

月刊

本誌正式名称は「シェルスクリプトマガジン vol.23」です
Shell Script Is The Supreme Glue Technology
FOR THE SOPHISTICATED SHELL SCRIPTERS
世界で唯一の

シェルスクリプトマガジン

2015
3 March
Vol. 23
¥500

新連載 よしおかひろたかの IT 散歩日記

TechLION 再録 セキュリティ大特集
満永拓邦 × 楠正憲 × まり

縁の木、育てよう

組織文化を変える汗と涙の物語

アジャイル改善塾

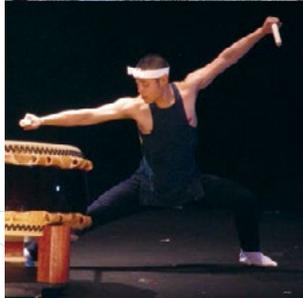
リナ女のたしなみ

特集

円滑コミュニケーション 世界を救う!

中小企業診断士が解説する、超実践的な会話術!

CONTENTS



世界で唯一の
月刊 **シェル**
スクリプト
マガジン
2015 March vol.23

- 04 中小企業診断士が解説する、超実践的な会話術!
特集 円滑コミュニケーションが世界を救う!
濱口誠一
- 12 Haskell版Open usp Tukubai 完成させるぞ企画
Haskellでやってはいかんのか? 第12回
上田隆一
- 16 **新連載 よしおかひろたかのIT散歩日記**
よしおかひろたか
- 18 **ユニケージ開発手法 コードレビュー 第12回**
大内智明
- 23 **姐のBENTO**
- 24 **リナ女のたしなみ 第5回**
平愛美
- 26 **縁の木、育てよう 第6回**
白羽玲子
- 28 **漢のUNIX 第19回**
後藤大地
- 34 組織文化を変える汗と涙の物語
アジャイル改善塾 第11回
山海一剛
- 36 **Ipv6新時代を体感しよう! 第6回**
波田野裕一
- 40 **戸倉彩の2.5次元の世界 第2回**
戸倉彩
- 42 **スズラボ通信 第15回**
すずきひろのぶ
- 46 **法林浩之のFIGHTING TALKS 第10回**
法林浩之
- 48 **UNIX大好き人間はどう育つ? 第2回**
鳥居高之
- 50 **TechLION再録 セキュリティ大特集**
- 56 **ユニケージエンジニアの作法 第13回**
松浦智之
- 60 **人間とコンピュータの可能性 第23回**
大岩元
- 62 未来に生きる!現場で使える!
データモデリング 第13回
熊野憲辰
- 65 **Tech数独**
- 66 **人間諦めが肝心**
シェル魔人/編集後記



ちんじゅうちゃん

本誌名称について
vol.21より「USPMAGAZINE」から「シェルスクリプトマガジン」に、変更となりました。バックナンバーは「USP MAGAZINE」としてお求めいただけます。
本誌正式名称は「シェルスクリプトマガジンvol.23」となります。



花見月

今回の特集のテーマはコミュニケーション！

まず一番に重要なのは「相手を理解すること」。

特集

田滑コミュニケーションが世界を救う!



中小企業診断士が解説する、
超実践的な会話術！

Writer：濱口 誠一



まずは自己紹介をさせていただきます。

筆者は、大学生時代、英語のディベートを行っていました。ディベートでは論理で第三者を説得することが求められるので、論理的に話す力を学んできました。

その後、就職。皆さんも痛いほど感じているとおり、社会人におけるコミュニケーションというのは論理だけでは通用しません。特に、筆者は調剤薬局チェーン店に勤務し、企業を買収して、買収された先のマネジメントを行うという仕事をしていたので、論理が通用しない場面には山ほど遭遇してきました。

その後、IT企業に転職。業界用語と常識を一から覚え直しながら奮闘し、現在は経営企画部のマネージャーを務めています。また、経済産業省認定資格である中小企業診断士を保有し、社外ではビジネスモデルに関するセミナーの講師や創業支援なども行っています。

筆者は、論理的に話すことを前提にしながらも、感情や理不尽に遭遇した場合にどう対処すればいいのか、を苦闘しながらも実践してきました。また、中小企業診断士として、様々な業界の様々な背景を持った方々と交流させて頂いており、これらの実体験も折に触れてご説明していければ、と思っています。

「自分は一生懸命やっているのになぜ評価されないのだろう？」

「これだけ説明しているのになぜ理解できないのか？」

「自分が言っても納得しないのに、なぜあの人が同じことを言うと素直に聞いてくれるのか？」

ひいきか！ えこひいきなのか!？」・・・

ビジネスシーンでもプライベートでも、このような「理不尽」な目に遭遇することがよくあります。そこで、「相手の理解力が低いから」と相手のせいにしてしまうことは簡単かもしれませんが、自分はちゃんと伝えるべきことを伝えたのに、それを理解できない相手が悪い、と。

しかし、ここで立ち止まって考えてみて下さい。自分が言ったことが相手に伝わらない状況と相手に伝わっている状況を比べてみて、どちらが得でしょうか？

自分を正しく理解してもらわないと損をしてしまう可能性が高いです。努力が正しく評価されなかったり、余計なトラブルを招くこともあります。

一方で、自分を正しく伝えると、より適正に評価される可能性が高くなり、自分の意見を通すことも容易になっていくので、場合によっては、多少サボっても大目に見てもらえることもあります(笑)。

このように「伝える力」は非常に重要です。また、相手によって知識も理解力も異なる以上、正しく伝えるためには相手を深く知ること欠かせません。この連載では、これらの力を伸ばすための「伝える技術」「聞く技術」にフォーカスします。また、伝える、聞くなどのコミュニケーションは決して天性の才能や子供のころの環境で決まるものではありません。ビジネスで必要になるコミュニケーション力というものは、「技術」で後天的に身につけることが十分可能です。

皆さんのビジネスを助ける実践的ですぐに使える技術をお届けしていきたいと思えます。

コミュニケーションはよく言われるように言葉のキャッチボールです。

ただし、ここで投げる球は1つではありません。

投げる球は2つで「内容」と「気持ち」。論理と感情と呼んでもよいでしょう。



ここで重要なことは、相手とTPOに応じてこの2つの優先順位が変わる、ということです。どちらの球を相手が欲しがっているのか？ どちらの球を投げるほうが自分にとって都合がよいのか？ 無論、どちらかが100%、ということはありませんが、どちらを重視すべき状況なのか？を考慮しておくことも大切です。

ここで内容と気持ちのどちらを重視すべきか？また、内容と気持ちでどのような内容を伝えるべきなのか？これは、すべて「相手が決めること」です。そのため、今回は、一番重要な要素である「相手を理解する」技術をご紹介します。

「相手を理解する」技術

「上司の指示通りに作ったはずの資料なのに『指示と違う』と激怒され、徹夜で考えた企画が一瞬で紙くずになった」「相手のためを思って提案したのに逆に相手の逆鱗に触れてしまいこじれてしまった」という問題がビジネスでは多々あります。「そこまでひどい職場じゃないぜ!」という幸運な方もいるかもしれませんが、大なり小なり、上司や顧客の指示通りに動いたにも関わらず、謎の地雷原にぶちあたったという経験はお持ちだと思えます。

無論、明らかに理不尽なものや、場合によっては意図的に陥れようとする事もあるかもしれません。しかし、よくよく考えてみると、お互いの誤解が重なり合った結果、こじれてしまったことや、相手の求めるものを十分にわかっていなかったから失敗してしまったことも少なくないはずです。

トラブルを事前に避けるためには「相手のことを知ること」が重要です。では相手の「何を」知ればいいのでしょうか？まず、なぜトラブルが起きるのか？を考えてみて下さい。

たとえば「企画書が紙くずになった」ということや、「顧客や上司といった依頼主が求めた内容ではなかった」という場面。相手の要件を具体的に確認することも当然重要ですが、そもそも「相手の求めているものが何なのか？」をつかんでいなかったことが原因と考えられます。依頼を受けた際に、個別に確認することも大切ではありますが、往々にして仕事の依頼者は、自分自身の目的を明確に把握してないことが多いのです(悲しいけど)。そんな奴とは付き合わない、というオプションは、なかなか取りづらい以上、何らかの対策を打つ必要があります。ああ、サラリーマンはつらいよ。

ここで知るべきは2つ。相手の「興味・関心」と「理解度」です。相手は自分の興味・関心によって、出されたアウトプットが宝かゴミかを判断します。また、全く理解できない用語が並んでいたら、そもそも宝かゴミかの判定自体が不可能です。

以下で事例を交えながら、2つの切り口について説明したいと思います。

想定の切り口1 相手の興味・関心があること / ないこと

相手は、何に興味や関心を持っているのか？何をメリットと感じ何をデメリットと感じるのか？そして、それはなぜか？ということを考えてみます。

具体例で考えてみましょう。仮にあなたが情報システム部の部員で、システム部長から「社内SNSの導入を検討してほしい」という指示が出たとします。下記の3つの例に対して、どのようなアプローチが考えられるでしょうか？全く同じアプローチでいいのでしょうか？



例
1

システム部長の興味は出世。役員からの評価をメリットと感じており、「この会社ではシステム推進できるのは自分しかない」という根拠のない自負がある。しかし、細かい技術的な背景などには興味がなく、役員に対して訴求しやすいメリット・デメリット、費用対効果に関心が高い。

例
2

システム部長の興味は部員の成長。仕事には厳しいが、エンジニアとしてスキルアップして欲しいと思っている。社内政治には関心がなく、実は「部長」という肩書きも窮屈に考えている。

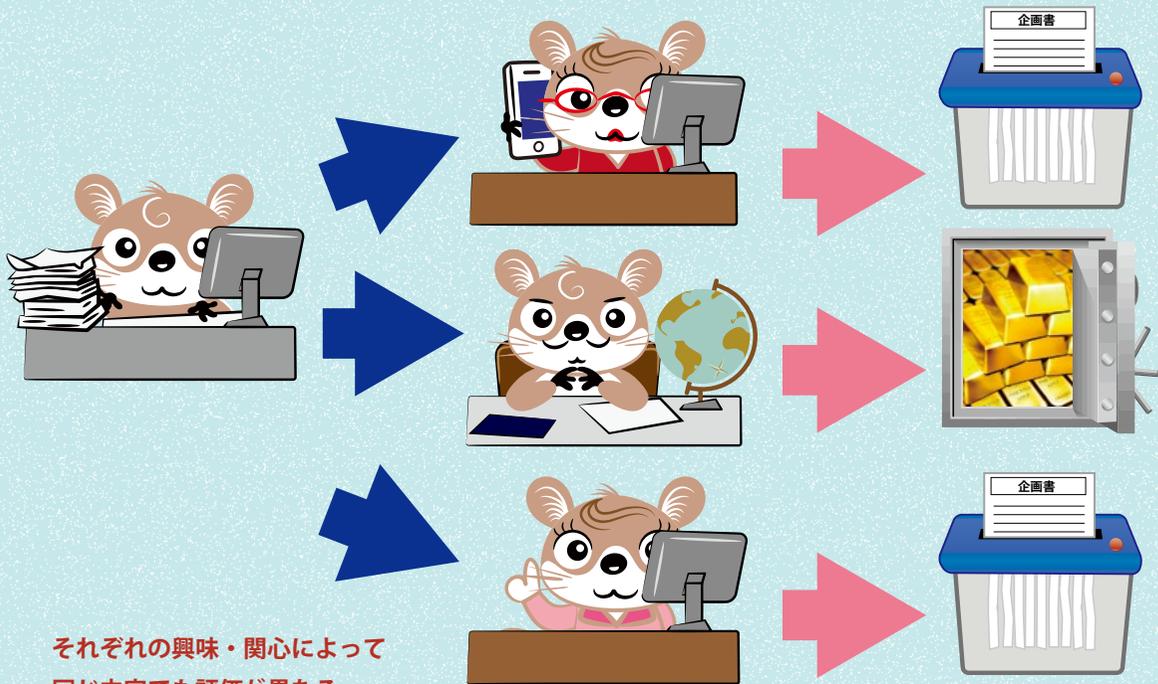
例
3

システム部長の興味は技術。自分自身もプレイヤーでいたいという想いが強くあり、勉強熱心でもある。ただし、経営や現場への興味が薄く、作ったシステムが利用され効果を出すかどうか、は自分の責任の範囲外と考える傾向がある。

たとえば、例1の部長に対しては、社内の情報交流の促進など、「SNS導入の費用とSNSを導入した結果、社内にもたらされるメリット」が好まれるでしょう。しかし、例2、例3の部長に同じ話をしたらどうでしょうか？逆にSNSの機能比較などは、例3の部長には好まれると思いますが、例1の部長は聞く耳すら持ってくれないかもしれません。

このように、相手によって、指示の目的も変わってくれば、求めるものも変わってきます。相手の興味・関心によって、何を伝えるべきか？ということが変わってくるのです。





それぞれの興味・関心によって
同じ内容でも評価が異なる

読者によっては、次のような反論を考えるかもしれません。「例1の部長が好む話をすることはできる。ただし、実際には、情報交流などのメリットよりも、システム部としてはスムーズに導入・運用できるのか?という視点の方が大切。部長にへりくだって本質を捻じ曲げるべきではない」と。

確かにそのとおりです。しかし、相手の興味・関心を知るといことは決して、相手に従属するということではありません。むしろ、相手の興味・関心を知ることが本来やるべきことを実現するための手助けとなるのです。

たとえば、例1の部長に対して、「AというSNSは、導入費用は安い、メンテナンスしづらく固まりやすい。システム部員の手間がかかるので、Aは導入しないほうがよい」ことを伝えたかったとします。ここで「手間がかかる」ことが、

例1の部長にとってのデメリットとなるかどうか?を考える必要があります。仮に「手間」をデメリットと考えてくれないのなら、「対応工数がかかる＝コスト増になる」という観点ならどうでしょうか?もしくは「役員がSNSを開こうとしてもすぐに固まってしまう。こんなに遅いなら入れる必要がなかった、という叱責につながるリスクがある」という伝え方ならどうでしょう?

これらはあくまで一例ですが、同じことを伝えるにしても、相手の興味・関心を知っているか、知らないかで、自分の意見がどれだけ通るのが全く変わってきます。そのため、相手の興味・関心、相手は何をメリットと考えて何をデメリットと考えるのか?を知った上で、そこに届くように内容を考える必要があるのです。



想定の切り口2 相手は何を知っていて何を知らないのか？

相手の理解度は？用語についてどの程度知っているのでしょうか？背景をどれだけ知っているのでしょうか？
どのようなソースから相手は情報を得ているのでしょうか？

筆者は様々な背景を持つ人たちとビジネスを行ってきましたが、ビジネスにおいては「自分の常識は相手の非常識」というくらいに考えたほうがよいと思っています。それくらいに、自分の知っていることを相手が知らないことが多いのです。特に専門性が高く勉強熱心なエンジニアの「当たり前」は、顧客にとっても、第一線から離れてしまった上司にとっても、決して当たり前ではありません。

そこで相手の知っていること、知らないことをしっかりと考えておく必要があるのです。たとえば、先ほどの「社内SNSの導入」の例で考えてみましょう。指示した上司は何をどこまで理解しているのでしょうか？

SNSを使ったことがあるのか？(=実体験として、何ができて何ができないか？を知っているのか？)

社内SNS導入をサポートするベンダーについてどの程度の理解があるのか？

導入に伴う、他のシステムへの影響について理解しているのか？

SNS導入により、社内の情報交換の成功事例・失敗事例を知っているのか？

それらの事例の要因を理解しているのか？

この理解度によって、説明すべき内容も変わってきますし、何よりも説明で使う「用語」も全く変わります。相手の理解度を考えずに話をしてしまうと「君のいうことには小難しい言葉が多すぎてよくわからない」という悲しい回答が待っています。



このように、何かを伝えるときには、

- ・相手の興味、関心
- ・相手の理解

を考えるとスムーズにいきやすくなるので、ぜひ、一度お試しいただきたいと思います。



仮に、考えようとしたけど浮かばない、ということでしたら、厳しいことを言うようですが、それは普段から相手のことを観察していない証拠と言えます。普段の相手の言動に対して、反発したり馬鹿にしたりするのではなく、何に興味や関心があって、何を理解しているのか？とプロファイリングするくらいのつもりで見えていくとよいでしょう。

とはいえ、具体的なコツに何も触れないままですと、連載一回目から読者の皆さんや、そして何よりも厳しい仕事熱心な編集部から嵐のようなダメだし改善要望が飛んでくることが想定されますので、いくつかコツをご紹介します。



普段指摘されたことを記録する

何もかもを記録するのは相当に大変なので、レビューを受けたときなど、相手の興味・関心が表れやすい場所に絞って指摘されたことを記録します。その際には、何を指摘されたのか?だけに留まらず、なぜ指摘されたのか?(何に興味・関心があったのか?)を推測して書くといでしょう。

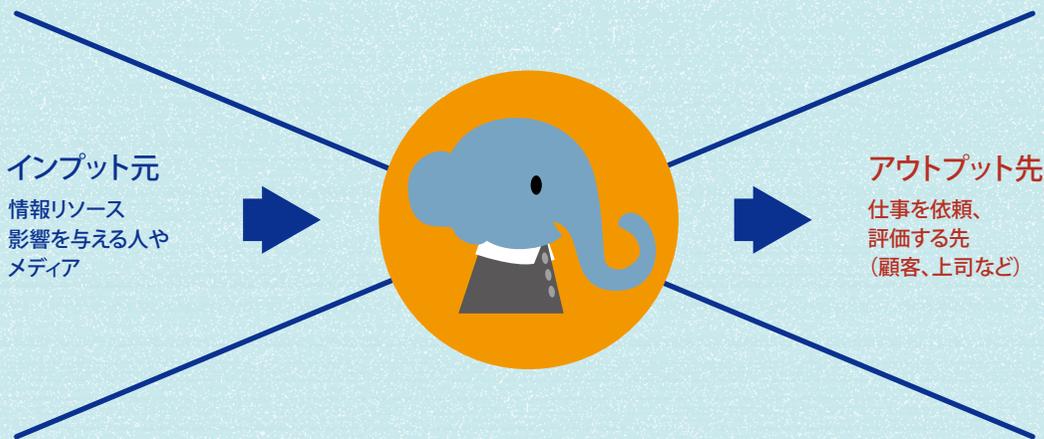
日付	相手の名前	レビューテーマ	指摘事項	指摘をした理由 (興味・関心)	特記事項

相手の視点に立ってみる

相手の気持ちなんかわかんねーよ」という声も聞こえてきそうですが、相手の「アウトプット先」と「インプット元」をつかむことが、相手の視点に立つためのヒントとなります。

インプット元とは、相手の情報リソースです。日経新聞かもしれませんし、2ちゃんねるかもしれません。同期のネットワークや、家族の意見なども挙げられます。

アウトプット先とは、相手が仕事の依頼先、相手の仕事の評価をする機関や人、を指します。たとえば情報システム部長であれば、役員かもしれませんし現場の事業部長かもしれません。



ここでは相手の判断に影響を与える度合いが大きいインプット先をつかむことが重要です。日経新聞やワールドビジネスサテライトが主要な情報源の場合は、パスワードに関心が高くなるかもしれませんし、幅広いビジネストレンドには理解がありますが、個別の事例の詳細な理解度は薄いかもしれません。

時には聞いてみる

謙虚・沈黙が美徳とされる日本社会なので、なかなか直接、相手の考え、特に上司の考えを聞くことは難しいかもしれませんが、時には直接、興味や関心を聞いてみることも1つの手です。人は頼られて嫌な気はしませんし、自分が知っていることは話したい生き物です。「○○さんしか頼る人がいないので教えてください」「△△という仕事を成功させるために□□を教えてください」などと、直接聞いてみることも大切です。この「引き出し方」についても様々な技術があるので、次回以降の

連載で取り上げたいと思います。

また、飲みニケーションも重要です。「どうせ行っても説教されるだけだし…」などと避ける人もいますが、それは非常にもったいない。飲み会は絶好のプロファイリングスポットです。普段、聞けないような話を引き出すチャンスでもあります。飲み会を躊躇してしまう人は、相手の興味や関心をヒアリングする場所、と考えて参加してみるといいかもしれません。



実際の活用事例

筆者が実際にこのノウハウを活用した事例をご紹介します。以前、優秀なエンジニアを表彰する、という表彰制度の審査を担当したことがありました。この制度では、各部門長が優秀なプロジェクトやエンジニアを推薦し、審査担当が役員にプレゼンを行い表彰の可否を決定する、というものです(複数の案件が表彰されます)。

ある顧客に関するサーバーの運用管理の標準化、という推薦案件がありました。現場のエンジニアに聞いても、「当たり前前の仕事を当たり前をやっただけです」と言うだけで、この標準化自体の素晴らしさがいまち伝わってきません。そこで、この案件を下記の2つを中心にアピールすることにしました。

- ・この標準化は会社の生産管理部門でも標準方式としての採用を検討している。
- ・運用という、地味だが重要な仕事に、会社が注目しているという会社のメッセージを出すことができる。

ここで考えたのは、役員の興味・関心です。地道に頑張っている人間にスポットライトを当てたい、という関心は強い傾向にあり、標準化への努力を表彰したいという意向は強いものでした。ただし、大したことない案件を表彰すると、表彰制度自体の価値が低下することをリスクとも考えています。

そのため、何らかの客観的な「お墨付き」がある方が表彰をしやすいのです。

この2つの方向性を軸にプレゼンを行い、この案件は表彰されました。このように、自分の成果を評価する人間にしっかりと届けることは、自分自身のメリットになるだけではありません。同じ道を進んでくる後進のためにもなりますし、自分が関わる業界や分野の発展にもつながっていくのです。

全く同じものでも伝え方によって評価のされ方が異なってきます。そして、それはちょっとした工夫で変わるものです。次回以降も、「ちょっとした工夫で成果を出すための技術」をお届けしていきたいと思います。



IT・マーケティングが世界を救う!
特集

ユニケーゼとは、ソフトウェアにおける一つの型である。
そこにはどんな作法があり、どんな理があるのか。
これまで口伝によって伝えられてきた作法を今、文章として残す。
今回は、ソート作業を用いずにデータを管理する術を紹介する。



作法その十四、順番は、揃えるよりも、乱さぬこと

今年は、年明け早々の1月5日、国際地球回転・基準系事業（IERS）が、日本時間の今年7月1日午前9時前に閏秒を挿入することを発表しました。閏秒は、様々な要因により変動する地球の自転周期に合わせて時刻を調整するためのものです。しかし、今のところどのくらい調整すればよいかを長期予測することができないことが最大の問題になっています。

つまり調整量を規則化することができず、調整を実施する都度例外的な対応を取らざるを得ないのです。ゆえに社会的影響が大きいと、今の閏秒調整制度は廃止すべきという意見もあります。

しかし閏秒に限らず、こういった例外的対応を迫られる場面は日常生活においていくらでもありますし、システム開発においても同様です。ユニケーゼ開発手法やそこで用いられるTukubai コマンドは、現場から要求される様々な例外的な扱いに柔軟に 대응できるように発達してきた経緯があるそうです。今回は、そういった中で確立された一つの作法を紹介します。

データを分割管理しないという例外

本連載では前回、データ（特にマスターデータ）はデータの種類ごとに分割して管理すべきであるという作法を紹介しました。

例えば、駅IDから駅名、路線IDから路線名、車両IDから車両名というように、内部コードから名称に変換するためのマスターデータがあるなら、それらを一つのマスターファイルにせずに駅名マスター、路線名マスター、車両マスターというように分割して扱う

方がよい。なぜなら、その方がファイルの規模が小さくなってファイル読み込みや sort といった操作が軽くなるなど様々な利点があるから、というものでした。

ところが、この作法に素直に従えばよいとは言えない「例外」も存在します。



分割できないデータをどう扱うか

思うように分割管理ができない事例というものを考えてみます。例えば、全国展開している小売チェーン店の勤怠管理システムを開発してもらいたいという要求がなされたとしましょう。具体的には、誰が、いつ、どこで、何時間働いたかという情報を管理し、給与計算や各種分析に活用したいというわけです。

全国チェーン店の勤怠データ表

与えられた情報をいかに整理整頓するかが、実用的なシステムになり得るかどうかの重要な鍵を握っています。そこでまず、データをどのように格納すべきかを考えてみます。

今は冒頭に記した要求を素直に実現するため、次の5つの情報から構成される表を設計することにしました。

列番号	項目名	凡例	整列順位
1	従業員番号	123456	1
2	都道府県番号	47	3
3	店舗番号	1234	
4	日付	20150130	2
5	勤務時間	6	

都道府県単位の情報も欲しいという要望で、都道府県番号の列を設けています。

ところでデータの規模はというと、この業者は全国チェーン店であり、従業員はアルバイトまで含めて数万人います。その数万人が毎日仕事をしますので、データは毎日数万行ずつ増えていき、一年経てば一千万行近くにも達します。これは膨大な量で、仮に sort が必要になったら行数に対して比例以上の計算量を要しますので分割管理したいところです。しかし、そう簡単にいかない事情があったのです。

通常、同じ従業員は同じ店舗で働きます。ところが、新規出店で現地従業員が足りない時や繁忙期で一時的に特定地域で人員が必要になる時、或いは災害によって急遽人員が必要になるなど、他県への応援勤務という「例外」があるというのです。こうなると、一人の給与計算をする場合でも結局全国のデータが必要になり、都道府県毎の分割管理は得策にはなりません。では毎月に分けるのはどうかというと、賞与計算その他の理由により、これも得策にはならないというのです。

とにかく何らかの例外的理由により、分割管理が向かない事例だったとします。このような場合でも役に立つ作法が今回紹介する作法なのです。

sort 作業を生じさせない

情報は、何らかの規則による順番で並んでいることに意味があります。究極的には、規則的な並びこそが情報の正体と言えるでしょう。sort 作業は、そのよう

な状態を作りだすために重要です。

しかし自明な事ですが、既に意図したとおりの順番で並んでいるのであれば sort 作業は不要です。そして、このように整列された状態であるなら、そこから一部の情報を取り去ったとしても尚、整列状態であることはやはり自明です。反対に、新たな要素を加える時でも、正しい位置を探して要素を挿入すれば、複雑な sort 作業は発生しません。

書店の本棚を思い浮かべてみてください。小説の棚などは大抵著者名順に並べてあります。ある著者がロングセラー小説の改訂版を出したとしましょう。店員はその著者の小説の旧版がある位置を探し、その場所に新刊を差しますが、他の本を棚から引っ張り出して並べ替えをするわけではありません。もともと整列されているので位置を探し出すのも容易です。

そして同時に、今探し出した位置にある旧版を抜き取ります。当然ですが、この時も並べ替えという作業は必要ありません。

このように、始めから整列状態にあれば、大掛かりな sort という作業は不要なのです。

勤怠データ表の編集

今の考え方に基づいて勤怠データ表を編集すると図1のようになります。この図は sort 作業なしで勤怠データを編集する方法を示しています。

まず図1のように、勤怠データ表を三段階のレベルに分けて考えることにします。ただしどの段階の表

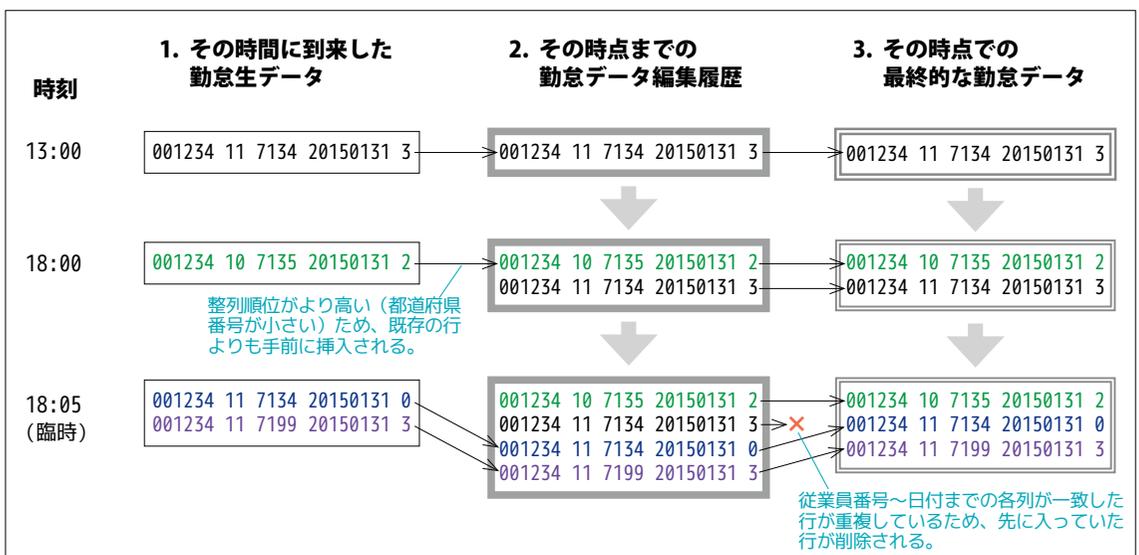


図1. 三段階のレベルで管理された勤怠データの推移

も先程示した5つの項目からなる行で構成されています。

左表は生データで、各店舗から各従業員が勤務を終える都度送られてくるものです。中央表は履歴データで、先程の生データを追記していったものです。ただし、従業員番号順→日付順で常に整列状態であるものとします。右表は結果データで、[従業員, 都道府県, 店舗, 日にち]で一意となるよう、先程の履歴データにおいて、その組み合わせが複数あれば最後のものだけを抽出したものです。

そもそも[従業員, 都道府県, 店舗, 日にち]が一意にならない場合があるのかという疑問が湧きますが、この事例では生データは従業員の申告に基づいて作成されるということで間違うことがあり、同じ組み合わせの訂正データを送ってくる場合があるそうです。

さて、この図では、2015/01/31に特別な働き方をしたある従業員Aさんから送られた勤怠データに基づく変化を示しています。この日、Aさんは午前中の3時間、応援で別県の店舗で働き、午後は自分の店舗に戻ってまた3時間働きました。その記録を、13時と18時に送信してきたのですが、臨時対応だったこともあってAさんは午前中の店舗番号を間違っていたことに気づきました。そこで18時5分、間違った店舗での労働時間を0、正しい店舗での労働時間を3と訂正したデータを送ったという筋書きです。

書店の例えで言うなら左表は続々と到来する改訂版で、中央表はそれを棚差した状態、右表は旧版を抜き取って最新版だけにした状態ということになります。とにかく左表から右表までがsortという作業無しで作れていることがわかります。これが、sort作業無しに整列済データを管理する方法です。



sortしない 整列済データ管理の実際

それでは、今説明した考え方を実践するコードやコマンド、他の例を挙げながら紹介します。

お客様センターの会員情報

あるショッピングサイトのお客様センターでは、顧客一人一人の会員情報を、いつ、どの手段で編集したかを全て記録した履歴データ表がありました。この表は各列の意味は次のとおりです。

列番号	項目名	凡例	整列順位
1	会員番号	654321	1
2	氏名	山田太郎	
3	住所	港区	
4	更新日	20150225	2
5	入力元	SYSTEM	3

整列順は、最初に「会員番号」、次に「更新日」、その次に5列目の「入力元」とします。「入力元」とは、編集に用いた手段を意味しており、具体的には次のどれかが入ります。

優先度	値名	意味
1	SYSTEM	郵便番号変更や市町村合併対応のための一括修正
2	TEL	お客様対応電話窓口での修正受付
3	MAIL	お客様からのeメールによる修正受付
4	PAPER	お客様からの郵送による修正受付
5	MYPAGE	お客様自身によるマイページでの修正

優先度とは、同じ日に同じ会員に対して修正要求があった場合にどれを優先するかの順位です。例えば同じ日に電話とお客様ページからの修正を受け付けた場合は電話を優先する、といった具合です。

このような表を作っておけば、例えば特定の会員の行を抽出することでオペレーターが、その会員の情報変更履歴や、最新の情報を確認することができます。

お客様センターの会員情報

この仕様に基づいて表を更新するプログラムが**リスト1**です。毎晩決まった時間に動かされることが前提になっています。

まず、取り扱うデータファイルについて一通り説明しておきます。31行目の\$RIREKI_ydayは前項で説明した履歴表(更新前)ファイル、32~36行目にある\$tmp-from-*という一時ファイルは、その日に送られてきた新しい修正(或いは追加)会員情報で、これらが入力データです。一方37行目の\$RIREKI_todayが本日更新される新たな履歴表ファイル、38行目の\$CUSTINFOが本日現在の最新会員情報ファイルで、これらが出力データです。

ではプログラムの動作を説明します。元の履歴データ表ファイル\$RIREKI_ydayは**リスト2**のような内容だったとします。これが1行目のcatによってパイプに流されます。そして本日、郵送(PAPER)と電話(TEL)の双方から会員番号000001さんと000003さんの住所変更受付(**リスト3**、**リスト4**)があり、それらの修正情報が33行目と35行目で追加されます。

リスト 1. sort 作業なしの会員情報データ更新プログラム

```

:
31 cat $RIREKI_yday
32 up3 key=1 - $tmp-from-mypage
33 up3 key=1 - $tmp-from-paper
34 up3 key=1 - $tmp-from-mail
35 up3 key=1 - $tmp-from-tel
36 up3 key=1 - $tmp-from-system
37 tee $RIREKI_today
38 getlast 1 1 > $CUSTINFO
:

```

up3 コマンド:
第2引数で指定されたテキストファイル (master 表) の行間に、第3引数で指定されたテキストファイル (tran 表) を挿入する。挿入位置は、“key” オプションで指定された列同士の値の大小比較をして決める。(同じ値の行については、master 表の行の後に tran 表の行を挿入する)

getlast コマンド:
重複する行があれば、それらの中で最後の行のみを出力する。ただし、uniq コマンドのように行全体を比較するのではなく、第1引数~第2引数で示された列番号の範囲での一致のみ見る。

※両コマンドともに、sort 済 (整列済) のデータを与えなければならない。

ただし履歴表の最後尾に追加されるわけではありません。up3 というコマンドが登場しますが、これが先程書店員の例で説明した「改訂本を適切な位置に入れる」作業を担います。“key=1” というオプションによって1列目 (会員情報) が一致する行を探すように指定されており、\$tmp-from-* から来る各行は会員番号を照らし合わせて適切な位置に追加されます。その結果、37 行目を流れている時点 (\$RIREKI_today にも出力される) の内容が **リスト 5** です。会員番号 000001 さんと 000003 さんの行が複数あります。

38 行目に getlast というコマンドが登場します。同じく書店員の例に例えるなら「最新版だけ残して旧版を取り除く」という作業を担うものです。これによって各会員の最終行以外が取り除かれますので結果として **リスト 6** (\$CUSTINFO) の内容ができあがります。

up3 も getlast も、与えるデータは sort 済 (整列済) でなければならないのですが、元々整列済であるため、sort コマンドはどこにもありません。しかもこのプログラムで作った第3の整列順である入力元列は、MYPAGE → PAPER → MAIL → TEL → SYSTEM という順番ですが、これは sort コマンドでは簡単にできない芸当です。

このように、データを常に整列された状態で管理すれば、処理を軽くでき、かつ例外的要求にも対応しやすくなります。それを助けてくれる Tukubai コマンドが、up3 や getlast というわけです。



頼りになる人の机は いつも綺麗

普段めったに使わない道具が急に必要になったが、しまったはずの机をいくら探しても見つからない！そうやって焦っていると、困った時いつも頼りになる上司がスッと自分のを貸してくれた。でも同時に「君は

リスト 2. 昨日時点の情報登録履歴データ (\$RIREKI_yday)

```

000001 榎取美和子 萩市 18430907 SYSTEM
000002 久坂玄瑞 萩市 18400501 SYSTEM
000003 榎取素彦 防府市 18290418 SYSTEM
000004 吉田寅次郎 下関市 18300920 SYSTEM

```

リスト 3. 本日届いた郵送による登録要求 (\$tmp-from-paper)

```

000001 榎取美和子 山口市 20140201 PAPER
000003 榎取素彦 山口市 20140201 PAPER

```

リスト 4. 本日あった電話による登録要求 (\$tmp-tel-paper)

```

000001 榎取美和子 防府市 20140201 TEL
000003 榎取素彦 防府市 20140201 TEL

```

リスト 5. 本日分を加えた情報登録履歴データ (\$RIREKI_today)

```

000001 榎取美和子 萩市 18430907 SYSTEM
000001 榎取美和子 山口市 20140201 PAPER
000001 榎取美和子 防府市 20140201 TEL
000002 久坂玄瑞 萩市 18400501 SYSTEM
000003 榎取素彦 山口市 20140201 PAPER
000003 榎取素彦 防府市 18290418 SYSTEM
000003 榎取素彦 防府市 20140201 TEL
000004 吉田寅次郎 下関市 18300920 SYSTEM

```

リスト 6. 本日時点での最新の会員情報 (\$CUSTINFO)

```

000001 榎取美和子 防府市 20140201 TEL
000002 久坂玄瑞 萩市 18400501 SYSTEM
000003 榎取素彦 防府市 20140201 TEL
000004 吉田寅次郎 下関市 18300920 SYSTEM

```

日頃から整理整頓をしないから……」と耳の痛い一言を言われた。このような経験はないでしょうか。

今回紹介した作法はこの話と同様に、日頃からの整理整頓の大切さを説いています。整理整頓は清潔にするためだけではなく、急な用事や想定外の用事にも咄嗟に対応できるようにするために重要なのです。

データも作業机も、日頃からの整理整頓を心掛けておきたいものですね。



HANDS LAB

ユニケージ®エンジニア数 **最大級!**

ハンズラボはユニケージ®開発手法に特化したITソリューション企業です。東急ハンズの営業システムを刷新したノウハウを駆使し、小売業における「オーダーメイド」のシステム開発を行います。

中途採用 大募集! 詳しくはHPで!

> <http://www.hands-lab.com/>

「ユニケージ®」は有限会社ユニバーサル・シェル・プログラミング研究所の登録商標です。

これであなたもユニケージエンジニア!

ユニケージ開発手法教育講座

「ユニケージ開発手法教育講座」は、ユニケージ開発手法におけるデータ管理の方法や、オリジナルコマンドの使用方法などをハンズオン形式で具体的に学べる講座です。UNIXの基礎からユニケージ開発手法による開発プロジェクトの進め方まで、ユニケージエンジニアとしてのトータルスキルを習得できます。

<http://www.usp-lab.com/LECTURE/CGI/LECTURE.CGI>



続々と新講座も充実中!

- K-BASIC** ユニケージ基礎編
- K-WEB** WEBアプリケーション編
- K-BATCH** バッチ処理編 (ウェブアプリケーション処理)
- K-ARCH** ユニケージアーキテクチャ編
- K-SETUP** ユニケージ開発環境セットアップ編
- K-UNYO** システム運用・管理編
- K-PROJECT** プロジェクトマネジメント・人材育成編
- K-SQL** 速習: SQLからの移行編
- K-STAT** ユニケージにおける統計コマンド編

UNIX 初心者のための講座などもご用意しています。

TechLION
For Independent Engineers

技術の草原で百獣の王を目指す
エンジニアたちの新感覚トークライブ!

<http://techlion.jp/>

上記のサービス内容は2015年2月現在のものです。最新の情報はホームページにてご確認ください。

ISBN 978-4-904807-17-0
C3455 ¥500



9784904807170

USP 研究所
定価 (本体 500円+税)



1923455005002