

株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月7日 投稿者: SATOXITS

そうだ、Shellを作るう

開発：プログラマとして生まれたからにはやはり、一度は自分のshellを作ってみたいものだと思うわけです。

基盤：どういう生まれ方をしたんですか。

開発：私がCプログラマになったころにはまだ、Unix V6でしたから、V6のshellしかなかったんです。ヒストリを参照するのにいちいち !なんとかを入れないといけない。面倒なのでカーソルで昔のヒストリに戻れるようなのを作りましたよ。それとか、プログラムのPATH同様のデータのPATHを定義して、参照したデータファイルをそのPATHで探索して開くようなこともしました。ファイル名を入れるとそのディレクトリにジャンプするとか。

開発：最近zshやsshに、大いに刺激されました。コマンドラインシェルは不滅だなという気もしました。ですが、コーディングを見るとまあ、昔ながらのCなわけです。それはそれで好きなわけですが、今はもうそういう時代では無いのではないかなという気もするわけです。

開発：shell の重要な役割は、実行するプログラムの外部環境を整えて提供することだと思うんです。環境としてはまず、ファイルディスクリプタで与えたり、動的ライブラリとして与えたりするもの。あるいはファイルシステムの可視性を与えるかも知れない。実行のタイミングを与えるかも知れない。

開発：Unixがくれた重要な概念として標準入出力というのがあるわけです。これはコマンドのパイプラインを筆頭として、プログラムを部品化して再利用性を高めるのに大い

に有益でした。そのうえ、入出力をどこにつなぐかというコーディングを各プログラムからなくすことができた。まあ必ずしもそうはなっていませんが。

開発：例えば、shellが開いてファイルディスクリプタとしてプログラムに与えて上げられるのはふつう、ローカルファイルかパイプだけなわけです。ですが、今やもちろん、リソースもさまざまだし、つなぐのだってパイプとは限らない。

開発：あるいは、コマンド引数というものを各プログラムが解釈するからてんでばらばらになったりするわけですが、shellが標準的な引数フォーマットを解釈して前処理して、プログラムは解釈結果だけを利用するようにすれば、もっと統一性がとれるし、プログラムのコーディングも省けるはずですよ。

社長：で実装の方針は？

開発：今から作るならやはりGoベースだと思うんです。豊富なパッケージを組み込みコマンドとして使えるし。

基盤：そういうのって既にあるんじゃないですかね。

開発：いや、既にあるとか関係ないです。これは、自分なりのshellを作ってみたいという欲求に突き動かされているものです。そして、実際自分なりに作って改良して行くと、既存のものには無かった何かが、必ず生まれるものです。無知や誤解はオリジナリティの父になります。だからこの件については、既存のものがあるか調べ物はしません。

開発：第一次トライアルは3日程度のスプリントで考えたいと思います。

社長：何という名前にしますか。

開発：まあとりあえず安直に、gsh ですかね。

社長：承認。

* * *

開発：というわけで、第1版ができました。こんなかんじです。

```
MacMini% go run gsh.go
0> -ver
gsh/0.0.1 (2020-0807a)
1> eval 0x1000
4096
2> eval 1 + 2 + 3
6
3> eval "abc" + "def"
"abcdef"
4> which date
/bin/date
5> date
Fri Aug  7 13:04:49 JST 2020
-- (0.003045727s)
6> sleep 1
-- (1.010282650s)
7> exit
```

開発：機能的には、golang の eval が出来るようにした点と、コマンドを実行するごとに実行時間を自動表示する点が特長です。実装上は、fork して exec する方法を調べるのに少し手間取りました。現在のコードはこんな感じ。

社長：内部コマンドも実行時間を表示してくれると良いですね。

基盤：システムコールナマ呼び出しですか…

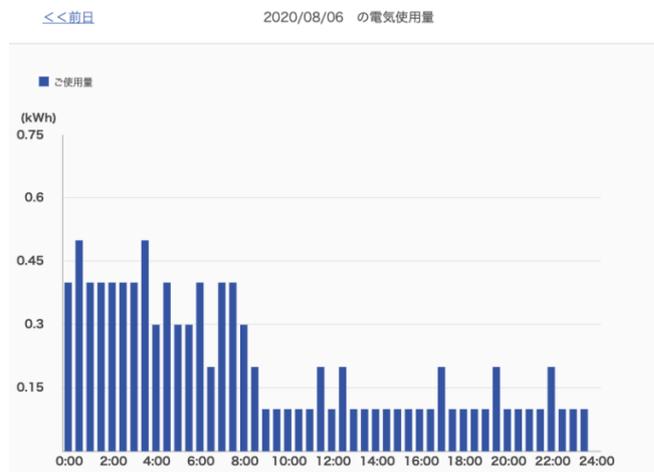
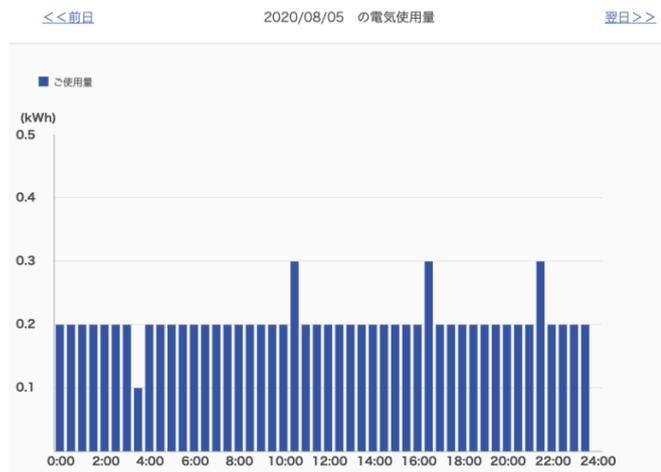
開発：Goの皮をかぶっていると自由が効かないんですよ。データ構造的には Goの機能を活用させてもらいますが、外部とのやりとりというか結線は、ナマでやろうと思えます。

社長：良い感じですね。ちょっと食事に行ってきます。

* * *

社長：帰りました。またえびすの小瓶のラーメンやでした。

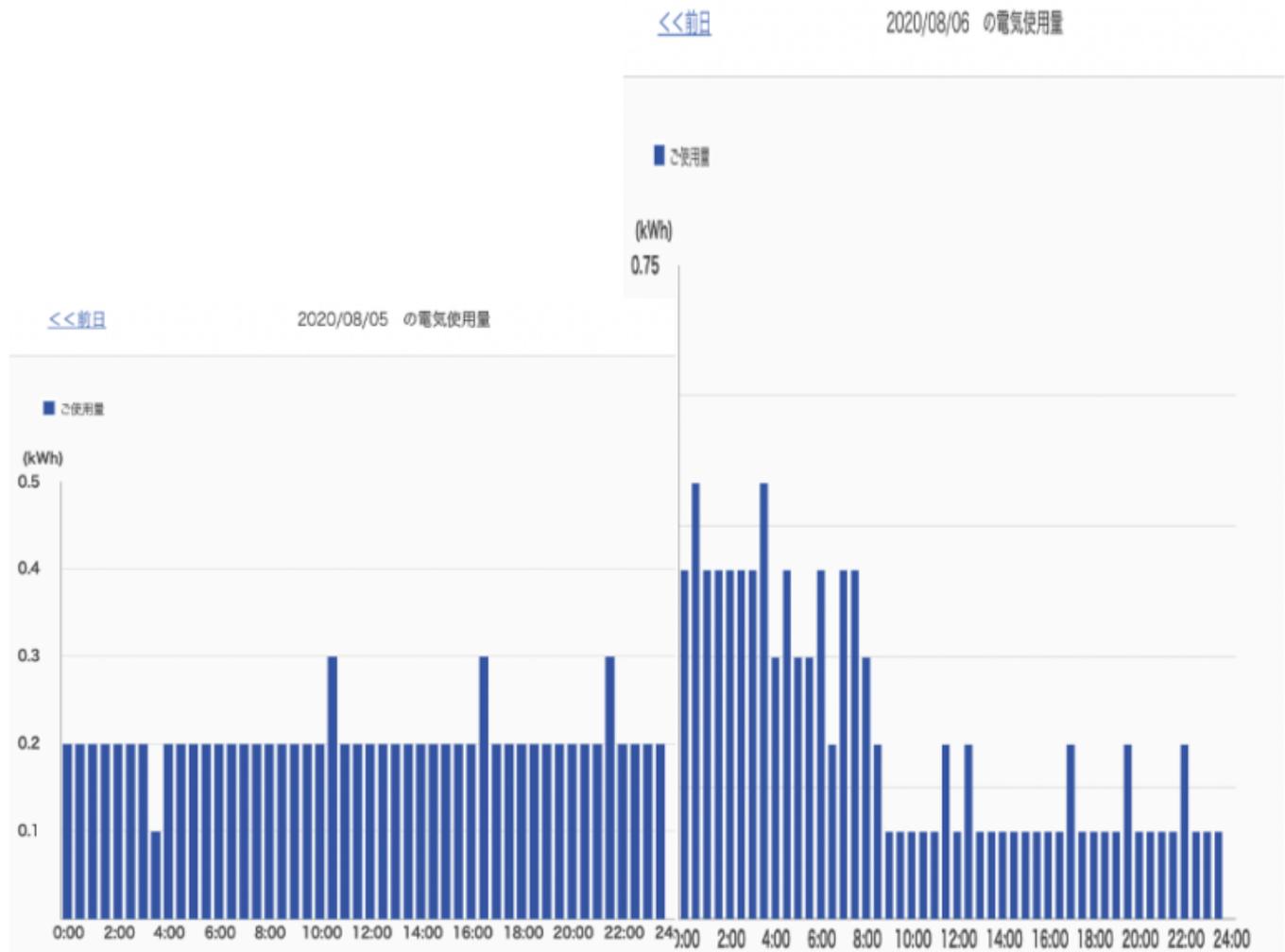
経理：大変な事になりました。これを見て下さい。



経理：左が一昨日の電気使用量。ほぼ200Wベたです。右が昨日。零時頃にエアコンをつけて8時に切り、ついでに各種の無駄機器の電源を切りました。

社長：劇的ビフォー・アフター。

基盤：これ、スケールが変わっちゃってますね。そろえるところなります。



開発：つまり一昨日まで200Wだったのが、昨日深夜から朝まで400Wになり、その後100Wになった。

基盤：ただ不思議なのは、UPSの電力計は常時150Wを超えていますから、100W刻みなのでふこでんこ計では200と100が交互になりそうに思うんですが、このグラフではほぼ100Wで推移していることです。



社長：まあ、これが逆だったら問題ですけどね。しかしうちって100Wで維持できちゃう会社なんだなあ。

経理：こうなるともう、基本料金が気になってきますね。契約の容量を下げましょうか？

社長：まあ話としては面白いですが… いざという時のこともあるし。他の方面に残っている巨悪から片付けたほうが良さそうです。

開発：あれ？でんこ計の縦軸ってkWhですね。一つの棒は30分毎。つまり30分に0.1kWhでことで、1時間平均にすると0.2kWh、つまり平均消費電力200Wということなのでは？

社長：あ。なるほど… それなら計算が合います。

基盤：なんにしても一昨日まで平均400W使ってたのを、昨日から200Wに削減できたと。照明を落としたり、無駄に電源に電源ONのまま待機していたテレビやオーディオを切った効果でしょうか。何ヶ月、というか何年も無駄に200W食い続けてたなんて、恐ろしすぎる…

基盤：ああ、それはそうと、このPoerMeter、結局 macOS版はしょっちゅ接続不能になってしまうので、Lenovoで動かすことにしました。いまのところ朝からずっと快調です。

* * *

開発：ふあ。あ。ああ。あー、よく寝た。

社長：よく規則正しい生活をとて言いますが、寝たい時に寝て食べたい時に食べる。どっちが身体にいいんでしょうね？

基盤：まあ無理に寝る必要が無いので、不眠症というものはこの生活スタイルには存在しないですね。

* * *

開発：第2版ができました。こんな感じです。

```
MacMini% go run gsh.go
refF opened
0> -ver
gsh/0.0.2 (2020-0807b)
--(0.000012340s)
1> echo gsh.pid
32689
--(0.000033458s)
2> nop
--(0.000001957s)
3> repeat 1000000 nop
--(0.008666423s)
4> repeat 3 sleep 500ms date
Sleep 500ms ns
Fri Aug 7 20:55:44 JST 2020
Sleep 500ms ns
Fri Aug 7 20:55:45 JST 2020
Sleep 500ms ns
Fri Aug 7 20:55:45 JST 2020
--(1.518794091s)
5> hello.go
-- /usr/local/go/bin/go {/usr/local/go/bin/go run ./hello.go}
Hello gsh!
--(0.461388411s)
6> hello
-- /usr/local/go/bin/go {/usr/local/go/bin/go run ./hello.go}
Hello gsh!
--(0.261602836s)
7> ls -l gsh.go
-rw-r--r-- 1 ysato staff 6160 Aug 7 20:45 gsh.go
--(0.009045461s)
8> -o xxx ls -l gsh.go
-- Opened [7] xxx
--(0.004344072s)
9> -i xxx cat -n
-- Opened [8] xxx
    1 -rw-r--r-- 1 ysato staff 6160 Aug 7 20:45 gsh.go
--(0.007131532s)
10> quit
--(0.000002197s)
```

開発：今回は repeat というコマンドを入れました (4>)。nop というコマンドを 1,000,000 回実行して所要8.6ms、つまり一回あたり8.6ns であることがわかります。

社長：さすが3GHzのCPUですね。私がかないだまで触ってた100MHzのARMとはえらい違いです(^-^)

開発：それと、lets.go というファイルを “go run lets.go” なしで、単に lets で実行できるようにしました (6>)。

開発：あと、外部コマンドの入力ファイルや出力ファイルを -i ifile -o ofile command のように指定できるようにしました。まあ通常のshellなら、command < ifile > ofile などところですか。

開発：全般的に、とにかくシンプルに左から実行して行くという型式です。

社長：まあ中置記法は実は人間にも優しくないという事もあるかも知れません。

開発：実行の順序を変えるには、その先に実行するコマンドをプッシュできると良いのかなと思います。

社長：コマンドや実行結果をスタックに積んでポーランド記法的に結果の掛け合わせができる面白いですよね。ある意味 repeat は、積んだり降ろしたりの例ですね。

開発：ということで、現在のコードはこんな感じです。

社長：今300行ぐらいですね。どのくらいで落ち着きそうですか？

開発：まあ1000行あれば、そこそこshellっぽいものになるかなと。3000行ぐらいで、けっこう実用的なものになる気がします。Goのパッケージに部品が豊富にあるので、いかに活用できるかかなと思います。

[gsh002.go](#)

ダウンロード

* * *

— 2020-0807 SatoxITS