

USP MAGAZINE

Vol.4
10Yen

よいぬき版

for the sophisticated shell scripters

ついに始動 Open usp Tukubaiプロジェクト

後藤大地 漢のUNIX
with 特別インタビュー
「コミッターという仕事」

TechLION再録
よしおかひろたか
ハッカー中心の企業文化を根付かせよ

壮絶?ガチンコバトル!!
AWK VS Bash どっちが強い?

好評連載
ユニケーゼンエンジニアの作法
シェルスクリプト大喜利 他



hello!

From Editor

読者の皆様、こんにちは。

前号まで副編集長だった**二代目編集長**です。

お世話になった初代編集長の意志を受け継ぎながら、より一層面白い雑誌を目指して舵を取らせていただきます。

最初に申し上げるのですが、私は編集長を一任された社外エンジニア兼ライターで、他にもいろいろやってます。プログラマー、インフラ屋、ライター&デザイナー（同人誌が主）の他、時に作業着に身を包み、様々なビルの機械室に入る仕事もやってたりして、**我ながらなかなかの宇宙人ジョーンズっぷり**です。（最近のビルって裏側も驚く程IT化されてて面白いんですよ）

さて、そんな私が USP MAGAZINE という雑誌をどう考えているか、就任にあたり述べさせていただきます。

私は USP MAGAZINE を、**枯れた技術が切り拓く新しい世界を伝えるメディア**だと考えてます。新技術探検も面白い。でも、何とも思わなくなった技術の新しい使い方を発見するのも、また面白い！ シェルスクリプトにはそういう発見が特に多いから、本誌で話題にすることが多いのだと思います。

もう一つ重要な部分は「メディア」。メディアは中立であってこそ、価値を持ちます。

美しいものは「美しい」。しかしそうでないものは「美しくない」と、それが例え、本誌発行元の USP 研究所のことであっても言えなければならないと思っております。私が社外の人間として編集長を務める理由も、実はそこにあります。

と！いう前置きを踏まえて言うのもアレですが……。

USP 研究所が提唱する、**ユニケーj開発手法は面白い！！**

前号から連載している「ユニケーjエンジニアの作法」や本号の特集1は私が書いているのですが、取材してる段階から実に興味深い話ですよ！ 逆に、これが思いっきり言えるのも、私が社外の人間だからなのかもしれませんね。

とにかく正直な気持ちで、**価値あるメディア**を創ってまいります。

価値あるメディア、創ります。
二代目編集長 松浦智之

Contents

ついに始動 Open usp Tukubai プロジェクト	3
菅雄一 中小企業手作り IT 化奮戦記—第2回	
後藤大地 漢の UNIX	
原色 Linux 美女図鑑—第3回 内山千晶さん	
TechLION 再録 よしおかひろたか—ハッカー中心の企業文化を根付かせるには	14
どっちが強い？ AWK vs シェルスクリプト	
大岩元 今私たちは何を学ぶべきか—第4回	
ユニケーjエンジニアの作法—第2回	
シェルスクリプト大喜利	11
Tech 数独／新刊案内	
ベンチャー起業家のアドベンチャー日誌／編集後記	

※ 薄字で記載している記事は、よりゆき版には収録されておりません。また、本誌とは掲載順序が若干異なります。

特集1

ついに始動

Open usp

Tukubai プロジェクト

USPマガジン編集部

2012年2月、システム開発にシェルスクリプトを本気で使い倒している技術者達が生み出したコマンド集「Open usp Tukubai」が公開された。

だが、蓋を開けてはみたものの、「Open usp Tukubai」の一体何が凄いのか、みな首を傾げるばかり。

それはまるで、飢饉を救う宝として秘蔵されてきた箱の中に、たった一本の鍬が入っていたかの如くだ。

Tukubaiの真意は深いところにあり、分かりづらい。

その真意を伝えるべく、本誌を手にした読者にこの記事を捧げる。

第一章

3分でわかる **Tukubai**

第二章

先人からの贈り物を次世代に — 生みの親に聞く —

第三章

これから始まる

Tukubaiプロジェクト

第一章 3分で分かる Tukubai



Tukubai — それで何ができるのか。それが何の役に立つのか。

Open usp Tukubai は世に公開されたばかりであり、その情報はまだまだ少ない。新しい技術や製品が出た時はまず、それが一体何の役に立つのかということを端的に説明してほしいもの。そこで本章では、そういった疑問に3分で答えられることを目標に、5つのQ&Aにまとめた。

Q1

Tukubai って何？

A1

データベース (DB) をシェルスクリプトに**本気で**置き換えるためのコマンド群だ。

普通なら「DBの利用を検討しなきゃ」と考えるシーン。しかし Tukubai は、シェルスクリプトとテキストファイルベースで、DB と同等以上のことをやってのける。シェルスクリプトの機能を、主に DB 用途向けに補完するためのコマンド群が、Tukubai だ。

その実用性に、まだ疑問を感じる人もいだろう。ところが、あなたがよく買い物に行く大手スーパーやコンビニの商品管理も Tukubai(ビジネス版)で行われている。つまり Tukubai は、本気で作られたモノなのだ。



あのコンビニの裏でも Tukubai が…

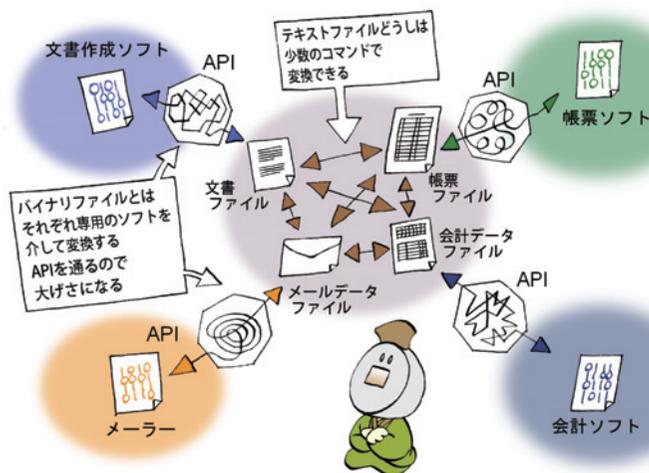
Q2

なぜシェルスクリプトを使うの？

A2

使えば分かる！ ハヤくて、安くて、ウマイことが。

「安易にバイナリデータで管理しようとする文化へ物申す」——これが Tukubai が生み出された根本的な理由である。



バイナリデータで扱うのって大変だよなあ

コンピューターは、データを受け取り、加工し、送り出す作業を繰り返す。バイナリデータの場合、受け取りと送り出しの過程で、APIを叩いてデータを変換し、加工し易くする処理が必須だ。しかし、これは本質的な処理ではなく、コンピューターのリソースやプログラム開発コスト等を、ある意味無駄遣いしていると言える。

これらの作業を、全てテキストファイルで処理できれば、ハヤくて、安くて、ウマイなる。それを体現する手段が**シェルスクリプト+テキストファイル+コマンド**だ。実はこの手段、設定ファイルをほぼ全てテキストファイルにした UNIX 発明者達のやり方に倣ったに過ぎないのだ。

Q3

どんなコマンドがあるの？

A3

COBOL や SQL でもお馴染みの処理を実現するコマンドを、まず 45 個リリース。

DB 処理やテキストデータを加工するものがある。さらに Web もテキストデータ処理が多いことから、それに便利なものも用意されている。(表 1 参照)

Q4

Tukubai は DB に特化したコマンド群なの？

A4

いいえ。シェルスクリプトを**開発言語**として実用化できるほどに、様々なコマンドが用意されている。

コマンドの集まりとはいえ、エンドユーザーにまでコマンドを打たせたくはないところ。そこで web インターフェースに対応すべく、web 系のコマンドも用意されている。

たとえば mojihame。リスト 1 のようなコードを書くと、HTML の <table> タグの中身が簡単に生成可能だ。

LAMP の M と P を、それぞれ Textfile と Shellscript に置き換えた "LATS" を構築することも可能になるのだ。



LAMP ではなく LATS で稼働中のサイト
「USP 友の会」(<http://www.usptomonokai.jp/>)

表 1 Open usp Tukubai コマンドのほんの一例

分類	コマンド名	機能
整形・変換	han, zen	全角文字 ↔ 半角文字 変換
	keta	全レコードを見て、列を桁揃えする
	comma	3 桁毎にカンマを打つ
DB 系	self	特定の列を抽出 (SElect some Fields)
	join2	2 つの表をキー列を見ながら右外部結合 (RIGHT JOIN に類似)
	loopj	2 つ以上の表をキー列を見ながら完全外部結合 (FULL JOIN に類似)
	sm2, sm4, sm5	列毎に全レコードの合計を求める (sum-up)
	ratio	指定列について、全レコードに占めるそのレコードの%を求める
web 系	cgi-name	ブラウザから送信された CGI 変数文字列を URL デコードしつつ、変数名と値に分離
	check_attr_name	CGI 変数に不正な文字が無いチェック
	mojihame	テンプレートテキスト中のマクロ文字を値に置換する

リスト 1

```
#!/bin/sh
cat <<HTML > ./template
<table>
  <tr>
    <td>人</td>
    <td>金</td>
  </tr>
<!-- RECORDS -->
  <tr>
    <td>%1</td>
    <td>%2</td>
  </tr>
<!-- RECORDS -->
</table>
HTML
cat <<INPUT_DATA > ./data
鎌田 5412 万円
濱田 2132 万円
上田 -400 万円
INPUT_DATA
mojihame -IRECORDS ./template ./data
```



```
<table>
  <tr>
    <td>人</td>
    <td>金</td>
  </tr>
  <tr>
    <td>鎌田</td>
    <td>5412 万円</td>
  </tr>
  <tr>
    <td>濱田</td>
    <td>2132 万円</td>
  </tr>
  <tr>
    <td>上田</td>
    <td>-400 万円</td>
  </tr>
</table>
```

Q5

Tukubai はどこで手に入るの？

A5

Web サイト「UEC」で入手できる。日経 Linux 2012 年 3 月号にも収録されている。

MIT ライセンスの無償版である Open usp Tukubai が、UEC (<http://uec.usp-lab.com/>) からダウンロード可能。システム要件やインストール方法等、詳しい使い方はそちらを参照されたい。

また、日経 Linux 2012 年 3 月号でも特集記事として紹介され、付属 DVD に収録されている。



第二章 先人からの贈り物を次世代に

—— Tukubai 生みの親に聞く ——

道具を使いこなすには、発明者が何を思ってそれを作ったのかを知り、その気持ちを理解することが近道である。Tukubai を使いこなしたいのなら、生みの親である當仲氏が何を思って作ったのかを知るべきであろう。そこで、Tukubai の生い立ちを、氏に余す事なく語ってもらった。



OS 設計者の人生観が見えて、システムの作り方がわかった。



とうなか・のぶあき ● 1966 年生まれ、兵庫県出身。株式会社タイエー在籍時に、UNIX システムを再勉強し感銘を受ける（本文参照）。そこで学んだ UNIX 的発想を活かし、当時年間数百億円費やしていた社内システム維持費を数十億円に削減させ、社長賞を受賞。2004 年独立し、有限会社 USP 研究所を創立、現在に至る。

手軽に問題を解決出来るっていうのがシェルスクリプトの面白いところで、シェルスクリプトになって初めて意味があることなんです。

コマンドは道具で、シェルスクリプトは料理のレシピ。道具が揃うと色んなレシピが作れるし、レシピが進むと、こんな道具があれば便利ってことで相互に進化します。やりたいことは、コマンドをオープンにすることより、そのコマンドを使ってどんなアプリケーションが書けるのかということで、今後それをオープンにしていくことが一番の主眼です。

実は OpenUSP 評価版っていうものを 2 年前に出したんですけど、誰も振り向きませんでした。バリューの意味が伝わっていませんでした。コマンドそのものにバリューがあると勘違いされていたから。伝えなきゃいけないのは、コマンドのバリューではなく、シェルスクリプトのバリューなんです。

コマンドを作ったり、データの配置の仕方、書き方のお作法、OS の使いこなし……、それに他の様々な最新技術との繋ぎ合わせ方、そういうもの全体でバリューなんです。

Open usp Tukubai を使うことでどんな効果があるかとか、その裏側にはどんな工夫があるかとか、マネージメント、教育、ガバナンス……と順番に伝えていきたい。その第一のステップが Open usp Tukubai なんです。

——とはいえ、そうやって商売のノウハウになっているものを何もかも開示してしまうと、自身の仕事なくなってしまうのではないかなと思うのですが。

ビジネスの観点からすると「タダにして出しちゃうことで、ノウハウを誰かに取られちゃうんじゃないの?」と思われるかもしれませんが、それは違います。

コンピューティングっていうのは技術の積み重ねで成り立っているの、オープンになったところで誰もが Linux や FreeBSD を

USP コマンドにバリューはない?

—— Open usp Tukubai がどうとうリリースされましたね。私はこの雑誌の編集長という立場上、面白さや凄さは実感しているんですけど、コマンドだけパッと見ても分からないですよ。何と説明すればいいんですか?

まず USP のコマンドって何だろうっていう話です。

バリューのあるものがクローズドからオー

プンになることによって、「どうなっていたんだろう」と思っていたことがパッと明るみになり、手軽に使えるようになります。例えば、暗号化のプログラムとか、それ自体にバリューがある場合、そのアルゴリズムがオープンになると皆「わー」って驚きます。だけど、不思議なことに USP のコマンドそのものにはバリューはほとんどないですね。

面白い部分も多少ありますが、基本的にコマンドひとつひとつは味味な印象を受けます。コマンドを組み合わせることで

一から書けるわけではありません。勉強をしなきゃ出来ません。

勉強の題材を提供しただけで全てわかってしまうほど、コンピューティングは浅い技術ではないと思うんですね。教科書がオープンになったからと言って「じゃあ学校は要りませんね」って言ったら、それは暴論でしょう。だから USP のコマンドがオープンになったって USP のすることがなくなっちゃう、なんてあり得ません。

むしろ、ますます関心を持ってもらえるようになって、色んな人が色んなことを考えて、同じやり方で仕事をする人が出てきて……結果的に我々も活性化します。

そういう風に思っているので、Open usp Tukubai をオープンにすることに何のためらいもないのです。そもそも自分が作ったっていう感覚は全然ありません。

——自分のものではないと？

ええ、先人たちが作った物を壊さないように、埃が降ってきたら払う……そうやって維持していきたいと思っています。強いて言えば、先輩がもう一段石を積み重ねる前に力をつけてしまったから、僕が代わりに石を積んでおこう、と。そういったところで貢献出来ればいいと思っています。

オープンにすることで、そういう姿勢をはっきり示すことになるでしょう。「自分のものだ、俺が書いたんだ」じゃなくてね。

僕は OS やシェルを発明しなかったんですよ。ただ先人達で作ってきた物の上で、あーでもないこーでもないと、ちょこっとやっただけ。その、ちょこっとやった物をクローズドにするなんて、メンタリティ的によくないでしょ。オープンにしてくれたから僕らが勉強が出来たわけです。思い返せば大学の時、UNIX のソースコードを読んだということが、何十年が経っても今のベースになっているのは間違いありません。だからオープンにしてくれたことに対する感謝の気持ちはものすごくあるんです。

そういう水のような、空のような愛に包まれているんだから、僕らはそういう物の子供だ、ということを姿勢として表すことは間違っているとは思わないですね。僕は今 45 歳なので、20 年かけて貯め込んだものを

30 年くらいかけて吐き出さないといけない(笑)、それくらいのつもりでやっています。

パイプを何十本も繋いでメーカーの人に嘲笑われた

——Tukubai というコマンドセットは、どのような経緯で作られたのですか？

大学の時に UNIX があったので、そこで学んで……と言いたいところですが、実はあまり勉強せずに、アルバイトばかりしていました。勉強は単位を取るため、卒業するため。

大学を離れて、アルバイト——肉体労働だったり接客だったり物売りだったりをやりました。それで、たまたま前職（大手スーパー）のプロジェクトチームに入って、そこで UNIX をもう一度勉強し直す機会があったんです。

上司がソニーの NEWS という UNIX ワークステーションを使い慣れているという理由だけで「僕ちゃんは NEWS が大好きなんだ。お前も使え」と言われました。そういう偶然から全てが始まったんです。「そうか UNIX か、なんか聞いたことあるな」というか、やったことある気がする(笑) と思って。

大学の第二外国語みたいな、それくらいの忘れ方をしていた時に、再びコンピューターと接することに。5 年ぶりくらいでした。

それで、「入門 UNIX」みたいな本を買って読みました。学生の時はず単位が取れば良かったので、書いてあることを表面的に理解すれば事足りましたけど、その時は現実的な問題がバーンと目の前にあって、それを解決しなければならなかったんです。会社の建て直しとか、データなどのハンドリングをし

て、いかに役に立つ情報を導き出すか、とかね。その時、UNIX の本に書いてあったことの意味が初めて理解出来ました。

小さな道具を組み合わせることで問題を解決するほうが柔軟性に富んでいて、スピードが早く、無駄がないということ、それが必要に迫られてみると、よくわかりました。僕はダイエーにいて、商売の世界にいたので、販売や在庫の分析、そういった中から有利な商談の方法とか売れている物、利益があがる物を見出すことを迫られていたんです。そんな時に UNIX の考え方がピタリと感性に合った。

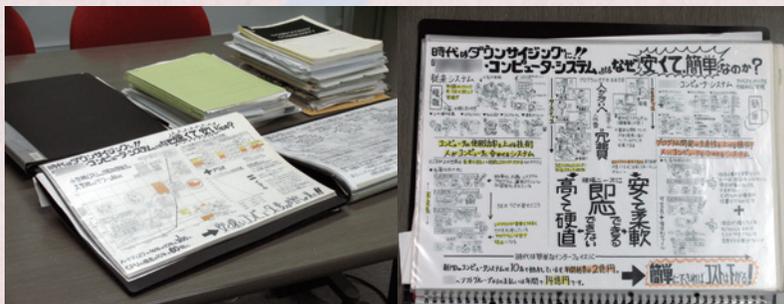
たぶん OS もそういったことから作られたと思うんですね。だから OS を設計出来る人っていうのは、自分の人生観や感性をアーキテクチャに反映させたのだと思うんです。

——人生観！そこまで読み解けるのですか。

意味がわかったことで、実にじっくりきました。UNIX に出合ったのがコンピューターの会社じゃなくて、問題が起こっているユーザー側の会社だったことも良かったと思います。誰も周りにコンピューターを知っている人がいなかったで、自分で一から勉強するしかなかったんです。

一から勉強とは、原理から勉強することです。問題を目の前にして、それがいかに素晴らしいことかわかりました。OS の生まれた背景を知らずに、プログラミングの技術だけで解決しようとするとおかしなことになってしまうよね。

今に至るお手製のコマンドを作ったのも、UNIX っていうのはコマンドを組み合わせる file to file でデータを順番に変換しながら、欲しい編集をするもので、その為には



▲前職在籍時に制作したというプレゼンテーション資料。UNIX 的発想がいかにシステム効率化・コストダウンに数效であるかということ、これらの資料を使って当時の首脳陣達に直接説明したという。数百枚にも及ぶそれらの資料のほとんどは手書きで作られており、そのパワーに圧倒される。

コマンドを作ってもいいし、コマンドとコマンドをパイプで結びつけていけばいいということが書いてあったからです。

——なるほど、USPの開発スタイルの源流はそこにあったんですね。

COBOLの時代から積み上げられてきた業務システムに必要な処理は、セレクト、ソート、マッチング、サムアップ、項目間演算……と、おだまりがあるんですが、UNIXのコマンドに用意されていない物がいくつかあったんです。

例えばサムアップ。コンピューターはデスクの電卓のはずなのに、単純に大根合計6本って出すための簡単なコマンドがないんです。大根1本、2本、といちいちファイルをリードしてオープンして……やりたいことは大根6本、って出したいだけなのに。

多分UNIXを作った人って天才なんで、そんな物は当たり前で作れたと思うんです。けど、そこまでやってしまっただけ残して、あとは自分たちで考えて作れよ、って。

僕みたいな凡才には切れ味のイイ物は作れませんけど、数を作って、その中から選んで残していけばいいんじゃないかと思って作り始めました。コマンドを作るっていうのも、当時のメーカーの人からバカにされましたよ。そんな物を作っても意味がないと。

パイプをいっぱい作って何十本と繋いでいってデータを流して、「やったー。ほら出来た！」と言っても、当時、シェルでパイプを繋ぐなんてありえなかったのを、それを見たメーカーの人が嘲笑いましたね。「コンピューターの使い方を間違えてる」って。

たぶん、嘲笑った人は大学時代に勉強して、コンピューター会社に入って、新しいことを追いつけることがコンピューティングだと思いついていたのでしょう。僕の場合、目の前にある問題を、一番ベーシックなアーキテクチャで解決するには、どうしたらいいか、って考えられる環境に自分がいました。それが非常に幸運でした。僕はラッキーの塊ですね。「シェルはあれもこれも出来ない、制約がある」って言われましたけど、僕はそれは解決していくものだと思ったんです。

ls コマンドは、 出来の悪いコマンドの見本

——そのやり方に先人の教えに基づく信念があったとはいえ、初めから確信を持ってたわけではなかったのではありませんか？

初めは素人がやることだから、そもそもの要求が鋭くなかったんですね。自分が出ることしかやりませんでした。なので、自分がギリギリリーチできる中で長年揉まれて、「あ、お前、これも出来るんだったらこれも」って順番に高い所にニンジンがぶら下げられていくうちに段々脚力がついてきたんです。

ドライバーは、
車と一体になれる。
だからマニュアルを見ながら運転しない。
コマンドも、そうあるべき。

その過程で、生まれたコマンドもあれば、淘汰されたコマンドもありましたね。同じプログラムでも最初AWKで書いていたのを次はCで、アセンブラで、とかね。同じ物でもアルゴリズムを工夫しようだとか、コマンド間で書式を統一しようとか、そういうことで段々練られていきました。その書式に関しても、オプションも増やすだけだとか何がなんだかわからなくなるので、出来るだけシンプルにマニュアルのいらぬコマンドを作ろうとしました。優れた技術は人に近づく、という言葉がありますけど、要はマニュアルを見ながら車を運転している人はいないでしょ。例えばハンドルは、見た目の感覚で右に回したら右に曲がるような気がする。それが、実際にも腕と連動して右に曲がるから、操作し

やすいものになっているんです。

コマンドは2000個以上作りましたが、そうした試行錯誤の後に残ったのは、シンプルなものだけ、数十個くらいしかありません。

——2000個も!? そうすると残ったコマンドは相当エリートなんですね。

いつも例に出しているのは、スプーンとフォークとナイフだと言っているんです。その3つがあれば、どんな食べ物でもうまく切ったり口に運べたりしますよね。スプーンとかの形を見たら、俺でも作れる、って世界中で多くの人が思うでしょう。でもスプーンやフォーク、ナイフがこの世に全く無かったとしたら、食べ物を上手く食べられる食器を一から考えるのは難しいでしょ。一度じゃ上手くいかなくて、変な食器を発明しては、口を切ったりこぼしたり……そして、結局はあいう形に落ち着いていくんだと思います。

ところで、lsコマンドのオプションって暇な人が数えたら80くらいあるらしいですよ。出力の幅を揃えるようなオプションだとかいろいろあるんですけど、foldやcutコマンドを組み合わせてやればいいのに、lsコマンド自身に色んな機能を実装してどんどん増えていったんですよ。だからあれはUNIXの中では最低なコマンドと言われていて、出来の悪い物の見本として覚えておけ、みたいなことが何かの本に書いてあったんです。

なので、Tukubaiのコマンドにオプションなんて物はほとんど無いんですね。基本的に頭に入らないものはコマンド化しません。こんなこと言うと怒られちゃうかもしれないけど、マニュアル見ながらコマンド打つなんて最低ですよ。歌手は楽譜を見ながら歌っちゃ駄目。歌手と歌は一体化していないと。

Tukubaiは一見、大したことないように見えます。一個一個が単純なので、ナイフを見る時のように「何だこんな物」と思うかもしれませんが、色んなシーンで使ってみると、そのコマンドでしか出来ない、このコマンドがあって良かったな、というものを揃えています。

——Tukubaiのコマンドは名前も独特ですよ。日本語そのままのものも多いです。

lsは「List」の略だしcatは「concatenate」の略、grepなんてGlobal Regular Expressionの略、

略です。なぜ略なのかと言えば、短いコマンドの方が早く打てるという現実的な理由です。速記と同じですね。Tukubai も同じで、速記が出来、かつ種類を抑えながら万能化するにはどうすればいいかということ重視し、コマンドの数はあまり増やさないようにしています。増やせば名前が長くなりますから。

かと言って、新しい物をつくることに否定的なわけじゃないですよ。使ってみて手になじむようだったらそっちにするとか、試行錯誤は常に行われています。

Tukubai の由来は「這いつくばう」こと

— USP 研究所のロゴも "つくばい" なので、その由来を聞きますが、なぜコマンドセットを Tukubai と名付けたのですか？

コマンドで、現実にある様々な問題に挑んできました。でも世界に目を向ければ、やはりその問題は限定的で、日本という環境の中のある偏ったところで生まれたものだと思うんです。つまりここで作ったものは日本の文化を色濃く反映しているはずですよ。企業文化や日本そのものの文化であったりとかね。なので、Tukubai って名前も日本文化を象徴する、茶道の精神にちなんで名づけました。

— 茶道の精神が、プログラムのお作法に通じているのでしょうか？

茶道の精神って色んな物を足していくという発想ではなくて引き算なんだそうです。例えば人間って丸裸では生きていけないので、何か身につけるわけですけど、最低限必要なものは何かという話になった時に、余計な物を削いでいったら何が残るでしょうか、というのが茶道の精神の根底にあるんですね。UNIX文化にも通じる物があるように思います。

また、茶道はお茶の追求だけじゃない。茶室があって、右から入るだの左手を添えるだの、座り方がどうの、書や華を愛するだの、非常にうるさいんです。あれはなんだろうと思っていましたが、突き詰めると相手に失礼のないように、かつ自分の言いたい事も通すために、無駄な動きや会話をしないための立ち居振る舞いを追求する道なんです。

USP のロゴや今回のコマンドセットの名として頂いた「つくばい」の語源。これは「這

いつくばう」で、人に頭を下げるとか、相手に畏敬の念を持って礼を尽くすという意味です。茶道でつくばいは水を汲んで手を洗うための器です。それは凄く低い位置に作られていて、それを使うためには膝を曲げてかがまないといけない。そういう姿勢をすること自体が礼儀を表現することに繋がるんです。

こういった、余計な物を削ぐとか礼を尽くすといった精神は、ある意味企業情報システムを作る精神に似通っているな、と。会社っていうのは無駄な動きをしちゃいけませんよね。勿論、何もなくても駄目。的確なことをテキパキやらないといけないでしょ。ちゃんと技術を進歩させつつ、内部崩壊もさせず、マネージメントもこなし、社員のやる気を維持しなきゃいけません。それをやるために無茶苦茶お金をかける、なんて乱暴なことをやっている、駄目になっちゃうんですね。システムを作る側の人間はそういう企業の立ち居振る舞いを理解していないと良いシステムは作れませんよ。

先人が積み上げた技術を得て、エンジニアとして自律して欲しい

— 今のお話を聞くと、Tukubai という名称自体に強いメッセージが込められていると感じたのですが、Tukubai を通じ、改めて技術者に伝えたいことはありますか？

イメージしているのは、「エンジニアの自律と協働」です。僕が良く使う言葉なんですけど、自分で立つじゃなくて、律する方の自律ですね。僕は、エンジニアはサラリーマンじゃなくて職人だと思っているんです。

給与所得者ということで立場はサラリーマンなのかもしれませんが、エンジニアから3K（危険・帰れない・きつい）と言葉を漏らしてほしくないです。それは自律していない人の愚痴ですよ。誰かが何かをしてくれる、という依存の気持ちがあるから、「僕がこんなに辛いのは誰かのせいだ」という甘えた気持ちになるんです。そういう甘えは職人にはあるまじき態度だと思います。

技術は自分でストックに勉強していかなくちゃいけない。だけど、技術が進歩していくならば、シンプルなことでも物が進むわけ、技術を追いかける時間を減らせるんです。昔

はOSのインストールでも丸一日かかったのが、今は20分とかで出来るわけです。その余った時間で酒を飲むのも、デートするのも、それでも別にいいんですけど、とにかくその時間を活かして自分のことを見つめ直す時間にあてて欲しいです。

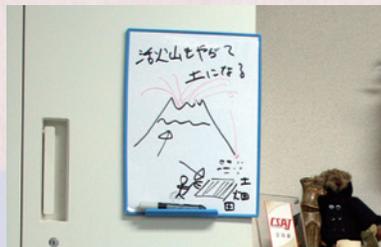
Open usp Tukubai を使えば、先人が積み上げた技術によって、時間や精神的な余裕を生み出せるでしょう。その余裕を使って自律をして立派な職人になって欲しいです。

そういう自律した人たちが共に働いて、独りじゃ成し得ない大きな仕事してくれたらな、と思います。突き詰めれば、原点に立ち返り、働くことの意味とか目標を見つめ直すきっかけになってもらえればと思います。

技術の積み上げにも悪いところはあって、技術が向上して効率化するとボーッとしちゃうんですね。どういことが積み重なって今のその幸せな状態に至るのを見失います。だからいつだって原点に戻れるということは大事。原点回帰っていうか、積み上げていくことも大事だし、時には階段を下りてみるのも必要。

シェルはベーシックなものなので階段も下りやすいです。一から勉強し直そうと思えばいつでも出来ます。そうやって原点回帰して、今現在の自分の立ち位置を振り返ると、毎回、新しい発見が出てくると思います。惰性に流されるのではなく、日々新しいものを見つけたいです。

原点回帰しやすくするためには、シンプルなものがいい。Open usp Tukubai はそういうのを目標にしています。出来れば使う人にもそういうメンタリティでいて欲しいです。



▲新技術と枯れた技術の違いを言い表すために當仲氏が描いた絵。噴火に目を向けるのが新技術に注目することで、噴出物が堆積した土地を耕し畑にするのが、枯れた技術を洗練させること。氏は後者が好きだという。

第三章 これから始まる Tukubai プロジェクト



Open usp Tukubai の公開は、Tukubai プロジェクトの第一歩に過ぎない。Tukubai はごく単純な道具であり、使い方を知らなければ、真の価値を見出すことが難しいからだ。

Tukubai の使い方を公開する準備も着々と進行中である。それが UEC という名の web サイトだ。

ユニケーj開発手法のポータルサイト "UEC"

Unicage Engineers' Community (UEC) は、Open usp Tukubai の公開と対を成す Tukubai プロジェクトのもう一つの重要なコンテンツだ。

シェルスクリプト+テキストファイル+自作コマンド群 (Tukubai 等) という開発スタイルをとる「ユニケーj開発手法」を実践する開発者達に向けた、ポータルサイトを目標している。

このサイトはコンテンツマネジメントシステム (CMS) により管理されているが、汎用品ではなく、ユニケーj開発手法によって一から作られている。HTML や JavaScript 等全部含めても、プログラムはここまで 500 行足らずである。

Tukubai は、世の中で求められる様々な要求に対し、全体の見通しが効く単純なシステムを作って応えるのに向いているが、やはり知っておくべきノウハウがある。誰もが手軽にこの情報を得られるようにすることが、UEC というサイトに課せられた使命である。



Unicage Engineers' Community web ページ
(<http://uec.usp-lab.com/>)

UEC サイトのコンテンツ

Open usp Tukubai のダウンロード配布・オンラインマニュアルは勿論、プログラミング Tips 等を公開しており、メニューは随時増えていく予定だ。公開して日が浅いため、まだ実現できていないものもあるが、現在進行形で各種サービスの準備を進めている。

ジャーナル ユニケーj開発に関連したホットな情報を提供する。	ダウンロード Open usp Tukubai 等、ユニケーj開発に関連したソフトウェアを提供する。
オンラインマニュアル Tukubai コマンドマニュアルを公開し、いつでも Web で見られるようにする。	プログラミング Tips ユニケーj開発手法の概論からコマンド活用法まで、プログラミングに役立つ情報を提供する。
リポジトリ オープンソースソフトウェアとして多くの人々に Tukubai の開発に参加してもらおうための場。	コミュニティ 公開元とサイト利用者 (開発者)、あるいはサイト利用者同士が交流し、情報を共有するためのスペース。

UNIX 哲学の価値を今、改めて世に問う

UEC の目的は、Tukubai の情報提供を通じて一つのオープンソースコミュニティを形成することだが、その先にはさらに大きな目標がある。

Tukubai は、形の上では USP 研究所の著作物ではある。しかし當仲氏はインタビューで、先人から学んだ UNIX 哲学を自分達なりに表現したに過ぎないものと語っていた。つまり、Tukubai を世に送り出すということは、40 年以上前に提唱された UNIX 哲学の価値を、改めて世に問うことなのである。日々登場する新技術に注目が集まり、この哲学への関心は薄れているかもしれない。だが今、見つめ直す時期に来ているのではないだろうか。コミュニティ活動を通じ、様々なシーンで活躍中のエンジニア達に、問うていきたい。

シェルスクリプト大喜利

第四回

司会：『もっど吹く』編集長・みかん

毎度お馴染み、3か月のご無沙汰でシェルスクリプト大喜利(略して sh 大喜利)のコーナーがやってまいりました。あたくし司会進行役のもっど吹く編集長みかんです。表の肩書が編集長になっちゃって「それじゃ吹けないじゃん」って困った挙句、「もっど」を頭に付けることにしました。元もっど吹く編集長ってね!……………はい、**今日も寒いですね。**

えー、ここは読者参加型のコーナーです。ということで、今日もまずは本コーナーのシステムをご説明!

シェルスクリプト大喜利とは

読者の皆さんに毎号お題を出して回答を募り、面白いものを紹介します。そして、良い回答には段位を授与、悪い回答では段位を強制返還させてもらいます。面白さに応じて授与・返還される段数は様々です。そして、見事十段になりますと!!**筆舌に尽くしがたいちんじゅうちゃんグッズ**をプレゼント!日曜夕方のアレとほとんど同じですね。ただ、扱うものがシェルスクリプトということでいくつか特有のルールがあります。以下にそれを記します。

シェルスクリプト大喜利特有のルール

- 一、sh 大喜利はクイズやテストではありません。なので決まった答えというものはないのです。あえて言うなら面白いスクリプトが正解!
- 二、面白いスクリプトとは例えば、こんなもの。
 - イ、人が考えつかない意外性がある
 - ロ、美しい or 芸術的 or 記述がシンプル・短い or 高速
 - ハ、アイデア・こだわりが光る
 - ニ、ネタになるようなバカバカしさ、くだらなさがある
 などなど、ただし最後のは段位強制返還の恐れありよ。:-)
- 三、スクリプトの動作環境は Linux とします。そして、特に断りがない場合は、Linux JM(<http://linuxjm.sourceforge.jp/>)に記載されているコマンド及び機能のみ使用可能とします。これは多くの人を楽しめるようにするためなのです。(但し JM に載っているので、**C シェル系での回答も OK!**)

四、sh 大喜利はシェルスクリプトを披露する場なので、Perl や Ruby、Python などは使っちゃダメです。そもそも JM にも載っていません。逆にシェルスクリプトにとって不可欠な awk や sed 等は OK です。JM にもありますし、でも、よっぽど面白ければ、なきにしもあらず? ルールもおさらいしたところで、それじゃイッてみよう。

本番開始

<第一問>

テキストを下の行から順番に表示させたい時、*BSD なら tail -r、Linux なら tac コマンドでできます。でもそれぞれ互いの環境では使えません。そこで、どちらでも使えるシェルスクリプトを書いてください。

Linux 使いの人は *BSD に tac コマンドが無いことを知らなかったり、*BSD 使いは Linux に tac コマンドなんてものがあるって知らなかったり、いろんな人に聞いてみると結構ローカルネタであることがわかって面白いです。(FreeBSD 派なアタクシも tac なんて知らなかった!) さてこのお題 (にばかり)、回答がたくさん来ましたよ。泣く泣く紹介できない方もいますが、皆さんごめんなまし。

◎@bsdhack さんの回答

```
#!/bin/sh
ex -s ${1} <<-EOF
g/ /mo0
%p
EOF
```

ほほう、ex エディターを使ってくるとは! 上の行から順に「(その時点での)0 行目に移動」という動作を繰り返すと逆順になるというわけか。いいね、二段撥としちゃおう!

◎松澤二郎さんの回答

```
sed '2,$G;h;$p;d'
```

今度は sed で来たか! 実は sed で回答した投稿者が何人かいたのですが、これ一番シンプルだったんですよ。しかも tcsh でも動作するようにと、"!" を使わない記述で再投稿。うーん気が利くねえ。これも二段撥とだ。

◎◎S◎P◎の◎会長さんの投稿

```
cat file | while read line ; do touch $line ; sleep 1 ; done ; ls -t | cat
```

来たな、会長！コメントによると、ファイル内にスペースや変な記号がないことが前提らしい。……確かに動くな。遅いし実用性0だけど、面白いから奇跡の**初段検与**だ！

◎杉田◎上海◎さんの回答

```
cat -n - | sort -rn | cut -f2-
```

パイプ使いこなしでこれはいい！行番号付けて降順ソートして行番号を外すわけだ。わかりやすい、**二段検与**！

◇ ◇ ◇

今日はサクサク行くよ！第二問。

<第二問>

『間違え計算機』

おかしな計算結果を出力するスクリプトを書いてください。

前回の printf のおかしな挙動を受けて出したこのお題。例えば、丸め誤差とか、あるいはバグとか。そういう珍しいものをスクリプトにしてもらいたいわけです。

◎hi_saito◎さんの回答

```
awk 'BEGIN{print int(70.21*100)}'
```

さすが、日本 GNU AWK ユーザー会会長！26 頁ではお疲れ様でした。へえー、こんな簡単な演算で表れるんですね。しかも逆に int 付けない方が期待通りの結果になるよ。こりゃ気を付けないと。面白い！**初段検与**！

◎phck◎さんの回答

```
php <<PHP_CODE
<?php
ini_set("precision", 19);
echo round(10000000000000000.001) . "\n";
echo round(10000000000000000.125) . "\n";
?>
PHP_CODE
```

こりゃ **PHP びゃなひか!** まあ一応シェルスクリプトの体を取ってるからいいか (いいのかよ!)。添付コメントによると、最新 ver.5.3.10 で確認したとのこと。5.3 系だと前者の round は成功するけど、後者の丸められないという、PHP 界限を賑わすホットなネタですな。じゃあ**初段検与**ね。

◎◎S◎P◎の◎会長さんの回答

```
#!/bin/bash
a=3.999999; x=$1
for i in {1..100}; do
    x=$(echo "$a*$x*(1-$x)" | bc )
done
echo $x
```

また来たな！えーなになに、引数に 0.3, 0.33, 0.333, … と指定していくと結果がでたらめになると。……確かに、収束しそうな漸化式なのに収束する気配がない。まあ、丸めには気をつけろということですね。うむむ、奇跡の**二段検与**。

◇ ◇ ◇

さあそれじゃ、今回のメインイベント、第三問！

<第三問>

本誌表紙のマスコット「ちんじゅうちゃん」のイラスト入りマウスパッドを作ろうと思います。でもマウスパッドは GUI 派の為の道具。CUI 派代表としてそんな GUI 派に向かい、彼(彼女?)に一言言わせてください。ただし、簡単なシェルスクリプトのコードをセリフの一部にすること(コードのみでも OK)。

いやいやアタクシが今使っているマウスパッドがつまらなくて、新しいのにしたかったんですよ。それでこのお題を募ってみたわけですが……。

皆さん、ちゃんとコードを添えてくださいませ。一応このコーナーのお題なのでプログラムでないとカッコがつかないですから。面白いけどコードがなくてボツというのが結構あって残念なのです。さて、そんな中で残った回答。

◎後藤◎大地◎さんの回答



これ、今号からの新連載「漢の UNIX」のインタビューやった時についてお願いしてみたんですが、即答でした。さすが FreeBSD のコミッター、殺しているのは FreeBSD のコンソール上をふらつくマウスカーソルのデーモンです。

とりあえず、マウスを使うなってことですか！どうもありがとうございました。**初段検与**させていただきます。

◎U◎P◎の◎会◎長◎(ケ◎ン◎シ◎ロ◎ウ◎, 34 歳)◎さんの回答



むむつ、またまた出たな！身近にたまたま山田さんとい

う名前の悪党でもいたんだろうか……。ホームディレクトリを/tmpに放り込むとは何て残虐なんだ。山田さんはきっと悲しむぞ！（というかログインできない）

まったく、例え悪党相手でも、ちんじゅうちゃんに言わせるべきセリフじゃないよ！これはひどい。せつかく奇跡の二段授与したばかりだというに……。 **段位回収**っ！

そしてそして、いい回答が来たんですよ。

◎猿二号さんの回答



コメント「Linux JM に載っていないし、今後でもどう考えても載りそうにないですが、ダメもとで投稿してみます」

いえいえ、これはすばらしい！JMには載ってないけど面白ゆからそんな望むひのことはヌキ。確かに、こういう操作ミスで楽しい思いできるのはCUIならではだ！！本人にとって必ずしも楽しいかどうかはわからないけど、複数人でマジメな作業をしている時であればあるほど周囲に笑い(or 気まずい空気)を提供できるね。 **三段授与**しちゃおう！sl コマンドについてご存じない方はここを見るべし。

<http://www.tkl.iis.u-tokyo.ac.jp/~toyoda/>



というところで本日の大喜利はこれにてお開き！読者の皆さん、投稿してくれた皆さん、ありがとうございました。

投稿大募集!!

シェルスクリプト大喜利は皆様からの投稿あってこそ成り立つ企画です。お題に対する回答、お題そのものを広く募集致します。さて、次号のお題はこちら！

次回のお題

一、本号の特集1のインタビュー中でfold というコマンドの話が出てきました。恥ずかしながらアタクシは知りませんでした。どうやら指定文字数に達したら改行するコマンドのようですね。これ見てふと思いました。

「fold コマンドで商と余りを求められないかな？」

というわけで、コマンドラインから2つの数値を受け

取り、fold コマンドを使って商と余りを求めるスクリプトを書いてください。ユニークなコード大歓迎！

二、つづけて記事からのネタですが、漢のUNIXにてrs というコマンドが出てきました。でもこれ、*BSD 特有のコマンドです。そこで今度は、このrs コマンド相当品をシェルスクリプトで作ってください。

ただし簡単にするため、下記の簡易仕様とします。

- ・ターミナルの縦横サイズの自動検出はしなくてよい
 - ・データは標準入力からのみ受け取る
 - ・-t オプション指定時の動作(行と列を転置)とする
 - ・引数は rows と cols、2つの数値だけを必ず貰える
- rs のわかり易い例は ls の表示です。ls ってファイルを ↓ ↑ ↗ ↘ … と並べますね。rs はああいう並び方をさせるコマンドです。詳しくは FreeBSD の man で。

<http://www.jp.freebsd.org/man-jp/search.html>

三、make マニアの皆さん、お待たせしました！make コマンドで「普通そんなことしねーよ」という変わったことをやってください。BSD make でも GNU make でも OK。

新ルールのお知らせ

シェルスクリプト大喜利ではこれまで、Linux JM に載っているもののみ使ってよいことになっていましたが(まあ面白ければそうでなくても採用しちゃってましたが)、Open usp Tukubai が公開されたということで、**Open usp Tukubai コマンドを使ってお題を解いても OK** ということにします。ただし、それなりに美しいコードになっていないと採用の可能性は低いですから、心してください。

Open usp Tukubai はここ⇒ <http://uec.usp-lab.com/>

投稿のしかた

お題への回答は、お名前(ペンネーム)、回答したいお題番号、回答スクリプト、簡単な補足の四点セットで下記の宛先へ！一人何問でも何個でも回答可です。尚、次回締め切りは **5月28日(月)午前0時**とします。しかもその間は何度でも回答の修正を受け付けます。

お題もどしどし送ってくださいー!

それからお題も大募集。考えてくれた方にも段位を授与します。自分で出題して回答するのも、今のところ可！

投稿先

どちらも投稿先は、mag@usp-lab.com です。 **アド**

レスが替わりましたんでご注意を！本誌をイベントで購入した方も、ぜひ参加してくださいね。

TechLION



よしおかひろたか(楽天株式会社/カーネル読書会主宰)が語る

ハッカー中心の 企業文化を 日本で根付かせるには

日々移り変わる技術。

その技術の密林を、いかに生き残るべきか。

TechLIONは「本物の技術者の生き様」を
浮き彫りにするトークライブイベント。

2011年最後を飾るTechLION vol.5。

大忘年会にふさわしいゲストとして、

楽天株式会社 技術理事のよしおかひろたか氏をお招きした。

「カーネル読書会」をはじめ、数多くの勉強会、
コミュニティを主催する

オーガナイザーとしても有名な同氏だが、

このような交流会を開催する意義と目的とは何か。

「幸せなプログラマ」が世の中に求められる背景とは。

そして、ハッカーセントリック(中心)文化の先に見据えるものは。

『プロの酔っ払い』を自称するよしおか氏の、

呑むほどに冴えわたるトークの一部始終をここに再録する。

(聞き手:法林浩之、まとめ:柏崎吉一)



法林: 2011年最後のTechLIONは、大忘年会ということで、ここ新宿ロフトプラスワンで開催することになりました。来場者の皆さんも呑んで食べて楽しんでいってください。

今日お招きする方も、すでにかなり出来上がっているという噂です。

よしおか: よろしくお願ひします。先につまみを頼んでもいいですかね。枝豆かなんかあると、ビールが進むんで。ではまず、かんぱーい。

(会場: かんぱーい♪)

よしおか: 今日話したいテーマは3つあり

ます。「ハッカー中心の企業文化とは」「なぜ、それが必要か」「どうやってそれを日本で根付かせるか」です。

(枝豆が到着)

ありがとうございます! テンションが125%くらいに上がりました。それでは、ハッカーとは何か、ということから。もちろん、コンピュータを利用する犯罪者のことではありません。辞書的には、コンピュータ技術に精通した人。もう少しいえば、ちょっとした技巧(ハック)を操る人。ライフハックなんていう言い方を

最近しますね。しかし、私は、社会を変える人のことだと思ひます。ザッカーバーグはFacebookを作って社会を変えました。私も大学を卒業した時は、そんなハッカーになりました。行動するエンジニアというか、プログラムで何かを変えるのに憧れていました。

法林: そういう原点があるんですね。

よしおか: で、そのハッカーたちの発言を見ると、共通する価値観が浮かび上がってくる。その価値観について分かりやすく説いている本が、スティーブン・レビーという人が

1970年代に書いた「ハッカーズ」という本です。和訳も出ていますが、ご存知ですか？（会場：ちらほら挙手あり）

法林：読んでいる人が結構いますね。

勉強じゃない。君の価値はGitHubのコードで決まるんだ

よしおか：この本には、ハッカー倫理というもの紹介されています。たとえばコレ。「コンピュータへのアクセス、加えて何であれ、世界の機能の仕方について教えてくれるものへのアクセスは無制限かつ全面的でなければならぬ」。つまり、**何でも試そうよ、実験しちゃおうよ。誰に許可を求めるでもなく、先にやっちゃおうぜ、という考え方**です。この手の発想が暴走すると時に法律の範疇を超えてしまうこともあるんですが、大事なのは、非常にラディカルで、自由な思想だということです。同著には「**権威を信用するな—反中央集権を進めよう**」「**ハッカーは、学歴、年齢、人権、地位のようなまやかしの基準ではなく、そのハッキングによって判断されなければならない**」。つまり、君の価値は、GitHubに上がっているコードやパッチで決まるぞみたいなね。

法林：今の時代でいえば、ですね。

よしおか：実際にシリコンバレーにある、テッキーな企業に応募しようとLinkedInでレジメを書く時、GitHubのアカウントがないと面接すらしてもらえない。私も時々、プログラマ志向の日本の学生から「将来のために、どんな勉強しておけばいいでしょうか」と訊かれます。でも、大事なのは「勉強」じゃない。それより、コード読もうよ、オープンソースのコミュニティに出入りしようよ。日本の若い人たちが、いい素質を持っているのであれば、ぜひとも世界に羽ばたいてほしい。だからこそ、シリコンバレーの「当たり前」を知ってほしいんです。

ハッカー倫理には、「芸術や美をコンピュータで作り出すことは可能である」という節もあります。さらに怖いのは「コンピュータは人生をよいほうに変えうる」という圧倒的な楽観・楽天主義。そして、ラフなコンセンサスと動くコード。**許可を求めるな、謝罪せよ**。これらが60年代のハッカーズが共有していた価値観なんです。

法林：ラフなコンセンサスと動くコード、は

今もよく言われますよね。

よしおか：そうですね。全員のコンセンサスを得るのは不可能に近いから、ある程度動く物を作って、これ面白いぜ、やろうぜ、となる。それで社会とコンフリクトが起きたならば調停する。あらかじめオプトインしてもらわない。Googleなんて、その最たるものですね。こっちで先に情報をたくさん集めちゃいました、ですから。

法林：ストリート・ビューみたいに。

よしおか：そうそう、コンフリクトが起きてから折り合いをつける。法務部とかで「これやると法律に触れるよね」なんて許可をもらうために議論して3年も待つような日本では、新しいものは到底できっこない。

もちろん、ハッカー的か、そうじゃないか、というのは0か1かときちつと分かれるものではなくて、どっちにより軸足を置くか、というスペクトラム、幅があります。一人の人間でも、「やっちゃおうよ」と思う時もあるわけでは、「これやばいよね」と思う時もあるわけなんです。

最高のプログラマは、最高のプログラマと働きたがる

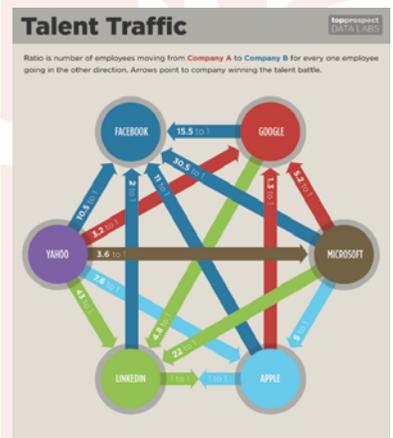
よしおか：それで、いまのハッカーの価値観というのは、何か。伊藤穰一さんの指摘がヒントになります。「インターネットの信念体系とは、誰もが接続する自由。イノベーションする自由。そして誰の許可を得ずともあれこれといじくる自由を与えられるべきだ、というものだ」「中央集権的に管理することは不可能で、イノベーションはネットワークの「外縁」で小規模なグループによってもたらされる」というものです。伊藤穰一 MITメディアラボ社長、いいこと言うね、みたいな。

法林：社長じゃなくて、所長ね。

よしおか：すみません。カツゼツ悪くて、で、次に「Yahoo!に起きてしまったこと」というポール・グレアムさんのブログが2011年夏に結構話題になりました。彼はもともとプログラマだったのだけど、その後、ベンチャーキャピタリストとして、目利きをしながらスタートアップ企業に投資している。その彼いわく、**Yahoo!がハッカーセントリックでなくなった大きな理由が、儲かり過ぎてしまったことにある**と。ハイテク企業にな

ろうという強い意志がなかった、とまで言い切っている。Hadoopを作ったりして、私はYahoo!ってすごい企業だと思うんだけど。ただし、企業というのは、儲かることが前提。でも、儲かり過ぎると、ハッカーセントリックになれない、という指摘です。確かに、Yahoo!はインターネットバブルの時、広告で儲かった。そのYahoo!にポールの会社が買収される。彼は、Yahoo!の役員にテッキーな提案をするけど興味を示されなかった。そこで違和感を覚えたことがきっかけで、そのブログを書いたんです。

この指摘は鋭いです。タレントトラフィックにも表れている。シリコンバレーを中心に活動するハッカーがどの企業へ転職しているか、を表すグラフです。2011年6月時点のデータですが、Yahoo、Facebook、Google、Microsoft、Apple、LinkedInの中で、かつての一番人気だったGoogleを超えるのがFacebookです。Yahoo!からFacebookへ10.5人も移っているのに、FacebookからYahoo!へは1人だけ。ちなみに、Yahoo!とGoogleにおける人材流動の比率は、3.5人対1人。10年前はMicrosoftが強かった。シリコンバレーに限らず、人材はあっちこっちに動くのが当たり前です。ハッカーたちは、自己実現できる場所、**いま一番ホットでセクシーな会社を求めて本能的に動く**。今この瞬間すでにFacebook以外の企業が人気かもしれない。ハッカー中心文化のあるシリコンバレーでは、特定の会社が人を囲い込めない。



図：「タレントトラフィック」アメリカの大手IT企業間の人材流動の様子を表している。

搾取することができないんです。

ところが**日本は、いかに安い単価で人を雇い、高く売れるものを作らせるか、という人月単価の国**です。シリコンバレーとはまったく違う。

ソフトウェアって、人が作っているんですよ。その当たり前のことが理解されていないのが今の日本。インターネットの時代に、ソフトウェアのビジネスを立ち上げたいならば、プログラマがその価値を創造していることを、経営者はまずは認めなくちゃいけない。素人を100人雇うのと、最高のプログラマを1人、2人、3人雇うのとどちらがよいか。少数精鋭がいいに決まっています。**大リーグでチーム作る時も、イチローとか、トップノッチの選手を雇うのが当たり前**じゃないですか。最高のプログラマは最高のプログラマと働きたがる。いまたまたま Facebook に人がいるのは、そこにハッカーがいるから。そういう場所を作らないと日本はヤバいよという話です。

良いソフトウェアは 幸せなプログラマが作る

よしおか：ポールは「どんな企業にハッカー中心の文化が必要なのか」について言及しています。それは「良いソフトウェアを必要とするすべての企業」だと。つまり、全業種の企業なんです。トヨタもソニーも、マスコミだってそうです。つまり、**インターネットを使っている企業であればどこでも、人材が動いて新しいものを生み出す、ハッカー中心文化が必要である**、という主張です。

私もこれには大賛成です。理由は、ハッカー

中心文化のほうが、自分にとって遥かに心地いいから。ただ建前として組織の偉い人に説明する際には、次のように言っています。

まずは、「共通善ですよ、社会をよくしよう、という価値観をみんなと共有できますよ」。次に、「企業の競争力につながりますよ。Google や Facebook に負けちゃいますよ、会社がつぶれたら元も子もありませんよ」。そして、「ソフトを楽しく安く安全に早く作るベストプラクティスをハッカーは知っていますよ」。これらを諄々と説くわけです。経営者もバカじゃないから話せばたいい分かってくれます。だって彼らも儲かるから。シリコンバレーの経営者は、それを直観的に理解しているのだと思います。もし、理解できないならば、その経営者はバカなんですよ。でも、そう決めつける前に、まず我々技術者が、経営者にそれはおかしい、そんなことしてたら会社が潰れてしまうから賢い方法で行きましょう、と説明しましょう。潰れるより儲かる方が絶対にいいからです。

良いソフトウェアやサービスを作るのは誰か。幸せなプログラマである、というのが私の信念です。宗教と言ってもいい。では、幸せなプログラマを作るには、どうしたらよいか。我々はこれを考えるために今日、ここに集まっています。実現すると、日本人が世界に出て Facebook や Twitter、Google と戦えるんじゃないかと妄想しています。ここまでで質問ありますか？

質問者 A：今お勤めの企業（楽天）でハッカー中心の文化の導入を実践されて、手ごたえは

ありますか？

よしおか：2009年に**楽天に入社して以来、私のミッションはただ一つ、会社が潰れないようにするため、社内にインターネットの常識を伝えること**なんです。週1回、社長や執行役員とミーティングする機会があり、意見を交換するのですが、2年経っても変わらないのが実感です。楽天がめっちゃくちゃ儲かっているんですね。

法林：あれ、Yahoo!と同じじゃないですか。

よしおか：楽天のサイトには、3万7千店が outlet しているのですが、ご夫婦で経営されるような小さな店も多い。イカの押し寿司を手作りで販売しているようなお店です。つまり、3万7千のストーリーがある。そういうサービスを作るのは、幸せなプログラマだと思っています。ただ、いまのビジネスモデルがうまく行っているので、現状肯定派も多い。あれ、残り5分ですか。手元にプレゼン資料がまだ30ページ分あるんですけど。

法林：質問1個で、回答にすごく時間がかかったらゲストだと分かったので先に進めます。素敵なプログラマ人生を送るために、どうやってハッカー中心の企業文化を作っていくべきなのでしょう。

技術は会社のものではない 社会のものである

よしおか：企業文化というのは、外側の人からは分からないんです。楽天もそうです。伝えるには、内側の人が自ら、大事だと思っている文化を話し、継承していかなければならない。現場で起きていること、**個人に蓄積されたノウハウなどの暗黙知を、いわば“民族誌”のように明示的に伝え遺すことが大事なんです。必要なのは、パッションと仲間**です。場は、勉強会、ランチ、飲み会、宿舎などいろいろ作り出せます。

組織というのは、何かの目的を持って設立するのですが、肥大化すると、「他の事業部になにする人ぞ」といったタコつぼ化が起こります。そこで、横串を通し、文化を融合させる必要性が出てきます。そこで有効なのがコミュニティです。

YLUG（横浜 Linux Users Group）のメーリングリストの議論がきっかけで始まったカーネル読書会は、99年4月から数えて2011



年11月で108回を数えました。98年末にシリコンバレーから帰国し、Linuxのカーネルを読みたくなり、YLUGのメーリングリストでSystem Callの実装について質問したところが懐かしい。

法林：まじめにカーネル読んでいたんだ。

よしおか：もちろんですよ（笑）、それ以外の活動もしていますが。私自身も当初、このインターネットの時代に、カーネル読みたい奴なんているかな、それを着に人が集まるのか、と半信半疑でした。でも、蓋を開けてみるといっぱい集まった。今日みたいに。で、技術を日本語で議論する、Linux Kernelへの貢献、他の勉強会とのコラボなど達成できた。第100回にはリーナスもきてくれた。伊藤さんが首謀者だったんだけど、この話をすると後3時間くらいかかるから今日はカットしますね（笑）。

こういう勉強会を社内で行うメリットは数え切れない。社内で仲間ができて、黙って受付やエレベーターガールとかやってくれるボランティアが現れたり出会いと発見が多いんです。上司も興味を持ち、社内スポンサーになってくれた。**皆さん、孤独じゃないですよ。絶対にこういう風に絡んでくれる人がいる。**外部から集まる人材との交流によって、組織内部の開発者は刺激を受けます。同じ会社の人でもあっても知らない人との出会いは、モチベーションアップにつながります。

新しい技術知識を得ることは副次的な効果でしかありません。誰が何をしているのか、理解することが重要です。対外的にはこうした勉強会の開催・運営は企業のイメージアップにもつながります。

もちろん、オープンにやる以上、リスクとコストを考慮します。例えば、情報流出のリスク。会場以外への立ち入りを禁じればいい。会場費もかかるけど、社外でやるのに比べたら格段に安い。勤務時間外にやれば会社に迷惑かけない。「開催のメリット>開催のコスト」（よしおかの勉強会第一の法則）が満足されていれば、会社も納得します。カーネル読書会以外にも、いろんなコミュニティを運営してきました。皆さんの会社にもいろんなコミュニティがあると思う。そこでやるのは、参加者が持っている暗黙知を形式知に変え、

言語化する作業です。

オープンイノベーション時代は、社外に価値の源泉を求めざるを得ない。特に、オープンソースの世界はそうです。会社に閉じてもってはいけません。コミュニティ的なノリが求められるんです。ハッカー的というと、そもそも技術は会社のもではなく、社会のもの。社会をよくしていくという価値観が前提にある。コミュニティはそのための道具なんです。

すごいハッカーはいっぱいいる 君は決して孤独じゃない

質問者B：社内にハッカー文化を持ち込もうと苦労している一人ですが、何かコツがありますか？

よしおか：嘘をつかないことですよ。自分が「これしかない」と信じているから、相手にも伝わる。小手先のテクニクはバレます。それより、「いいから俺が言うんだから間違いないよ。とにかくこっちが得だよ」「カーネル読書会に行くと、こんな素敵なことがあるんだよ、あなた知らないでしょ」ということを延々と言い続ける。最初は『こいつ、うさんくせーな』と思われるけど気にしちゃいけない。

私は、自分自身が幸せになりたいと思っています。会社の経営者が、何かビジネスのネタを持っていて、それを実現するために「君の技術が必要なんだ。その技術をで何かを作ってくれよ」とお願いされる方が、安く買い叩かれるより100倍気持ちいいじゃないですか。たとえ、給料が同じでも、そういう経営者と仕事をして貢献したいですよ。

私自身が幸せになる道だと思うからこそ、日本にもハッカー文化を根付かせたい。米国発、シリコンバレー発のルールに一方的に乗せられるのではなく、日本発で一緒に何か新しい文化を作り出していきたい。**酔っ払いの親父が枝豆食いながら、6割くらいはいいこと言っているね、とツイートしてくださいね、みなさん！**

質問者C：IT以外の分野においても、たとえば食だとかアパレルのような領域でも、埋もれた人材、才能がいっぱいあって無駄遣いされているのではないかと思います。それについて、よしおかさんはどう考えますか？



よしおか：そういう分野ともどどん交流すればいいと思います。**勉強会やコミュニティというのは、有為な人材を見つけて「君は孤独じゃないんだよ」と声をかける場所、仲間を作る活動**だと思います。日本は元気がない、という発言を時々耳にしますが、そんなことはない。東京なんて無茶苦茶面白い。すごいハッカーもいっぱいいます。でも、世界に打って出ている人もいます。若い人にもっと外に出て行けよ、と背中押しています。もったいない。TechLIONなんて、素敵な場所ですよ。色々なジャンルの人が集まって、「ノンセクションの“50”」（※）みたいな。

法林：年齢高くないと分からないギャグだけど、いまのツイートされそうですね。

よしおか：法林さんも、Lightweight Language イベントは10年やっているでしょ。**世界でも例がない。**

法林：勝手にみんな集まってきてるだけですけどね。米国のLLは2年で終わりましたしね。

よしおか：こういう人が日本にいることを**我々は、もっと誇りに思った方がいい。**

法林：ありがとうございます。僕もかなり才能の無駄遣いと言われてはいますけど。

よしおか：最後にもう一度、言います。コミュニティへの積極的な参加、その運営を通じて、個人の持つ暗黙知を仲間と共有し、継承することが、幸せな社会への第一歩。海外にも日本から情報を発信していきましょう。そんなところで時間かな。今日はありがとうございました。

※ 70年代に放映されたクイズ番組「クイズグランプリ」で使われたセリフ

< イベント予告 >

TALK LIVE THURSDAY 12 APRIL TechLION vol.6 まつもとゆきひろ



TechLION vol.6

MC : 法林浩之、馮富久 Talk Guests : 後藤大地、閑歳孝子、角俊和
2012.4.12 Thursday start 19:30 ~ close 22:30
Ticket : 前売り・予約 2700 円、当日 3200 円 (1 ドリンク込) Place: 六本木 SuperDeluxe
主催 : ユニバーサル・シェル・プログラミング研究所 協賛 : 技術評論社 後援 : 日本 UNIX ユーザ会

USP MAGAZINE バックナンバー 好評発売中!



USP MAGAZINE は、**世界初! シェルスクリプト技術情報誌**。でもご存知のとおり、シェルスクリプトはグルー言語。OS 深層から様々な言語・アプリの話まで、さらには技術の先にいるエンジニア達にもスポットを当てます。**目指すは、シェルスクリプトとエンジニア達の地位向上!**

☆自炊不要☆ 定期購読又はバックナンバーをお申込みいただくと、PDF 版が無料で手に入ります!

ご購入、お申込みはこちらから! =>

USP友の会 会員募集中

シェルでつながる技術の輪! なんでもありのグルーコミュニティ

UNIX/Linux/ シェルスクリプトの可能性を極限まで追求します!



詳しくは
<http://www.usptomonokai.jp>
にアクセス!

・約 300 名の会員が定例会や勉強会で活発に交流をしています。