

コンピュータの各パーツは、自分が熱を発するくせに熱に弱いという、理不尽な性質を持っています。スマホを充電しながら重たいアプリケーションを動かさない方がいいよ、というアドバイスを耳にしたことはありませんか。充電による熱と、アプリケーション利用による熱の二重苦が**電池の劣化を早めてしまう**のです。

そこで、データセンターでは冷却を行うのですが、この冷却費がばかになりません。マイクロソフトは自社のデータセンター（移動やメンテナンスが簡単なようにコンピュータをセンターに詰め込んだもの）を**海中に沈めたり**、**グーグルは山中に雨ざらしにしたり**しています。こういったことをする理由はいろいろありますが、自然環境が勝手に冷却してくれるからという側面が最も大きいです。

また、巨大テック企業が各国にデータセンターを設置するのは、先述したリスクの分散・回避の他に、単純に「拠点のどこかが夜になるから」という理由もあります。

夜になれば当然冷えますから、自らはあまり冷却しなくていいわけです。極端な話をすれば、**夜になったらデータセンターを稼働し始め、朝を迎えるとともに、これから夜を迎える場所にあるデータセンターに稼働を切り替えていけば、冷却費を減らすことができます。**

これはデータセンターにとっては死活問題ですので、なるべく電力料金の低廉な国や地域にデータセンターを作ろうとします。国の側も、データセンターに対する電力のディスカウントなどを提示して誘致を図るのです。

分散処理の中で、役割を分割せず、すべてのコンピュータが同じように（平等に）振る舞う形態は P2P（peer to peer）です。古い時代のファイル共有システムである Winny や、近年ではビットコインが P2P システムを採用しています。

P2P システムの長所は、参加しているすべてのコンピュータが平等に振る舞っているため、どこかのコンピュータが故障しても、他のコンピュータによる代替が利くことです。そのため、対障害性能が高いしくみといわれています。

また、管理者がいないため、たとえば誰か 1 人の管理者がその強権を発動して不正を行ったり、多くの利用者が望まないシステム改変を行ったりするのが難しいということも長所と考えられています。

これらの点は、ひっくり返せばそのまま短所にもなり得ます。管理者や管理マシンがあるわけではないので、「もうこのシステムは終わりにしよう」と思っても、継続を望む利用者がいれば稼働し続けることでしょう。不正ファイルの温床と言われ続ける Winny が、開発者が亡くなった現在もいまだに現役であることから、この事実が導けます。

しかし、音楽や映画、雑誌に続いてソフトウェア販売でもサブスクリプション（定期購読）モデルが力を持つようになり、登場したのが Office365 である。たとえば、1年間の契約を結んで Office365 の Word2016 を使い始め、その期間中に Word2019 がリリースされるとそれに乗り換えることができる。しかし、1年経過後は何も使えなくなる。世界的にはサブスクリプションモデル主流になりつつある。

情報機器（ハードウェア）を動作させるためのソフトウェア

は、大まかに基本ソフトウェアと応用ソフトウェアに分けることができます。

基本ソフトウェアの具体的な例は、OS、ファームウェア、BIOS、デバイスドライバなどです。

**OS** の機能は、その情報機器を人間が楽に使えるようにすることです。「コンピュータ、ソフトなければただの箱」という名言がありますが、コンピュータだけがでんと目の前に置かれていても、人は使いようがありません。

キーボードからの入力を受け付けたり、それを画面に表示したり、マウスで操作ができるように周辺機器のやり取りをして、クリックできるようなアイコンを表示して……とたくさんのサポート機能が働いて、やっとコンピュータを「使える」ようになります。これらの機能を提供するのが **OS** です。

・・・

**ファームウェア** は、滅多なことでは書き換えたりしないソフトウェアのことを指します。変更しないのでソフトではなくファーム（かたい）のです。ソフトウェアはそれがたとえ **OS** であっても適宜変更していくことが可能ですが、たとえばエアコンや、冷蔵庫、炊飯器の制御ソフト、身近な情報機器でいえばルータの制御ソフトなどはほとんど書き換えません。機能強化やセキュリティ対策のために更新することはありますが、頻繁ではありません。

**BIOS**（Basic Input/Output System）は、ハードウェアの最も基本的な制御を担当するファームウェアです。PC で電源を投入し、最初に謎の黒画面やメーカー名が表示されるときに **BIOS** が動いています。**BIOS** を人間が操作することはほとんどなく、起動処理が済むと **OS** へ動作を引き継ぎます。

**デバイスドライバ** は、コンピュータに接続した周辺機器を動作させるためのソフトウェアです。

・・・

現在ではデバイスドライバの使い勝手はだいぶ向上し、利用者がデバイスドライバを意識しなくても使える **PnP**（Plug and Play：プラグアンドプレイ。つないだらすぐ使えること）や、（いまでは当たり前ですが）PC の電源を落とさなくても周辺機器の抜き差しができるホットプラグが常態化しました。

なお、PC 業界では、アプリケーションソフトウェアのことを一般的に「ソフト」と略していましたが、通信業界では「アプリ」と略すことがあり、スマホの普及に伴い、こちらの

呼称の方が一般化してきています。

Excel の黎明期から普及期に至る 1990 年代は、PC の浸透と 1 人 1 台化により、**EUC**（エンドユーザコンピューティング：末端利用者による主体的なシステム構築・運用）が推進された時期でもありました。

情報の専門部署である情報システム部門は、現場の業務をあまり知らないこともありますし、増え続ける一方のシステムや端末を持て余している時期でもありました。業務に精通し、人員潤沢な利用者部門が、簡単なシステムやプログラムであれば自分で作ってしまうことはニーズがあり、またそれを実現するツールも容易に入手できるようになってきていたのです。

EUC は、確かにこのとき優れた効果を発揮しました。

・・・

多くの EUC は、コンピュータに詳しい人が個人で作っていたため、その人が異動や退職でいなくなってしまうと、しくみがブラックボックス化してしまったのです。

・・・

消費税率が変更になるからプログラムを書き換えよう、元号が変わるからシステムを改修しなくては、と思っても、誰も VBA で書かれたプログラムの意味が理解できず、脆弱性があるのに修正なしで放置されたり、税率の変化に対応できずシステムごと葬られるようになったりします。そして、情報システム部門が完全に掌握・管理しようとする、結局、最初からシステム部門主体で作っておけばよかったと思うほどの作業が発生するわけです。

こうした経緯があるため、先端的な企業では「Excel 外し」の動きが加速しているのです。Excel が前時代的なシステムの象徴になっているわけです。代わりに使われるのは、ERP や BI です。

ERP（Enterprise Resource Planning：企業資源計画・統合業務支援システム

BI（ビジネスインテリジェンス）

IT 機器の導入にしても、IT 研修を受ければ自動的に効果が上がるといったものではありません。社内で開発部門と利用部門が密接に連携している環境があつて、初めて導入の効果やその大きさを理解することができます。

作る人と使う人が協力し合っているということは、当たり前のように見えて、実はあまり見られないことであるのが実情です。利用部門、つまりシステムを使い、業務を行い、儲ける人のことにまったく興味がなく、どんな仕事が行なわれているのか知らない開発部門がルールや IT 機器、すなわちシステムを作り、業務ルールの意味や IT 機器の効果をよく理解していない人が、与えられたシステムを言われるままに使ったり、最悪のケースでは与えられたシステムを使わずに回避するなどは、日常茶飯事です。

**IT 化はシステム化の中で必要**

に迫られてやる部分業務で、それが目的ではありません。当然、「ここは IT 機器に頼るよ

り、人手で置き換えてしまった方がコスト効率もいい」と思ったら、IT 機器の削除も行います。

一定規模以上のサイズになった企業、組織は専門分化が進み、各部署がブラックボックス化します。これは何十年も前から言われていることです。だから監査などを行い、巨大な組織やシステムを可視化していくわけですが、容易な作業ではありません。

ブラックボックス化、あるいはそれに対する監査というと、不正や非効率が行われているイメージになってしまいますが、ほとんどの組織は凡帳面に仕事をこなし、効率化も考えています。でも、それが重層的に構築された部署内での効率化や最適化にとどまってしまうのです。

複雑で膨大な業務のしくみや手順は、ERP という共通のプラットフォームに乗せることで可視化できます。これでやっと、「組織の全部の仕事が見えるようになる」のです。当たり前前的ようですが、いまの企業や社会は、そうしなければ可視化できないほど複雑なシステムと化しています。

可視化することで、それまで組織内の個別最適にとどまっていた改善や効率化の範囲を全社、あるいは子会社や協力企業を含めたサプライチェーン全体に拡張することもできます。

スクラッチとカスタマイズという言葉は、業務改善やシステム化のコンテキストで登場する場合、スクラッチは一から作ること、カスタマイズは IT 機器（ハードウェアやソフトウェア）に個別修正を加えることを指します。

それ自体は「ふーん」といった話なのですが、システム化や BPR の文脈でこれらの用語が出てきたときは、否定的な意味を持つ場合もあることを押さえておきましょう。

**DX**（デジタルトランスフォーメーション）にも触れておきましょう。2017 年頃から勃興したバズワードで、定着するかどうかはまだ先が見えていません

・・・

いかにもバズワードらしく、みんなが自分のいいように使うので、極めて曖昧模糊とした用語になっています。

セキュリティにおいて管理者の重要な仕事は、細かい、あるいは**最先端のセキュリティ技術に通曉することではなく、受容水準を決めることです。**

受容水準を適切に確定させることができれば、それを維持するための技術的な方策を、下の技術者や外部のセキュリティ企業は検討し始めることができます。

発達障害のあるお子さんに ABA（応用行動分析学）という療育が行われることがあります。これは一種の**強化学習**です。たとえば、自閉スペクトラム症の子どもはなかなか言葉が出ず、お菓子が欲しいときに叫んだりします。このとき、叫んでもお菓子は与えない、「お菓子」と言うことができればお菓子を与える、言葉が出なければジェスチャーを促し、うまくジェスチャーできたらやはりお菓子を与える。こうした練習を繰り返します。

すると、叫んでもお菓子はもらえない（報酬なし）、ジェスチャーをすればお菓子もらえる（報酬あり）ことを学習して、報酬がもらえる行動を取ることが多くなって（強化されて）いきます。

**自動運転車を積極的に騙す手法**もいくつかわかっています。

たとえば、自動運転車は画像診断を利用して車線を認識していますが、ステッカーなどを使って誤認識を起こさせ、反対車線へ誘導する手法などが知られています。

こうした手法は**敵対的機械学習**と呼ばれていて、たとえば 2016 年にマイクロソフトが公開した会話ボットの「Tay」は、利用者によってあつという間に人種差別や性差別を学習させられてしまい停止に追い込まれました。

レインボーテーブル

~~~~~

レインボーテーブルは、パスワードのハッシュ値から元のパスワードを割り出すために使うテーブルの一種。

~~~~~

**C 言語**は通好みの汎用性

と効率性の高い言語ですが、その一方で**習得が難しく**、同じ仕事をさせるにも無限の書き方があるため人が書いたプログラムを読み解くのに苦労します。

**Java** は特定の OS に依存しない使いやすさがありますが、それを担保するために仮想マシンを使っているので、実行速度で他の言語に後れを取ります。

最近、AI を活用したプログラムによく利用されている **Python** は可読性が高く（人間にとって読みやすい）、初心者にも使いやすいのにプロも満足する高機能が特徴です。しかし、大抵の言語でそうなのですが、「使いやすい」→「曖昧さを許容している」ので、思っていたのと違う動作をすることがあります。

統計処理を得意とする **R** も人気が高まっています。統計解析を行うのであれば、簡単なプログラムで高度な解析処理が可能ですが、他の用途に転用しにくく汎用性はありません。ちなみに、先に例として挙げたプログラムは **C 言語** で書かれています。

初級のプログラミング教室でよく使われる**マインクラフト**という教育アプリがあります。レゴのブロックのようなものを仮想空間に積み上げ、世界を作り、そこでキャラクターを動かすことができます。単にブロック遊びをするだけでなく、ブロックを作って論理回路（ピ

タゴラスイッチを想像してください) を構築することができるため、教育効果が期待されています。もともとゲームのデザイン、インターフェースですから、楽しいのはある意味で当たり前だともいえます。

簡便で使いやすい BI ツールの代表例を紹介しておきましょう。この分野のアプリケーションはいまやレッドオーシャンになっていますが、頭一つ抜き出ているのは **Tableau (タブロー)** です。

直感的な操作だけで回帰分析や相関分析を行い、手持ちのデータから未来予測などを行うことが可能です。画面上で何回でも試行錯誤できます。もちろん、使いやすいからといって、何の考えもなく使えるわけではないのは前述した通りです。統計手法はたくさんあるので、誤った使い方をすればいかようにも結果を導くことができます。

故意に行うのは論外ですが、総当たりの試行錯誤の結果、「**誤った、でも見た目は説得力のあるデータ**」は作れてしまいます。その陥穿にはまらないデータサイエンスの基礎力が必要です。

高度 IT 人材も、先端 IT 人材も、情報セキュリティ人材も、既存の業務のしくみの延長線上にあり、効率化が主目的だった時代を離れ、情報技術によって新たな価値を創造していかなければならない現代、そして未来において欠くことのできない高付加価値人材と位置づけられています。