

実務家の見たコンピュータ環境の変遷*

プログラムローダー から OS まで

松谷 泰行**
多摩大学経営情報学部

概要

1950年代から1960年代の始めまで、第三代コンピュータが出現するまでの間、プログラマ達がどのようにコンピュータを使っていたかを話すことにしたい。まだOSやソフトウェア開発環境といった言葉はなかったが、それに近い環境の必要性を感じ、いろいろな工夫がなされていた。私は今でもOSやソフトウェア開発環境はあの時の苦勞を解決するために生まれたものだと思っている。その実感を話すことにしたい。

Abstract

I tell my experiences of how I used computers as a programmer from 1958 to 1964. At that time there was no word of "operating systems". When we had problems for computer operation, we made many efforts to construct environment like OS. When the third generation computer was announced, it seemed that OS is the solutions to the problems we had.

1. はじめに

私はコンピュータを作ったこともなければ、OSやコンパイラなどのシステムソフトを作ったこともない。いわゆるアプリケーションシステムの構築に携わってきた。その意味で、わが国でコンピュータが使用され始めた1950年代の終頃から、第三代コンピュータとしてIBM

システム360が登場する1960年代半ば迄、プログラマ達がどのようにコンピュータを使っていたかを、私の経験を通じて話すこととしたい。残念ながら私の記憶に頼って話しをすることになる。間違いもあるかもしれぬが、それは全て私の責任であることをお断りするとともに、お許しを頂きたい。

2. 当時のコンピュータ状況

1950年代の半ば、国産コンピュータはまだ試作段階にあり、商品としてはIBMを始めとする外国製品の独壇場であった。コンピュータ製品としては、第一世代コンピュータの時代で、IBM650やIBM704がわが国にも輸入されていたが、導入していた企業はまれであった。「事務の機械化」に手をつけ始めた企業もあったが、いわゆるPCS（バン

* Advances of Computer Operating Environment in Early Days

** Hiroyuki Matsutani

School of Management & Information Sciences, Tama University

(C) 松谷泰行

チカードシステム) を使ったのシステム構築であった。私の在学した東京大学工学部応用物理学数理工学コース(現計数工学科)でも、計算機械第1・第2という講義があったが、第1はアナログコンピュータ、第2はPCSの講義で、プログラミングの話は行われていなかった。それでもこのような講義があるのは珍しい存在だった。

1958年、大学院に進んだ頃から、幸運にも、幾つかの国産コンピュータを使わせていただくことができた。

最初に使ったのは、電気試験所(現電総研)の作ったリレー式の計算機ETL Mark IIである。電子計算機としては日立のHIPAC101など国産メーカーの製品を幾つかと、東大工学部が作ったTACを使うことができた。これらはシステム環境ということからすると、プログラムローダーしかない計算機で、機械語によるプログラムであった。機械語といっても、ムネモニックコードに似た書き方が可能で感覚的にはマクロのないアセンブラーというところであった。

1960年に八幡製鐵(現新日鐵)に入社しPCSなどもいじったが、1961年から第2世代コンピュータのIBM7070と1401を使うことになった。これらはファイルとして磁気テープを使用しており(磁気ディスクが普及するのは第3世代からである)、アセンブラ言語にもマクロが使用されていた。まだオンラインシステムは、我々にとって夢の世界であり、バッチシステムとしての使い方にもいろいろな工夫が必要であった。例えば次のような問題である。

- (1) プログラムのテストをどうやるか
- (2) 日常のバッチジョブの効率的なやり方

これらの問題をある程度解決することはできたが、どこかで壁にぶつかった感

じがあった。そしてその時感じた問題点は、第3世代のコンピュータとOSの説明を聞いた時、全て解決されたように感じたことをよく覚えている。

現実のOS(例えば360/OS)は必ずしも使い易いものではなかったが、何故OSができたかということを考えると、第1世代第2世代のコンピュータでの使い勝手の悪さを改善するために生まれたものという感じが強い。この実感を伝えることができればよいがと思っている。

3. TACの使用経験

3.1 TACの特徴

TACは確か1959年から東大工学部で動き出したと記憶している。その仕様を正確に述べることはできぬが、私なりに記憶に残っている特徴は、次の通りである。

(1) ブラウン管をメモリに使った1024語(確か17ビット)のコンピュータ。ブラウン管だからビットの並びが写っているところが面白かった。

(2) 演算素子は真空管、先のブラウン管メモリと共に、大きな部屋を占拠しかつ猛烈な熱気を放出していた(クーラーもあったがなきが如くであった)。

(3) 命令体系はEDSACの第2版 [1] に浮動小数点演算命令が付け加わったものである。固定小数点演算については、加減乗除全ての命令があり、インデックスレジスタに当たるBレジスタも使うことができた。

(4) 命令語の体系はいわゆる1アドレス命令 対象番地の形であった。

この中でも最大の特徴はEDSACと同じ命令体系だったということだろう。

EDSACのイニシャルオーダ（プログラムローダ）がそのまま使え、その結果機械語のプログラミングでありながら次の書き方が可能であった。

- (1) 命令コードにA (Add) S (Subtract) などムネモニックコードの使用が可能。
- (2) 対象番地にはアドレス番地（数字）を記入。実際の番地の他に、プログラムの先頭アドレスを0番地として数えた相対番地の使用が可能。これがライブラリプログラムを作りやすくしていた。
- (3) 限られた英文字であったが、それをシンボリックアドレスのように使うことができた。

コーディングをした感覚からいうと、マクロのないアセンブラ言語といった感じである。そしてアセンブラを通すことなくプログラムをロードさせることができた。これらは全てイニシャルオーダの働きである。このイニシャルオーダの優秀さについては、高橋延匡先生が折にふれ力説されているので詳細は省略したい。

3. 2 森口流教育法

TACは広く多くの人に使ってもらうことが期待されていた。そのために、プログラムライブラリの整備やプログラミング教育とその普及を、森口繁一先生、高田勝先生、清水留三郎先生が行うことになり、筆者もそのお手伝いをする事となった。

最初のプログラム講習会はTAC開発者側の主催で行われ、我々もそれを受講した。受講した感想は残念ながらあまりいいものではなかった。内容的にはイニシャルオーダの説明がかなりのウエイトを占めており、例題があまりなかった。森口先生も含めた議論の中で、例題を中

心にしたテキストを作ろうではないかということになった。まず A, B 2つを入力して $S = A + B$, $D = A - B$, $P = A * B$, $Q = A / B$ を計算して出力することから始まり、幾つもの例題が用意された。例題に沿って学習してゆくことで自然と命令コードや文法を覚えてゆくという考え方である。例題の設定も、命令コードが理解し易い様に考えていった。森口先生がこの種のテキスト（TACの場合は例題集という感じだったが）を作るのは実は初めてではなかった。ただそれまでの経験も踏まえ、プログラム教育はこういうスタイルでいこうという感じになったのは、この頃からではなかったかと思う。

また、その頃、Wilksなどが記したEDSACの本（第2版 [1]）や、書名は忘れたがマクラッケンが記したプログラミングの本、IBMのテキスト FORTRAN Programmer's Primer（だったと思うが）なども手に入り、それらがいずれも例題中心でわかり易いことを、森口先生は高く評価されておられた。

当時メーカーが実施していたプログラミング教育は、命令コードを記した分厚いマニュアルを、始めから順番に話してゆくというスタイルで、ちっとも面白いものではなかった。この点からも、例題を中心とする森口式のやり方は、非常に新鮮で受講者の評判もよかった。例題を中心とする教育方法は、学校教育で現在ほぼ定着しているように思う。そのきっかけとなったのが、これらテキスト作りであったように思えてならない。

4. 第2世代のコンピュータ

4. 1 IBM7070, 1401

1960年、私は八幡製鐵（現新日鐵）に入社した。最初PCSをいじったりして

いたが、1961年に導入するIBM7070と1401を担当することになった。これらは、素子としてトランジスタ、メモリには磁気コアを用いた第2世代コンピュータで当時の最新鋭機械であった。

当時は技術計算用、事務計算用という区分があった時代で、7070は両者併用（汎用）、1401は事務用という位置づけである。技術計算用としては、八幡製鐵には導入されていなかったが、7090が有名であった（表1参照）。7070の用途は生産管理がターゲットで事務用のウェイトが高かった。7090も含め特徴を表2に記す。

7070では、アセンブラでプログラムを書いた。アセンブラといってもマクロジェネレータのついたもので、COBOLでいうMOVEに相当するマクロが始めから提供されていた。

4.2 磁気テープの読み書き

私がそれまで使った国産コンピュータには、磁気テープがついていなかった。磁気テープの読み書きをどうプログラムするのかは非常に興味があった。仕組みとしては現在と基本的に同じである。すなわち、まずレコード（論理レコード）がある。このレコードが幾つか集まってブロック（物理レコード）となる。このブロックがハードウェアの読み書きの単位になる。1ブロック当たりのレコードの数のことをブロック化係数という。磁気テープ上のブロックとブロックの間には隙間（IBG Inter Record Gap）があり、その長さは記録密度に比べはるかに大きい。ブロック化係数が小さいと、格納できるレコード数や1レコード当たりの読み書き時間が小さくなる。

表1 IBM7070, 1401の位置付け

	事務処理用	汎用	技術計算用
大型		7070	7090
中型	1410		
小型	1401		

表2 IBM7090, IBM7070, IBM1401の特徴

	1401	7070	7090
演算	10進演算	10進演算	2進演算
浮動小数点	なし	あり	あり
記憶	可変長語単位 4K桁前後	固定長語単位 10K語 (1語10進10桁)	固定長語単位 10K-32K (1語36ビット)
周辺装置	カードリーダー プリンタ 磁気テープ	カードリーダー プリンタ 磁気テープ	カードリーダー プリンタ 磁気テープ
言語	アセンブラ	アセンブラ FORTRAN	FORTRAN アセンブラ

主記憶装置と磁気テープ間のデータ転送はCPU (Central Processing Unit) の動作と平行して行われ、データ転送の終了あるいは異常の時割り込みが発生する。正直言ってこのあたりの仕組みはなかなかわからなかった。

実際のプログラムとなると、IBMからIOCS (Input Output Control System) というパッケージが提供されており、それを使えばよいということである。このIOCSは次の機能を持っており、OSのデータマネージメントに当たるものと言えよう。

- (1) レコードのブロッキングやデブロッキングの処理
- (2) エラー発生時のリトライ
- (3) 多重バッファを持つ事が出来、これにより入出力動作の平行度を上げる
- (4) プログラマが磁気テープを捜査するためのマクロ (OPEN, GET, PUT, CLOSE など) の提供
- (5) 我々は利用しなかったが、SPOOL という仕組みが提供されていた。これは多重プログラミングによるカード読み取りやプリンタによる印刷などの平行オペレーションを行うもので、現在のPCなどの世界でもこの言葉にぶつかる時がある。SPOOL という言葉の当時の意味は、Simultaneous Peripheral Operations On Lineだったと記憶している。

IOCSはアセンブラ (Autocoder と呼ばれた) に組み込まれており、プログラムとしては次のように書いた。

- (1) IOCSが指定した形のテーブルに次の項目を設定する
 - (7) まずファイルとしての名前か番号
 - (4) テープ装置の入出力チャンネル、
ユニット番号、

- (4) レコードの長さ (バイト数)
ブロック化係数、バッファの数
- (7) レコードの格納領域の名前
バッファ領域の名前
- (2) アセンブラで下記の領域を確保する
 - (7) レコード格納領域
 - (4) バッファ領域これらにつけた名前がIOCSのテーブルに記入されることになる。
- (3) アセンブラのコーディングでは、マクロ (OPEN, GET など) を使う。この対象となるファイル名か番号には、IOCSテーブルに設定したものを使う。

チャンネルやユニット番号を書いたのは機械語がそういう形になっていたからである。内容的には COBOLのENVIRONMENT DIVISIONと DATA DIVISIONを書いてゆくという感じであった。

コンピュータ処理方式としては、バッチ処理しかしていなかった。データは全て紙カードに穿孔され、コンピュータに入力された。結果は全て紙に印字してもどされた。我々はこれを SPOOLを使わずに、以下に述べるサテライト方式と呼ぶやり方で処理していた (図1)。

まずデータがカードに穿孔され、それを1401を使って磁気テープに書き込む。その磁気テープが7070に渡されそこで必要な処理がなされる。印字される結果も磁気テープに記録され、それが1401に渡され印字される。7070にもカードリーダーやプリンタはついていたが、原則としてそれは使わないことにし、磁気テープのみを使って出来るだけ高速で処理しようとした。この方式は、IBMの第2世代コンピュータでは、かなり広く使われていたように思う。

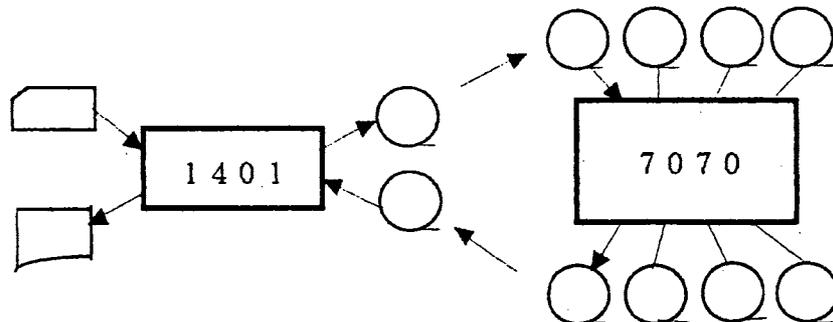


図1 サテライト方式

4. 3 テストとユーティリティ

カードからのデータを読み結果を印字するだけのプログラムについては、そのやり方はいくらでも見当がついた。サンプルデータをカードに穿孔し、結果を印字させれば良かったからである。

ところが磁気テープとなるとどうしていいかちょっとわからなかった。サンプルデータは磁気テープに書き込まなければならない。まずデータ自身が1枚のカード(80桁しかない)に入りきらないことが多い。それを磁気テープに記録するのにプログラムが必要なことは明らかである。このプログラムを、テストの度に作らなければならないとしたら大変だと思っていた。いわゆる Card to Tape や Tape to Print などのユーティリティプログラムがあることはわかったが、1つ1つ単独に使うのはわずらわしい感じがした。この時IBMが提供してきたのがPATというパッケージである。

PAT (Procedure for Automatic Testing)を使ったテストのやり方を概説すると次のようになる。

問題として、TP1 と TP2 の2つのファイルを読んで、TP3 と TP4 の2つのファ

イルを書く、APというプログラムのテストを考えることにする。APについては、すでにコンパイルが行われ、そのオブジェクトカードが作られていることにしよう。PAT を使うためのインプットと PAT から得られるアウトプットは

インプット

- ・ TP1 のサンプルデータ
- ・ TP2 のサンプルデータ
- ・ APのオブジェクトカード
- ・ TP3 のダンプ指示
- ・ TP4 のダンプ指示

アウトプット

- ・ メモリーダンプ
- ・ TP3 のファイルダンプ
- ・ TP4 のファイルダンプ

となる。

PAT のパッケージはカードデッキで提供されていたが、そこに図2に示すようにサンプルデータなどを挿入し(若干のパラメータカードを含む)て7070のカードリーダに投入していた。そうすると自動的にサンプルデータがテープに書かれ、プログラムが実行された。終わると若干の操作でダンプ結果がテープに取られ、それを1401で印字してデバックすることになる。

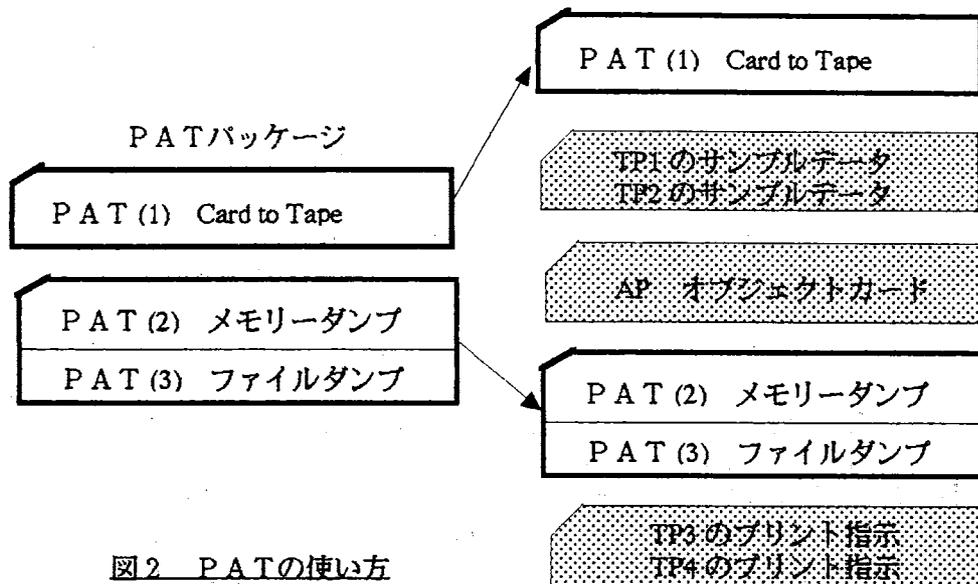


図2 PATの使い方

これはなかなか便利なものだった。素朴な仕組みだが、サンプルデータのファイルを作ること、結果を印字するという基本的な仕組みが出来上がっているのに感心した。始めは中の仕組みもよくわからなかったが、途中からこのパッケージは、ユーティリティプログラムの集合にしか過ぎないこと、ただそれらを若干修正して、1つが終わると次のプログラムがロードできるようにしてあること、などがわかってきた。それだけで結構便利に出来ていることに感心した。

5. オペレーション環境の改善

5.1 本番オペレーションの改善

その頃、OSに類するものはまだ提供されていなかった。プログラムを動かすには、そのプログラムのオブジェクトカードを直接入力して実行するという原始的なやり方だった。ただこれはオブジェクトカードがカードリーダーでつまったり

することも時々あるので、テープにして実行させる形をとっていた。この頃のオペレーションの様子を記すと次のようになる(図3参照)。

- (1) まず PRG 001のプログラムの入っているテープを #0 のテープユニットにかける。ファイルAのテープを #1 に、ファイルBのテープを #2 に、新しいテープを #3 にかけて、実行する。
- (2) PRG 001 が終わると、PRG 002 のテープを #0 に、ファイルDを #1 に新しいテープを #2 と #4 にかけて実行する。

このやり方はどうみてもスマートなものではなかった。まず第1に、磁気テープ1本に1つしかプログラムを入れられないというのは、いくらなんでも不経済だった。第2に、プログラムテープをいつも掛け替えなければならないというのも、相当繁雑だった。このやり方を改善しようとして作ったのがモニタである。

当時、モニタとか、エクゼカティブという名前で、OSというジョブスケジューラに相当するパッケージが出てきていた。7090のFORTRAN Monitorなどの様子はよく聞こえていた。残念ながら7070にはなかったか、機能面で満足しなかったのか、同じ職場にいた西木俊彦氏が独力でモニタを作成した。

モニタの機能の概要を述べると次のようになる。1本の磁気テープにジョブ単位などを基準に、複数のオブジェクトプログラムを格納した。そしてそのテープの中から実行するプログラムを指示したパラメータカードを作成し、そのカードを実行したい順番に並べ、カードリーダーから読ませることで、連続的にプログラムの実行を可能とした。更にプログラムテープへのプログラムの挿入、修正、廃棄などの機能が備えられていた。

5.2 テストオペレーションの改善

先に述べたモニタのおかげで、本番オペレーションはかなりすっきりしたが、テストのオペレーションは全然すっきりしたものではなかった。PATのおかげでツールとしてはいいものが入っていたが作業の仕方はスマートでなかった。

まず第1に、本番のオペレーションは専任のマシンオペレータが担当していたが、テストはプログラマ単位（またはプロジェクト単位）に時間が割り当てられその時間にプログラマが自分でオペレーションしていた。

プログラマ達は（私も含め）、自分で自分のプログラムを動かす。しかしどこかで突っかかり止ってしまう。そうすると、その場でコンソールのタイプライタを通じて、対話的にデバッグ（このやり方をコンソールデバッグと言った）をしていたのである。

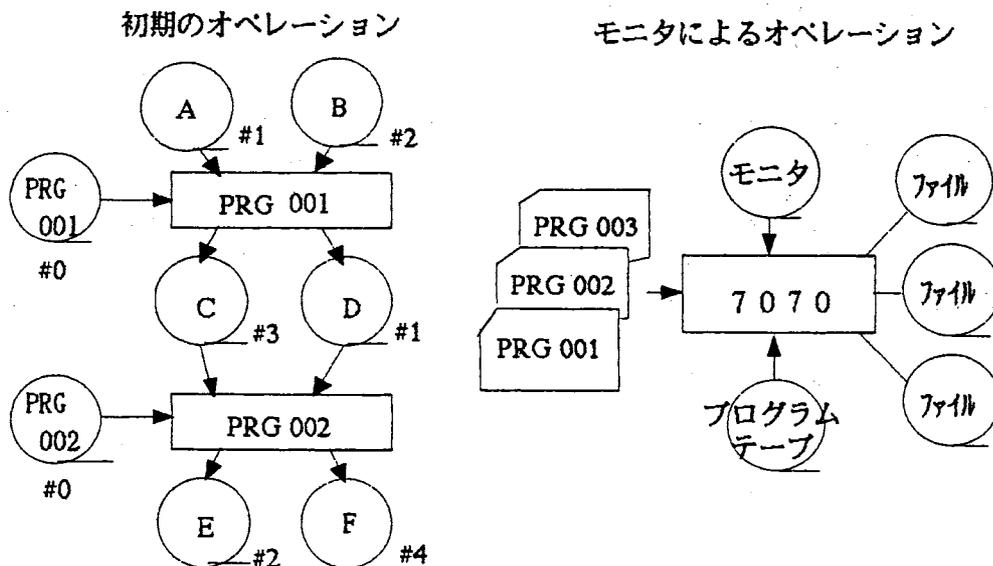


図3 本番オペレーションの変遷

PCの世界で対話式にデバッグすることからみると、何の不思議もないかもしれないが、1台何億円もする機械をたった1人で独占し、20分から30分使ってゆくということになると、問題と言わざるを得ないであろう。

一方で本番業務は増えてくる。日中にプログラマが貰える時間はだんだん無くなってくる。このまま放置するわけにはいかない。テストもオペレータにやってもらおうということになり、私もその検討グループに参加した。

ここでの検討は、一言で言えば「いかに作業ルールを決めるか」という問題であった。例えば、テスト中にプログラムが異常終了する、オペレータはどうするか。プログラマからするといろいろやって欲しい。しかし、オペレータにプログラムリストを見ながら何かやれというのは無理な話しである。せいぜいダンプを取ることに位しか出来ない。このように作業の中味を決めてゆくことが、最も重要な課題であった。

作業面だけでなく、PAT もかなり改良した。先に述べたように中味がわかってきたことが改良をやりやすくした。具体的には、カードベースだったのを磁気テープベースに変える、連続オペレーションの要素を拡大する。個々のユーティリティプログラムを改良するなどである。

ユーティリティの改良では1つ思い出がある。それはファイルダンプのプログラムをまったく新たに作り直したことである。それまでのファイルダンプは、ブロック単位に1行120桁で印字するだけだった。これを普通のファイルは1行10桁単位に間をおいて総計100桁、しかもそれまでは数字だけの印字だったが、数

字・英字の2段打ちとした(7070では英字は数字2桁で表現されていた)。また、報告書スタイルのファイルはまったく実際の報告書と同じスタイルで印字できるようにした。これらは些細なことだがプログラマの要望も強く、また実際にやってみて非常に喜ばれた改善だった。

5.3 7070の限界

モニタとテストシステムの改善を通じて、我々ではどうにもならないことがあるということがわかってきた。それは、次のような問題である。

(1) プログラムの異常終了など、異常事態が起こった時は対応が出来ず、機械が止るだけだということである。入出力装置の異常はIOCSがキャッチしそれなりの対応をしていたが、それ以外は何もなかったように思う。割り込みはいろいろあったから、それを使えば何か出来たのかもしれないが、とてもそこまでは手がまわらなかった。

(2) テストシステムの改良の中で、アセンブラの処理系やFORTRANのコンパイラを1つのプログラムとして扱うことは出来なかった。このあたりはアンタッチャブルな感じであった。

(3) 入出力装置を変えることが難しかった。先にも述べたようにプログラム内に装置番号みたいなものを書かなければならない。オペレーション時点でそれが変えられないかと考えたことがあったが、よくわからなかった。もっとも磁気テープ装置のユニット番号は、表示板形式になっており、手で簡単に換えられたのでかなり助かった面があった。

これらの問題は第3世代コンピュータの登場を待って解決されることになる。

6. 第3世代コンピュータ

第3世代のコンピュータ IBM 360 の話しを聞いたのは、1964年だったと記憶している。その時先にも述べた問題点などが、完全に解決されたという感じがして嬉しかったことをよく覚えている。実際は、膨大な JCL体系に悩まされることになるが、発表時点では非常に新鮮に感じたことは確かである。7070体制と変わった点として実感したことを挙げると次のようになる。

(1) コンパイル時間が短縮され、コンパイラ言語が実用化されたこと。7070では、マクロをかなり含む Autocoderのプログラムは、数百行のものでもコンパイルに30分近くかかっていた。FORTRANは比較的コンパイルが早かったが、それでも10分から15分はかかっていた(200行位で)。COBOLが実用化したのも第3世代からである。

(2) オンラインシステムが実用化したこと。これは第2世代に比べ格段の違いであった。

(3) そしてOSの登場。OSという形で環境が作られるようになったこと。これも第2世代には無かったことである。

第3世代と第2世代には大きな違いがある。ただ第2世代の中で我々はOSと同じような環境を作り、コンピュータを使いやすくする努力を重ねてきた。第3世代はその努力に応える形で生まれてきたような気がする。今後もコンピュータを使いやすくする努力が、次の世代のコンピュータを生み出すと考えるのだが、どうだろうか？

参考文献

- [1] M. V. Wilkes, D. J. Wheeler, and Stanley Gill,
The Preparation of Programs for an Electronic Digital Computer, Second Edition, Addison-Wesley, 1957