] 101 65 e      0.000ms
324.338ms #7  !1  [Aug 30 01:01:06.800153] 207 CF IDLE  231.602ms
558.634ms #8  !1  [Aug 30 01:01:07.031776]  10 0A \n      0.000ms
!3! date
Sun Aug 30 01:01:09 JST 2020
!4! █
```

基盤：機械で再生すると、人間のもっさり感を感じますねw

社長：いよいよ GShell IMEで行きますか。

開発：あれ、BGM止まった。盛り下げてくれますね。



基盤：やはりYouTubeプレイリスト再生を自作するのが良いかも知れませんね。

開発：それでは再びプリキュアをBGMに。行きます。

社長：その前にいっぷく…

* * *

開発：では行きます…

基盤：できた！

社長：できましたね。あれ？でもWordPressに動画がアップロードができないです…



基盤：セッションが切れてるような切れてないような…

社長：一度入り直しますか… これはどうかな？



基盤：なんかダメですね。昔の QuickTime は普通にアップロードできるようですが…

社長：サイズが小さすぎるとか？

開発：ああ、Safari では再生できますよ。Vivaldi が死んでるとか。

基盤：Vivaldi 以外のブラウザなら再生できますね。ていうか、iMacのVivaldiならOK。

社長：そうですか、立ち上げ直せば大丈夫なのかな。Vivaldi だけ動画の再生の色が変なことがあるんですが、あれも酷使し過ぎですかね。

社長：それはともかく、昼間見た「夢」が実現しました。

開発：夢は未来への道ですね。明るい未来が見えた気がしたw

基盤：そろそろMacMiniをやめろというお告げかも知れぬ。もう購入して4ヶ月、十分使いました。RAMを増やしてあげればよいのかもしれませんが、保証外になっちゃいますし。

開発：Appleに保証サービスで直してもらった事は無いですけどね。ああ、iPhoneは交換しましたけど。

社長：それはそうと、このIMEの「世界」専用辞書は、専用だけあって「世界」を出すには最高ですね。今まで生きてきて一番、使いやすいです。



```
iMac%  
iMac%  
iMac%  
iMac%  
iMac%  
iMac%  
iMac%  
iMac%  
iMac% gsh  
gsh/0.2.5 (2020-08-29) SatoxITS (^-^)/  
!! |
```

社長：それでこの例で気づいたのですが、replayする時にIMEの変換モードが再現されていないので、表示が実際のリプレイにならないということです。なので、上の例ではreplayの前に手動で再現しています。

開発：さすがにそこは自動で再現するようにします。

社長：ただ、どこまで環境を回復するかは難しいところですね。あらゆる時点の環境を保存できるわけでは無いし、そもそも回復できない外部環境もありますから。

開発：まあ、1コマンドごとにGShellのコンテキストをダンプしちゃうとか、可能だとは思いますが。再実行した結果を現在の環境に取り込もうとか、そもそも現在の環境で過去の操作を再実行しようという場合もあるでしょうし。

社長：GShellの環境まるごとダンプは良いのではないですかね。ホストがいきなり落ち

た場合にも、直近の状態から再開できそう。

開発：プロセスレベルのハイバネーションも出来ますね。

基盤：目覚めたときには浦島太郎だったとかw

開発：複雑な情報を含んだ履歴ファイルとか作るの面倒だから、ずっと内部表現で維持しようかなw

社長：今日は面白い機能ができました。ちょっとした機能ですが、当社の one giant leap だったのかも知れません。

開発：あとと思ったんですが、マウスの動作もこれ式でいくつかキャプチャしといて、自動でreCAPTCHAボタン押させてやれば、「私はロボットではありません」で破れるんじゃないですかね。

— 2020-0829 SatoxITS

<http-im3-gsh-gsh-0.2.5.go>

ダウンロード

/*

GShell version 0.2.5 // 2020-08-29 // 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(^-^)

0 | | Fork | Stop | Unfold | */ /*

▼ Statement

Fun to create a shell

For a programmer, it must be far easy and fun to create his own simple shell rightly fitting to his favor and necessities, than learning existing shells with complex full features that he never use. I, as one of programmers, am writing this tiny shell for my own real needs, totally from scratch, with fun.

For a programmer, it is fun to learn new computer languages. For long years before writing this software, I had been specialized to C and early HTML2 :-). Now writing this software, I'm learning Go language, HTML5, JavaScript and CSS on demand as a novice of these, with fun.

This single file "gsh.go", that is executable by Go, contains all of the code written in Go. Also it can be displayed as "gsh.go.html" by browsers. It is a standalone HTML file that works as the viewer of the code of itself, and as the "home page" of this software.

Because this HTML file is a Go program, you may run it as a real shell program on your computer. But you must be aware that this program is written under situation like above. Needless to say, there is no warranty for this program in any means.

Aug 2020, SatoxITS (sato@its-more.jp)

/ /

▼ Index

Documents

[Command summary](#)

Go lang part

Package structures

[import](#)

[struct](#)

Main functions

<u>str-expansion</u>	// macro processor
<u>finder</u>	// builtin find + du
<u>grep</u>	// builtin grep + wc + cksum + ...
<u>plugin</u>	// plugin commands
<u>system</u>	// external commands
<u>builtin</u>	// builtin commands

```
network           // socket handler
remote-sh        // remote shell
redirect         // StdIn/Out redireciton
history          // command history
rusage           // resouce usage
encode           // encode / decode
IME              // command line IME
getline         // line editor
scanf           // string decomposer
interpreter     // command interpreter
main
```

JavaScript part

[Source](#)

[Builtin data](#)

CSS part

[Source](#)

References

[Internal](#)

[External](#)

Whole parts

[Source](#)

[Download](#)

[Dump](#)

*/ //

▶ [Go Source](#)

//

▶ [Considerations](#)

// /*

▶ [References](#)

/ /

▶ [Raw Source](#)

/ /



→ *///