

株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月11日 投稿者: SATOXITS

続々々々GShell

社長：帰りました (; ∇ ;) ハハハ

基盤：ただいま社内室温34.9度に達しております (; ∇ ;) ハハハ

開発：まあでも、扇風機の風が当たればなんてことはないですけどね。神様がくれた気化熱に感謝です。

社長：外はもう日差しといい何といい、違う惑星に来てしまったかのようでした。



基盤：普通にあづまのウエルシアですけどね。

開発：空気の温度を写真にとらえるのは難しいですね。

社長：それで今日のかき氷を買ってきましたよ。やわらか赤城しぐれ「いちご」じゃ〜ん。

開発：すでにかなり柔らかくなってますね。食べごろと言いますか。しゃぶ。

基盤：みるみる間に解けていきますね。しゃぶ。

社長：既にかき氷を飲んでる状態になりました。じゅるじゅる…。

基盤：ただいま室温35.0度に達しました。

社長：思うに、身体の内側から冷やすのが、最も効率的なんではないかと思うのです。これ以上直接的な方法はないですよ。

開発：扇風機で気化熱方式は良いと思いますが、汗をかかないといけないので、それはそれで気持ちは良くない。なら、シャツの外側に水を垂らして蒸散させるようなやつがあると良いかと。

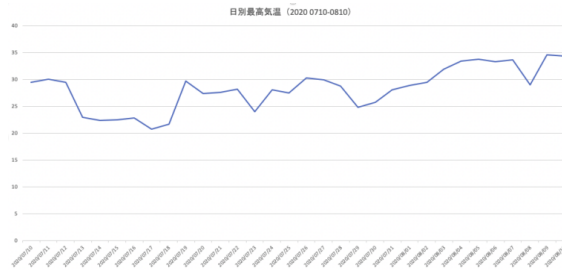
基盤：単にミストでも良いような。ベランダに。

社長：いや以前住んでたところで、日光直射のベランダにある室外機が加熱しすぎて動かなくなった時、ガーデニング用のシャワーを霧状にして当てたら効果がありました。今の状況はかなり違いますが、やってみると面白そうです。今日ボーリングの試合に行くついでにホームマックで探してきます。

開発：やりすぎでお隣さんに迷惑がかからないようにしないと。

* * *

基盤：それで、先日8/2の脱エアコン声明の際に見てた[気象庁のページ](#)で、この一ヶ月のデータをダウンロードしたらこういうことになってました。



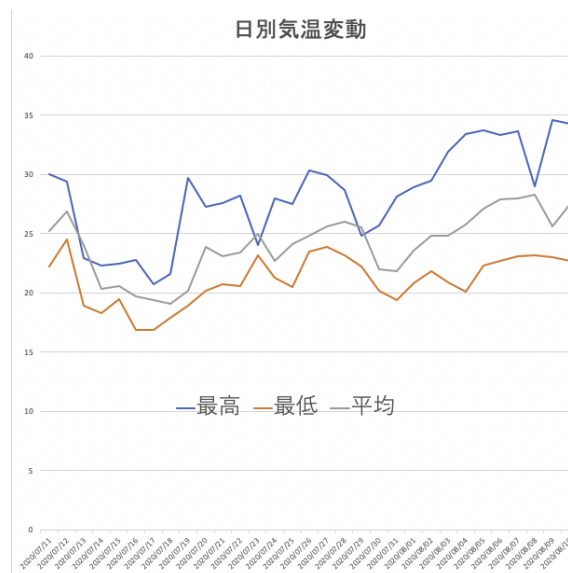
開発：やはり30度を超えると世界が変わりますね。

社長：気温が下がらないので建物が冷却しないというのも問題じゃないですかね。

開発：やはり建物全体に地下水を循環させる計画ですね。

社長：ついでにミストも出しちゃう。

基盤：どっちにしろ入居率100%だから、いまいち工夫するモチベーションが無いかもですが。それで、最高だけでなく最低と平均もグラフにするとこうなります。



開発：なるほど、さがるのはちゃんと下がってるんですね。

社長：確かに、朝はちゃんと涼しい感じがします。

基盤：といいますか、館野？だかどこかの観測所の値ではなくて、当社のこの部屋の室温の変動を知る必要があります。

開発：そだね。ラズパイでやろうという計画でした。

経理：買い置きのカップ麺の蓋がみなさんぶっくりとしているのが気になります。

開発：パスカルの原理ですね。

社長：それだけ気密性が高いってことじゃないですかね。

基盤：だけど、中と外が同じ温度なら、ふくらむのは変じゃないですか？

社長：まだ賞味期限内ですし…

開発：なんでミリバールじゃなきゃいけないのかがどうも。ヘクトなんて単位よりミリのほうが正統ですよ。デシリットルだのCCだのって撲滅したくせに。

* * *

開発：さて、GShell のほうですが、たぶんリスナーを設定してやれば、やりたいことように出来ます。

社長：単にTCPのacceptまでは自分でやって、そこから先をhttpパッケージに投げらるって感じにできないんですかね。ソケットを渡すとか。

開発：そうですね。まあ、並列性の管理とかもhttpパッケージでやりたいんじゃないですかね。

社長：というか、単に自分でacceptして、リクエストのヘッダを解釈して、応答を書き出して閉じる。それだけですからね。われわれのような用途にはそれで十分です。流石にGoみたいに6行じゃ書けないですが、50行で十分に書けますよ。

開発：それはそうと、まずHTTP から受け取った文字列を、端末から受け取った文字列と同じように実行してみます。こんな感じ。

```

MacMini% gsh
--I-- GSH_HOME=/Users/ysato/.gsh
!! serv
--I-- HTTP Server Start at [localhost:9999]
--I-- Got HTTP Request(/date)
--I-- date
Tue Aug 11 15:55:38 JST 2020
--I-- (0.003153913s)
--I-- Got HTTP Request(/echo Hello!)
--I-- echo Hello!
Hello!
--I-- (0.000011446s)
--I-- Got HTTP Request(/df .)
--I-- df .
Filesystem      512-blocks      Used Available Capacity iused      ifree %iused
Mounted on
/dev/disk1s2    489620264 329921072 113298424    75% 2718313 2445383007    0%
/System/Volumes/Data
--I-- (0.006633106s)
--I-- Got HTTP Request(/which ssh -a -ls)
--I-- which ssh -a -ls
-rwxr-xr-x  2157868 Aug  5 21:28:36 /Users/ysato/bin/ssh
-rwxr-xr-x  2142352 Jul 10 07:27:01 /usr/bin/ssh
--I-- (0.006268503s)
--I-- Got HTTP Request(/uptime)
--I-- uptime
15:57  up 5 days,  5:58, 43 users, load averages: 2.45 3.12 3.89
--I-- (0.004976702s)

```

基盤：普通に実行できますね。

開発：まあ、system()関数実行しているのと大して変わらないですからね。で、問題はHTTPですし、それぞれのリクエストは単発で他とはコンテキストを共有していないということです。順序も無い。

社長：複数のユーザが並列に使う場合にはセッション管理が必須ですね。

開発：まあ、そういう使い方をするようになりましたら、各人が実際に、ttyからログインしたのと同じ状態をもたせるんでしょうね。

基盤：ユーザ側とTCPが物理的に切れてもGShellはログイン継続状態で残っているので便利ですね。永久不滅のログインみたいな。

社長：コマンド履歴がすごいことになりそうです。

開発：CLI版の仮想マシンとして生き続けるshellは面白いと思います。

開発：それでちょっとおもしろかったのは、Chromeからやった時に、URLのバーからリクエスト文字列を入れるわけですが、途中まで入れたところで勝手にリクエストとして出したりするということです。大変よろしく無いと思います。

社長：GoにはTelnetサーバパッケージとかないんですかね？

基盤：標準のパッケージには無いですね。FTPもNFSサーバも。

開発：そのへんは、うちでufs3をGo化するとかして提供する道もありますよね。

* * *

開発：それで、HTTPサーバに制御をとられてはこまるので、これを別の並列タスクにしたいわけです。

社長：GoのスレッドってGoルーチンてやつですか。

基盤：御社の御ルーチンとか。

開発：で、どうするのかと思ったら単に、関数を呼び出す時に go というキーワードを前置きするだけで良いようです。なので早速 go という内部コマンドを足しました。

```
MacMini% gsh
!1! go sleep 10 echo ====Finished(^-^)/at %T ====
--I-- (0.000150607s)
!2! date
Tue Aug 11 17:09:51 JST 2020
--I-- (0.002324102s)
!3! ====Finished(^-^)/at Aug 11 17:09:47 =====
```

基盤：楽ちん！

社長：go を実行する前に %T を解釈してしまってますね。

開発：おっと、では%%Tではどうか？あれ、あそうか、go と sleep と echo の3段階に解釈されるから %%%Tでないといけない。これでどうでしょう？

```
MacMini% gsh
!1! go sleep 10 echo ==Fin(^-^)/at %%%T==
--I-- gshellv(6) [ [go sleep 10 echo ==Fin(^-^)/at %T==] ]
--I-- Aug 11 17:28:42 (0.000123324s)
--I-- gshellv(5) [ [sleep 10 echo ==Fin(^-^)/at %T==] ]
!2! --I-- gshellv(3) [ [echo ==Fin(^-^)/at Aug 11 17:28:52==] ]
==Fin(^-^)/at Aug 11 17:28:52==
```

基盤：正解。

社長：うーん、バックスラッシュとか%とか、何段階か解釈ムキムキされると、わけがわからなくなりますよね。なんかうまい方法は無いですかね。

開発：まあこの場合は、echo コマンド実行する時にだけ解釈しろよという指示が与えられれば、%T だけで良いはずですね。つまり、将来的に実行するコマンドについては、起動時点とか途中段階では解釈をしないとか。

社長：見た目ダサイかもですが、go { sleep 10 { echo %T } } みたいに明示的に書いたほうが良いのかも。

基盤：というか普通の感覚なら、逐次的に実行するのは go { sleep 10; echo %T } っしますよね。

開発：いやあ、セミコロンは何かのために温存したいなという気がするんですが。だから go {[sleep 10]{echo %T}} とか。

社長：すべてカッコだけで済ませてる言語はありますよね(^-^)。たしか閉じカッコを省略する記法がありました。例えばインタラクティブモードの時には行末で全てカッコが閉じたことにするとか。go { sleep 10 { echo %T 改行、で済ませちゃう。

開発：そもそもが、全てのコマンドが find の -exec オプション的なものを持ってたら面白いかなと思ひまして。実行した結果とか環境を次のコマンドに渡すのがデフォルトと言うか。

社長：なんかすこい昔に「次に実行するコマンド」みたいな属性を見た記憶があります。まあ、環境を引き継ぐものかどうかは知りませんが。要は順次 exec() して行くみたいな。テールリカーション的なやつ。

開発：それって、shell的なものがない時代の話じゃないですかね。とか、OSが無くて次々にオーバーレイしてくみたいな。

* * *

社長：さてさてそれでは、かき氷と一緒に買って来たメロンシャビーwをいただきますかね。

経理：結局冷蔵庫を復活させたんですね。

社長：やはり文明の利器。せいぜい月30kWhでしょう。どれどれ。ぱかっ

一同：・・・

基盤：水になってますね。

社長：冷凍庫って機能するまでに時間がかかるんですね・・・

開発：精神的ショックのせいか、変な汗が出てきました。

基盤：ああ、扇風機止まっています。

開発：なんでですかね。切タイマーとか設定してないのに・・・

基盤：全力で7Wしか食わないのにそのうえ自動で勝手に切れるとか意味がわかりません。省電力アピールなんですか？わたしはあの、洗濯機槽内照明のLEDが10秒で勝手に切れるのが非常にムカつきます。

* * *

社長：ただいま。

基盤：今日はずいぶん早く試合が終わったんですね。

社長：いえ、お盆期間で試合が休みでした。そういえば先週そんな場内アナウンスをしていたような。祝祭日と年末年始とお盆は試合は試合が休みというルールなんです。20年前の開店時から。

開発：まあ、うちは曜日とか祝日とか関係ないから忘れちゃいますよね。

基盤：夏休みだから一般客で混んでるんでしょうね。

社長：まったくガラガラでした。コロナもありますしね。まあ、昔決めたルールが意味無くなってるのに惰性というか見直す事もなく機動性も無いんでしょうね。経営者が現場を認識してないんだと思います。

開発：試合に出る人がその時期休みたいのを考慮してるのかもですが。

社長：それで、久しぶりにカスミに寄って果物を仕入れてきました。待望のスイカです。じゃーん。

基盤：おお！スイカだ。1年ぶり？

開発：やはりスーパーは入り口にある果物売場が花ですね。ドラッグストアには真似できない。

社長：だもんで、わくわくしてスイカ、桃、プラム、メロン、パイナップル、ぶどう、トマト、とうもろこし、全部仕入れてきました。

基盤：スイカ最高ですね。あま〜い。シャグシャグ。

社長：売れ残りの値引き品ですから、熟成してるかもです。シャグ。ん、あま〜い。

開発：冷蔵庫を復活させておいてよかったですね。ぶっ。

基盤：やはり果物のある生活っていいですね。人間性回復みたいな。

社長：フルーツフルってなもんです。

開発：まあ、続けてるとすぐに当たり前になっちゃうんでしょうけどね。

— 2020-0811 SatoxITS