

■はじめに

みなさん、こんにちは。この7月に入会した株式会社ワタナベジョブズ 代表取締役の渡辺幸生です。

昨年、還暦を迎え大手製造業を定年退職し、妻と立ち上げた会社の代表取締役に就任しました。事業内容は、妻が主体の『障害児福祉・教育事業』と私の『IT/DX 関連事業』の二本立てです。

前職ではIT関連の仕事（システム開発、導入、運用、保守）を主に業務部門でしておりました。私の今年20歳になった一番下の子が重度障害なので仕事と家庭の両立を考えた場合、パソコンとネット環境があれば場所、時間に捉われない『IT関連の仕事』で最後まで務めることが出来、前職の会社とITの仕事に感謝しております。

このようなことから少しでも、今までの経験がお役に立てばという思いで『IT/DX 関連事業』を会社の一つの柱としています。

特に前職のような大手老舗製造業のようなところでは、メインフレームコンピュータ利用が始まった1960年代からのシステムが未だ基幹系システムの一角を占め、多くの問題を現場でリアルに体験しましたのでシニア世代として何か出来ないか考えていました。そして、これら業務やシステムの要はデータマネジメントであると確信していたところ JDMC と出会う機会を得、このレガシー基幹系システムの『2025年の崖』問題を回避するために出来ることを少しでも学びたい思いです。

1. コンピュータとの出会い

ジャパン・アズ・ナンバーワンの時代、理系学生生活を送りまだPCが世に出る前のキーパンチャー*1 という職業が存在していたころ、最初のプログラム作成は紙のカードやテープに穴を開けマシン語やアセンブラ言語のプログラミングで、その後FORTRAN、COBOL、PASCAL、PL/1も経験しましたが、入出が紙ベース、一台のコンソールディスプレイ付端末を順番待ちで使うなど、UIは今から考えると劣悪の極みです。ですから、学生時代の大半は、出来るだけコンピュータからは離れたいとの思いが強かったものです。

しかし卒業間近な頃、パソコンに触れる機会を得、紙、磁気テープ、を持ち運び、コンソール端末へ移動するなど、の肉体労働的作業から解放され、指示すればすぐに結果が返りパーソナルな知的サポートマシンとしての可能性を感じるなど、パソコンの将来性に大いに感激しました。

しかし、現実の世の中のコンピュータシステムは、UIの悪いメインフレーム、その小型のオフコンが主流で魅力は感じられず、また今でいうディープラーニングのニューラルネットワークを使ったパターン認識のシミュレーションを卒業研究とするも、AIの実用化はまだまだ何十年も先のことであろうと肌で実感しましたので、仕事としては当面コンピュータとは距離を置こうと考え、一般の製造業へ就職しました。

2. 仕事の中でのコンピュータ～システム刷新はデータマネジメントから

1980年代の初め頃、一般製造業へ就職したのですが業務のシステム化は既に必須事項です。一般の業務の中でシステム関連では新規開発や変更保守をメインフレームやパソコンなどで行い、エンタープライズシステム概念もなく、業務部門、関連子会社の中での部分最適化のようなことをしていたのです。全国の現場の中へ直接入り込んでの、システム開発・支援なのでよいシステムならばダイレクトに喜んでもらいそれが嬉しくもありました。

しかし、はじめに述べたように家族の都合で、在宅での時間が多く必要になり、仕事と家庭の両立を考えた場合、IT系の仕事なら在宅勤務などと相性がよいので、よりIT系に特化し、基幹系のメインフレーム関連ならば当面の安定的な業務ニーズがあると考え、この仕事へ変更しました。基幹系メインフレームシステムは、考えていたようにブラックボックス化し使いづらく、何が起きているか、すぐにわからないなど問題満載なので、まずはデータの見える化、データ分析基盤構築をしていきました。

このときデータモデリングの手法を外部コンサルさんから学ぶ機会を得、データマネジメントの重要性を理解する有意義な経験でした。そして、このデータモデリング、データ分析基盤構築の先は、本命の40年以上使っている老朽化基幹系システム刷新なのですが、かなりの難題で努力はしましたが、ここで残念ながら退職です。

この難題の解決は、業務、システムの両面での具体的なリスク、問題、課題を明確にした上で、この国の産業全体を俯瞰した今とは違う業務が描け、経営トップがそれを理解実行するかだと思えます。

3. 社会の中でのコンピュータ～予想外の展開

80年代初期にパソコンに触れ、世の中（個人の生活、社会）を大きく変え得る道具となるだろうと確信しました。そして、それから約10年後、東西冷戦が終結するという予想外の展開、それとともにインターネットが解放され劇的な情報革命に遭遇。毎月何万円もかかる通信代で300bps、2400bpsでパソコン通信し文字のやりとりだけで感動していたのがもう化石時代のような感じです。

さらにこのインターネット革命から約30年、ChatGPTの登場。40年前ニューラルネットワークでパターン認識のシミュレーション実験をしていた頃、AIの実用化は生きている間にはないだろうと思っていました。せいぜい画像認識（顔認証は無理）ぐらいかと。

私がこの現役時代に起きた予想外の発展（インターネット、生成AI）で、過去のIT資産を持たない後進国やスタートアップ・中小零細企業がいきなり最新のIT/DXでビジネスの中心に躍り出るリープフロッグ現象が起き、産業革命かさらにそれ以上に社会が変わりそうです。しかし、我が国の現実には行政府、老舗大企業などは既存システム、既得権益のしがらみなどでDXに頓挫している感じで歯がゆい思いです。

4. おわりに～『失われた30年』*2からの脱却への『道程』*3

ChatGPTをはじめとする生成AIの登場により、これまでも増して、大きく社会、産業の変革が求められている現在、企業の経済活動を支えるべく基幹系システムが古いままでよいのか。変革要求に対応出来ず『失われた30年』が更に延長か、滅びるか。

この懸念を払拭すべく、老朽化基幹システム刷新への『道程』を昔のコンピュータを知る最後の世代として残りの人生、少しでもよい方向へと歩んでいきたい思いです。肝は経営トップへどう腹落ちさせるか、ここはシニア・シルバー世代の人脈、経験を生かして行きたいです。

私と同世代の元マイクロソフトエンジニアの中島聡さん*4のように、大変革の今をエキサイティングの生きたいですね。

そして我が社のもう一つの妻の事業の発達障害児福祉学習において、生成AIの登場で教育のあり方、社会へ出てからの働き方が大きな転換期です。未来のある子どもたちが方向を見失わないよう、このJDMCで生成AIの活用事例、可能性なども学びたいと思っています。

みなさん、宜しくお願い致します。

【注】

*1 キーパンチャー

寅さん映画第一作（1969年公開）の寅さんの妹さくらさんの職業。彼女は大手町あたりに本社がある総合電機メーカー勤務で当時の先端OL。

*2 『失われた30年』

1990年代初頭のバブル経済崩壊から現在に至るまでの低成長時代。

一番の原因は、日本では何もしないでも、そこそこ社会、経済が安定しているため大きな変革を先送りしてしたため。その結果、気が付けば先進国下位クラスへ転落。

～未来予測士ユーチューバー 友村氏より

老朽化基幹系システム刷新もこれと同じ構造、気が付けば『2025年の崖』へ転落か。

*3 『道程』（詩人 高村光太郎作）

僕の前に道はない

僕の後ろに道は出来る

道は僕のふみしだいて来た足あとだ

だから

道の最端にいつでも僕は立つてゐる

…

*4 中島聡（元マイクロソフトエンジニア）さんの弁

ChatGPTの登場は多分3回目の大変革です。初めてパソコンをいじった時の感動。

それから、初めてインターネットに出会った時の感動。それと同じくらいのインパクトがあります。エンジニアの直感として、パソコンの誕生とかインターネットの誕生並みの大きさのインパクトだとひしひしと感じていて、もう興奮してしょうがないです。

■プロフィール

1983 年高専の情報系学科を卒業後、大手製造業へ就職。

主に事業部門やその関連会社で、業務部門側でシステム開発、運用・保守に従事。

退職前の約 10 年間は、レガシー系基幹システムのデータ分析基盤構築、BI ツールの導入・活用・社内展開を実施。2022 年退職。

2023 年、株式会社ワタナベジョブズ 代表取締役就任。

<http://www.watanabe-jobs.com>



■参考：パンチカードと 1980 年代初頭の OS 入門のテキスト（初版は 1972 年）

カード 1 枚（1 行）80 桁～実効入力桁数は 72 桁。

ハードスペックの制約で今では考えられないプログラミング。これが今に続くレガシーシステムの源流。

この写真の中のプログラムは、今の Z/OS マシンでも動く。



SYSTEMS PROGRAMMING

INTERNATIONAL STUDENT EDITION

Copyright © 1981
Exclusive rights by McGraw-Hill Kogakusha, Ltd. for
manufacture and export. This book cannot be re-exported
from the country to which it is consigned by McGraw-Hill.
12th Printing 1981

Copyright © 1972 by McGraw-Hill, Inc. All rights reserved.
No part of this publication may be reproduced, stored in a retrieval
system or transmitted, in any form or by any means, electronic,
mechanical, photocopying, recording, or otherwise,
without the prior written permission of the publisher.

Library of Congress Catalog Card Number 79-172263

When ordering this title use ISBN 0-07-085175-1

TOBHO PRINTING CO., LTD. TOKYO JAPAN

163

LOADERS

ESD cards

| Reference no. | Symbol | Type | Relative location | Length |
|---------------|--------|------|-------------------|--------|
| 1 | JOHN | SD | 0 | 64 |
| 2 | RESULT | LD | 52 | --- |
| 3 | SUM | ER | --- | --- |

TXT cards

| Reference no. | Relative location | Object code | Length |
|---------------|-------------------|-------------|-------------|
| 4 | 0 | BALR | 12,0 |
| 6 | 2 | ST | 14,54(0,12) |
| 7 | 6 | L | 1,46(0,12) |
| 8 | 10 | L | 15,58(0,12) |
| 9 | 14 | BALR | 14,15 |
| 10 | 18 | ST | 1,50(0,12) |
| 11 | 20 | L | 14,54(0,12) |
| 12 | 24 | BCR | 15,14 |
| 13 | 28 | 1 | 7 |
| 13 | 32 | 9 | 9 |
| 13 | 36 | 10 | 10 |
| 13 | 40 | 3 | 3 |
| 13 | 44 | 28 | 28 |
| 14 | 48 | 0 | 0 |
| 17 | 60 | | |

RLD cards

| Reference no. | Symbol | Flag | Length | Relative location |
|---------------|--------|------|--------|-------------------|
| 14 | JOHN | + | 4 | 48 |
| 17 | SUM | + | 4 | 60 |

Object deck

FIGURE 5.8 Example object deck for a direct-linking loader

12

Monitor is a term that refers to the control programs of an operating system. Typically, in a batch system the jobs are stacked in a card reader, and the monitor system sequentially processes each job. A job may consist of several separate programs to be executed sequentially, each individual program being called a *job step*. In a *batch monitor system* the user communicates with the system by way of a control language. In a simple batch monitor system we have an execution card may be in the following format:

```
// step name EXEC name of program to be executed, Argument 1, Argument 2
```

The job control card, a definition card, may take on the following format:

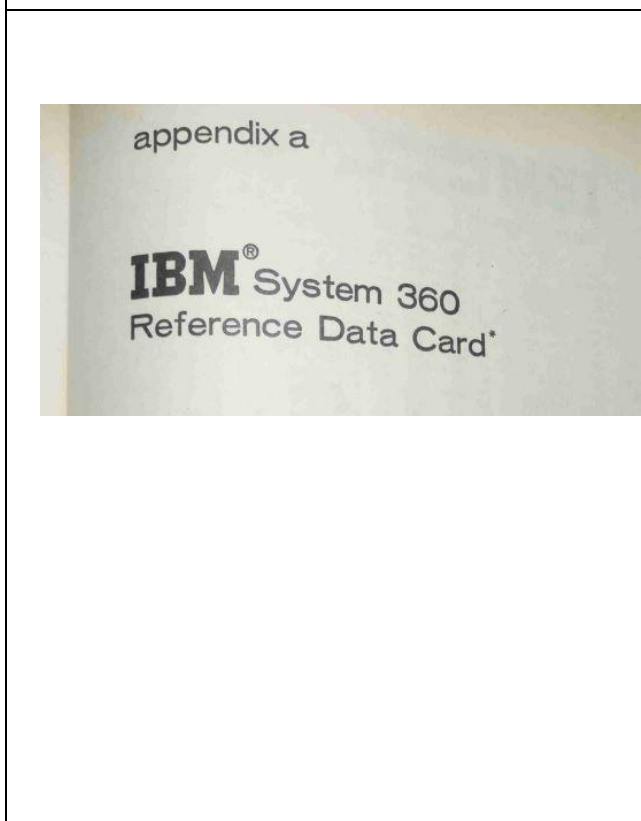
```
// job name JOB (User name, identification, expected time use, lines to be printed out, expected number of cards to be printed out)
```

Usually there is an end-of-file card, whose format might consist of /*, signifying the termination of a collection of data. Let us take the following example of a FORTRAN job.

```
//EXAMPLE JOB DONOVAN, T168,1,100,0
//STEP1 EXEC FORTRAN, NOPUNCH
READ 9100,N
DO 100 I = 1,N
  I2 = I*I
  I3 = I*I*I
100 PRINT 9100, I, I2, I3
9100 FORMAT (3I10)
END
/*
//STEP2 EXEC LOAD
/*
//STEP3 EXEC OBJECT
/*
10
/*
```

The first control card is an example of a definition card. We have defined the user to be Donovan. The system must set up an accounting file for the user, noting that he expects to use one minute of time, to output a hundred lines of output, and to punch no cards. The next control card, EXEC FORTRAN, NOPUNCH, is an example of an execution card; that is, the system is to execute the program FORTRAN, given one argument - NOPUNCH. This argument allows the monitor system to perform more efficiently; since no cards are to be punched, it need not utilize the punch routines. The data to the compiler is the FORTRAN program shown, terminated by an end-of-file card /*.

The next control card is another example of an execution card and in this



APPENDIX A

IBM System/360 Reference Data

MACHINE INSTRUCTIONS

| NAME | MNEMONIC | OP | FOR | OPERANDS |
|------------------------------|----------|----|-------|---------------------|
| Add (c) | AR | 1A | RR | R1,R2 |
| Add (c) | A | EA | RK | R1,D2(X2,B2) |
| Add Decimal (c,d) | AP | FA | SS | D1(L1,B1),D2(L2,B2) |
| Add Halfword (c) | AH | 4A | RK | R1,D2(X2,B2) |
| Add Logical (c) | ALR | 1E | RR | R1,R2 |
| Add Logical (d) | AL | 5E | RK | R1,D2(X2,B2) |
| AND (c) | N | 14 | RR | R1,R2 |
| AND (c) | N | 54 | RK | R1,D2(X2,B2) |
| AND (c) | N | 94 | SI | D1(B1),I2 |
| AND (c) | NC | D4 | SS | D1(L1,B1),D2(B2) |
| AND (c) | OS | RR | R1,R2 | |
| Branch and Link | BALR | 45 | RK | R1,D2(X2,B2) |
| Branch and Link | BAL | 0D | RR | R1,R2 |
| Branch and Store (e) | BASR | 4D | RK | R1,D2(X2,B2) |
| Branch and Store (e) | BAS | 07 | RR | R1,R2 |
| Branch on Condition | BCR | 07 | RR | R1,R2 |
| Branch on Condition | BC | 47 | RK | M1,D2(X2,B2) |
| Branch on Count | BCTR | 06 | RR | R1,R2 |
| Branch on Count | BCT | 46 | RK | R1,D2(X2,B2) |
| Branch on Index High | BXH | 80 | RS | R1,R3,D2(B2) |
| Branch on Index Low or Equal | BXLE | 87 | RS | R1,R3,D2(B2) |
| Compare (c) | CR | 19 | RR | R1,R2 |
| Compare (c) | C | 59 | RK | R1,D2(X2,B2) |
| Compare Decimal (c,d) | CP | F9 | SS | D1(L1,B1),D2(L2,B2) |
| Compare Halfword (c) | CH | 49 | RK | R1,D2(X2,B2) |
| Compare Logical (c) | CLR | 15 | RR | R1,R2 |
| Compare Logical (c) | CL | 55 | RK | R1,D2(X2,B2) |
| Compare Logical (c) | CLC | D5 | SS | D1(L1,B1),D2(B2) |
| Compare Logical (c) | CLI | 95 | SI | D1(B1),I2 |
| Convert to Binary | CVB | 4F | RK | R1,D2(X2,B2) |
| Convert to Decimal | CVD | 4E | RK | R1,D2(X2,B2) |
| Diagnostic (e) | D | 83 | SI | |
| Divide | DR | 1D | RR | R1,R2 |
| Divide | D | 5D | RK | R1,D2(X2,B2) |
| Divide Decimal (d) | DP | FD | SS | D1(L1,B1),D2(L2,B2) |
| Edit (c,d) | ED | DE | SS | D1(L1,B1),D2(B2) |
| Edit and Mark (c,d) | EDMK | DF | SS | D1(L1,B1),D2(B2) |
| Exclusive OR (c) | XR | 17 | RR | R1,R2 |
| Exclusive OR (c) | X | 57 | RK | R1,D2(X2,B2) |
| Exclusive OR (c) | XI | 97 | SI | D1(B1),I2 |
| Exclusive OR (e) | XC | D7 | SS | D1(L1,B1),D2(B2) |
| Execute | EX | 44 | RK | R1,D2(X2,B2) |
| Half (c,d) | HD | 96 | SI | D1(B1) |
| Insert Character | IC | 43 | RK | R1,D2(X2,B2) |
| Insert Storage Key (a,b) | ISK | 09 | RR | R1,R2 |
| Load | LR | 18 | RR | R1,R2 |
| Load | L | 58 | RK | R1,D2(X2,B2) |
| Load Address | LA | 41 | RK | R1,D2(X2,B2) |
| Load and Test (c) | LTR | 12 | RR | R1,R2 |
| Load Complement (c) | LCR | 13 | RR | R1,R2 |
| Load Halfword | LH | 4E | RK | R1,D2(X2,B2) |
| Load Multiple | LM | 98 | RS | R1,R3,D2(B2) |
| Load Multiple Control (e) | LMC | 88 | RS | R1,R3,D2(B2) |
| Load Negative (c) | LNR | 11 | RR | R1,R2 |
| Load Positive (c) | LPR | 10 | RR | R1,R2 |
| Load PGM (c,d) | LPGW | 82 | SI | D1(B1) |
| Load Real Address (c,e) | LRA | B1 | RK | R1,D2(X2,B2) |
| Move | MV | 92 | SI | D1(B1),I2 |
| Move Numerics | MVC | D2 | SS | D1(L1,B1),D2(B2) |
| Move with Offset | MVN | D1 | SS | D1(L1,B1),D2(B2) |
| Move Zeros | MVZ | D3 | SS | D1(L1,B1),D2(B2) |
| Multiply | MR | 1C | RR | R1,R2 |
| Multiply | M | 5C | RK | R1,D2(X2,B2) |
| Multiply Decimal (d) | MP | FC | SS | D1(L1,B1),D2(L2,B2) |
| Multiply Halfword | MH | 4C | RK | R1,D2(X2,B2) |
| OR (c) | OR | 16 | RR | R1,R2 |
| OR (c) | O | 56 | RK | R1,D2(X2,B2) |
| OR (c) | OI | 96 | SI | D1(B1),I2 |