

タイムシェアリングシステム ETSS の研究開発

弓場敏嗣

工業技術院電気試験所電子計算機部

(現在、国立大学法人電気通信大学・名誉教授)

【要旨】 通商産業省大型プロジェクト「超高性能電子計算機の研究開発」の一環として、1966年10月から1971年3月まで、電気試験所はタイムシェアリングシステム ETSS(ETL's Time Sharing System)の研究開発を行った。1967年2月に港区芝公園に位置する機械振興会館内に電子計算機部計算機方式研究室分室(芝分室)が設置され、総勢20余名からなる開発体制が整備された。1968年5月には、日立製作所の高性能計算機 HITAC8400 上で ETSS を稼働させ、電気試験所永田町庁舎1台、東京大学本郷地区3台、駒場地区宇宙研究所1台、電電公社電気通信研究所1台の端末を設置し、専用通信回線を介して利用できる運用環境を構築することに成功した。それ以後約3年間に渡って、ETSS の試験的運用サービスを継続した。

1. 背景

1960年代、情報社会の基盤をなす計算機技術は飛躍的な発展を遂げた。中央処理装置(Central Processing Unit)の処理速度の向上、記憶装置の容量の拡大、周辺装置の多様化が計算機のシステムとしての応用領域を飛躍的に拡大していた。通産省工業技術院電気試験所(現在、特定研究開発法人産業技術総合研究所)は、1966年度から5年計画で超高性能計算機の研究開発を開始した。このプロジェクトは、1970年代初期において世界最先端に位置する国産の超高性能計算機の開発を目指すものであった。1965年、米国IBM社が大型計算機IBM360/67を発表し、それに衝撃を受けた通商産業省が国策として始めたものである。電気試験所が技術的中核となり、官学民一体となって行う国家的な技術開発プロジェクトであった。

当時の計算機の技術開発はハードウェアが中心であり、オペレーティングシステムと呼ばれるシステムソフトウェアに関しては十分な位置づけが為されていなかった。日本国内においても、国鉄の座席予約システムの開発などで、システムソフトウェアの重要性が知られ初めた時期であった。1台の高性能計算機を、複数の利用者が複数の遠隔地から同時に利用することを可能とする時分割方式(TSS; Time Sharing System)が考案された。従来の計算機センタにおける利用形態は、一括処理方式(Batch Processing System)と呼ばれていた。一括処理方式は経済性の点から無駄が多く、計算機資源の有効利用に限界があった。電気試験所は、超高性能計算機のシステムソフトウェアが、プロジェクトの重要な位置を占めると判断していた。プロジェクト受託企業は、高性能計算機本体のハードウェアとそのシステムソフトウェアの開

発を担当していた。システムソフトウェアの設計、製作に先だって、電気試験所は当時の先端的オペレーティングシステムである TSS の研究開発を行い、プロジェクトの研究的側面を担う役割を求められていた。

1966 年 10 月、通産省大型プロジェクト「超高性能電子計算機の研究開発」の一環として、電気試験所においてタイムシェアリングシステム ETSS (ETL's Time Sharing System) の研究開発が開始された[1]。1967 年 2 月、電気試験所は港区芝公園に位置する機械振興会館内に研究室分室(芝分室)を設置し、総勢 20 余名からなる開発体制を整備した。ETSS の開発を開始して約 1 年後の 1967 年 12 月、単一利用者環境で動作する実験に成功した。1968 年 2 月、芝分室内に 11 台の端末を配置し、複数利用者環境での稼働を実現した。同年 5 月には、電気試験所永田町庁舎 1 台、東京大学本郷地区 3 台、駒場地区宇宙研究所 1 台、電電公社電気通信研究所 1 台の端末を設置し、遠隔地から専用通信回線を介して利用できる実験的運用環境を構築することに成功した。

2. TSS の技術動向

2. 1 TSS の特徴

1960 年代のオペレーティングシステムにおいては、多重プログラミング方式が中心的な課題であった。多重プログラミング方式は、高速な中央処理装置と相対的に低速な入出力機器を効率よく制御して、中央処理装置に無駄な空き時間を生じさせない方式として提案された。1 台の中央処理装置の上で、複数の仮想的な制御単位(プロセスとかタスクと呼ばれた)がそれぞれ付与された役割を実行することが可能となるように管理する。TSS 方式のオペレーティングシステムでは、プロセスと呼ぶ利用者の仮想計算機を、現実の物理的な中央処理装置上を実現する。1 つのプロセスの実行が長くなると、予め決めた時間で処理時間を時分割し、強制的に別のプロセスに切り替える。そうすることで、複数のプロセスに対する処理時間の割り当てを平等に行うことが可能となる。TSS の技術的特徴をまとめると、次の通りである。

1. 計算機のオンライン実時間利用環境
2. 計算機資源の時分割を含む多重化制御
3. 記憶装置の動的な仮想記憶管理
4. オンラインファイルの利用と共有
5. 複数利用者による計算機の同時使用
6. 対話型の計算機利用

ETSS の開発が開始された頃、米国では既に MIT(マサチューセッツ工科大学)におい

て CTSS (Compatible TSS) が IBM7090 上で実験的に稼働しており、TSS 的な利用形態の有効性が実証されつつあった。さらに CTSS の大規模化を目指して、GE 社の GE645 に搭載する MULTICS (Multiplexed Information and Computing Service) と呼ばれる大規模な TSS の開発が進められていた。日本国内でも、日立製作所/東京大学 (HITAC5020)、大阪大学/日本電気 (NEAC2200/500)、慶応大学/東芝 (TOSBAC3400)、九州大学/沖電気 (OKITAC5090) などの産学協同体制で、TSS 開発の実験的な試みが開始されようとしていた。当時、米国では、公衆電話網を介した計算機利用を販売する TSS サービス会社が新しく企業化され始めた。しかし、提供される利用者環境、通信費用などの制約から、利便性と経済性を追求するビジネスとしては成功しなかった。

2. 2 TSS の技術課題

上記のような技術的特質をもつ TSS に対して、システムによるオーバーヘッドが大きく、一括処理的な大規模計算の処理は効率が悪いこと、データの入出力能力が弱体であることなどの利用者から見た課題があった。具体的には、1968 年当時の TSS の技術課題として、以下のようなものが挙げられていた。

1. 仮想計算機を管理するための時空間オーバーヘッドを最小にする。
2. オンラインファイルの便利な利用環境を提供する。
3. 会話型利用におけるシステムの応答時間を短くする。
4. 障害対策によって、安心して利用できる高信頼利用環境を実現する。
5. 利用者に対して、公平感のある課金体系を提供する。
6. 充実した機能をもち、使い勝手がよく、かつ安価な端末を提供する。
7. 会話型処理を有効利用した新しい計算機応用を創出する。
8. 人間の誤りに柔軟に対応する人間機械系を実現する。

2. 3 Saltzer と Wilkes の講演会

電気試験所は、1966 年 4 月、米国マサチューセッツ工科大学 (MIT) から Jerome H. Saltzer 助教授を招聘した。同氏は自分の博士論文[2]を参照しながら、当時 MIT で開発が進められていた MULTICS (Multiplexed Information and Computing Service) に関する講演を行った。これにより、世界で最も注目を浴びていた TSS 開発計画の全貌が明らかにされた。さらに、1968 年 2 月には、EDSAC (Electronic Delay Storage Automatic Calculator) を開発した英国ケンブリッジ大学の Maurice V. Wilkes 教授を招いた。同教授は当時執筆中の著作[3]を用いて、TSS 技術の世界的動向に関する講演を行った。これらの講演会には産学官の若い研究者たちが参集し、1 週間に渡って積極的な質疑応答が行われた。国内ではオペレーティングシステムの技術開発

は緒についたばかりであり、当時の欧米の技術動向を把握することなしに研究開発を進めることは不可能であった。ETSS のシステム設計にあたっては、MULTICS の影響を受けている。工業技術院の海外研究者招聘制度に基づくこれら欧米の権威者との交流は、当時の技術水準の向上のみならず、その後の知見の交換と人材の交流に大いに貢献した。

3. ETSS 開発の基本方針

ETSS の研究開発は、大型プロジェクトで開発する「超高性能電子計算機」のシステム設計に意見を反映させることを目的としている。従来の一括処理方式を前提とした中央処理装置中心のシステム構成から、効率的な多重プログラミングや仮想記憶の実現を可能とするシステム設計が必要とされた。中央処理装置のハードウェア構成に対して、オペレーティングシステム側からの要請を明確にすることが重要となっていた。ETSS には、最先端のオペレーティングシステムを「超高性能電子計算機」に実装する場合の実験システムという意義付けもあった。ETSS では、日本の当時の技術水準に配慮しつつ、以下の独自の目的を設定した。

1. 本格的な TSS の在り方、オペレーティングシステム設計法を研究する。
2. オペレーティングシステムの能率的な開発方法を研究する。
3. 設計自動化を支援する道具の在り方を研究する。

一般にオペレーティングシステムは複数のサブシステムから構成され、さらに何段もの階層化が行われて、全体システムが機能する。TSS においても、多数のサブシステムがバランスよく統合されて、効率のよいシステムとして機能する。このようなバランスの最適な均衡点は設計後に修正を重ねることで導かれる。つまり、オペレーティングシステムの開発にあたっては、適切な全体システムの体系化を行った後、個々の階層において設計、試作、評価、再設計を繰り返すことが重要となる。オペレーティングシステムのような大規模なソフトウェア開発では、多人数による作業分担が不可欠である。多人数による開発体制でのデバッグの方法、評価・検証の在り方など、ソフトウェア構築の生産性向上のための方法論を試行することも本研究開発の課題であった。設計自動化のための道具造りについては、既存のオペレーティングシステムの上で、新しいオペレーティングシステムを開発する手法を提案し、その有効性を検証するというものである。この手法を、当該計算機上で動作する既存のオペレーティングシステムを、新しいオペレーティングシステムの開発環境として道具化する狙いがあった。システムソフトウェアの記述言語についても、書きやすさとコードの実行効率の兼ね合いを追究するという課題を設定していた。研究開発に携わった研究者たちが、高性能計算機に対する理解を深め、システムソフトウェアに関する経験を積む

ことも、副次的な動機付けとなっていた。

ETSS 開発にあたっての具体的な基本方針は、以下の通りであった。

1. ハードウェアは商用の国産電子計算機を用いる。
2. 専用通信回線を介して端末装置を外部機関に提供し、運用する。
3. システムの拡張性を重視し、当面は最小構成とする。
4. システムの体系化、抽象化に務め、ソフトウェア開発の生産性向上を図る。
5. システムの開発期間は 1.5～2 年間とする。

4. ETSS のシステム構成

4. 1 ハードウェア構成

当時最高速の処理能力をもつ国産計算機として、日立製作所の商用 1 号機 HITAC8400 が選ばれた。主記憶は 256KB、サイクル時間 1.44 μ 秒の磁心記憶装置である。2 次記憶としては、仮想記憶およびオンラインファイル用として当初 4 台、後に 8 台の磁気ディスク装置(各 10MB)が設置された。ファイルの信頼性を保証するためのオフラインファイルとして、6 台の磁気テープ装置が付設された。商用 HITAC8400 の中央処理装置に対して、ETSS 開発に不可欠な以下の 2 つの改良を施した。

1. 記憶保護機構
主記憶への STORE 命令の他に、READ 命令、EXECUTE 命令についても、保護機能が動作するように改良した。不正なページアクセスによって、アドレスエラーというハードウェア割り込みが発生する。それを利用して、ETSS では疑似ページ番地方式と呼ぶ仮想記憶を実現した。
2. 計時機構
計時精度を 100 μ 秒とし、ETSS の時分割制御と性能評価の精度向上に役立てた。なお、利用者プログラムは、2 秒間隔のラウンドロビン方式で時分割実行される。

HITAC8400 の中央処理装置は、P1 から P4 まで 4 つのプログラム実行モードをもつ。利用者プログラムは P1、システムプログラムは P2、割り込み処理プログラムは P3 で実行される。P4 はハードウェアのエラーが検出されたときに制御が移るモードである。ETSS では、割り込み処理プログラムを中核として、時分割かつ多重プログラミングの制御を行っている。

利用者端末としては、ディスプレイ端末装置 8 台とデータタイプライタ端末装置 8 台が装備された。ディスプレイ端末装置は構内での利用者端末として利用され、毎秒 2,000 字の表示能力をもつ。データタイプライタ端末は電電公社が新しく TSS 端末として開発したもので、機械式シリアルプリンタとしては世界最高速の毎秒 20 字の性能をもっていた。データタイプライタ端末装置は、変復調機 (Modem) を介して専用通信回線に接続され、遠隔地からの利用者に提供された。

4. 2 ソフトウェア構成

ETSS では、中央処理装置という計算資源を多重化するための制御単位として、「プロセス」の概念を導入した。プロセス管理システムはプロセスを利用して、仮想プロセッサを管理、制御している。時分割処理を行うためには、例えば 200m 秒ごとに計時機構からタイマ割り込みを発生させ、そのタイミングでプロセスの適切なスケジューリングを行う。記憶管理システムは、記憶保護機構を利用した疑似ページ番地方式を適用し、利用者プログラムに対して仮想記憶空間を提供している。ページは 2kB を単位として、主記憶 (磁心記憶装置) と 2 次記憶 (磁気ディスク装置) の間で入れ替えが行われる。磁気ディスク上に構成されるオンラインファイルは、ファイル管理システムによって管理、制御される。ファイルを利用するとき、1 つのファイルは仮想的に 1 本の磁気テープイメージとして見なすことができる。利用者が名前を付けて使用するファイルを管理するために、ディレクトリ管理システムがある。ディレクトリの基本構造は木構造であり、個々のファイルへのアクセス制御、ファイル共有などをオンラインで管理する。端末からの利用者インタフェースとして、コマンド管理システムがある。利用者は、コマンドを介してシステムが提供する機能を使用する。ユティリティシステムとして、端末からプログラムやデータベースなどのファイル作成を行うための道具であるエディタが提供された。その他のユティリティシステムとしては、オフラインによるファイルの格納、障害ファイルの修復、性能評価など、開発者側で内部的に利用する機能が実現された。外部運用開始の時点では、ETSS の利用者用プログラミング言語として、ETSS アセンブラのみが利用可能であった。

システムプログラム (モジュール) は、リエントラント型プログラムとして主記憶内に常駐する。システムコールとしてシステムプログラムが呼ばれるたびに、当該プロセスが関与する各システムプログラムの作業領域が主記憶上に確保される。システムコールはソフトウェアスタック機構により、再帰的に制御されている。実行が終わると同作業領域は解放され、自由記憶領域となる。TSS の制御では、複数のプロセスが同時に並列実行される。従って、処理のある部分は他からの割り込みを禁止して、排他的に実行される必要がある。排他実行の保証は、ハードウェアによる割り込み禁止モードを利用している。

5. ETSS の開発環境

5. 1 HITAC8400 システムシミュレータ

オペレーティングシステムの開発支援ツールとして HITAC8400 システムシミュレータを作製し、ETSS の開発環境として利用した[4]。既存のオペレーティングシステム上で、新しいオペレーティングシステムを開発する道具である。HITAC8400 の磁気テープおよび磁気ディスクベースのオペレーティングシステム TOS/TDOS 上で、ETSS のシステムソフトウェアを作成する環境として不可欠な道具であった。先に述べた ETSS の諸機能は、HITAC8400 システムシミュレータ上で動作する特別なシステムプログラム(モジュール)として作成される。TOS/TDOS 下でのプログラム実行中に、同モジュールへのシステムコールが特権命令として発せられると、制御は ETSS モードに切り替わる。所要のシステムコールが終了すれば再び TOS/TDOS モードに戻る。

開発中のシステムプログラムのデバッグに有用であり、ETSS の開発期間の短縮に貢献した。一般に、オペレーティングシステムの開発途上あるいは開発後に、性能向上のための評価が繰り返されなくてはならない。システムシミュレータは、そうしたシステム解析、評価を行う支援環境としても活用可能である。しかし、時間的制約から、HITAC8400 システムシミュレータでは、定量的な性能評価を行うための支援機能は実現されていない。オペレーティングシステムの設計に際して要請される事項は、それを構成する個々のシステムプログラムの機能と性能の統合として実現される。個々のシステムプログラムは、場合によっては矛盾した要求をするものである。システム総体としての高性能と高機能を得るためには、大局的見地からの定量的なバランスが考慮されねばならない。そうした適切なバランスの発見に寄与する機能がシステムシミュレータに求められる。

5. 2 システム記述言語

HITAC8400 には適当なシステム記述用プログラミング言語が存在しなかったため、ETSS のシステムプログラムはアセンブリ言語で記述された。高水準プログラミング言語による記述は、一般にソフトウェアの生産性向上に貢献するが、作成されたプログラムの実行効率をおとす。ETSS の開発にあたっては、プログラムの保守性を考慮して、プログラム仕様を PL/I 言語でコメントとして記述することを心掛けた。また、アセンブリ言語のマクロ機能を利用してデータ構造の扱いを形式化し、見通しのよいプログラム記述を実現した。プログラム記述における見通しのよさは、デバッグの効率を高めるのみならず、プログラムの実行効率の向上にも貢献する。ETSS 開発経験から、新たなオペレーティングシステムの開発には、強力なシステム記述言語が不可欠であることが認識された。

5.3 デバッグ

当初は、HITAC8400 システムシミュレータを開発支援ツールとして利用して、ETSS システムプログラムの開発を行った。システムプログラム(モジュール)ごとにアセンブラ言語で記述したプログラムをパンチカードに入力し、モジュール単位でデバッグした。モジュール単位でのデバッグが終了すると、他のモジュールと統合し、より大きい単位でのデバッグへと段階的に進行する。虫(バグ、エラー)の在処を巡っては、関連システムモジュール担当者との協調が不可欠であった。各モジュールの完成度が高まった段階で、システムシミュレータを外して ETSS だけの姿で走らせる状況となる。すなわち、TOS/TDOS を取り外して ETSS が直接 HITAC8400 上で動作する環境となる。初期の段階ではまだ虫は捕り切れていないので、異常を発見すると原因究明のため即時にシステムを停止させる措置を講じていた。停止後は、主記憶のシステムプログラム管理領域を 16 進符号でラインプリンタ出力し、その時点でのシステムの状況を辿り、異常の原因を解明した。異常発見時の即時停止は、オンラインファイルに障害を発生させる原因となった。それで即時停止のとき、その時点でオープンされているファイルは強制的にクローズさせ、その後停止する措置をとった。

デバッグのとき、各システムプログラム担当者は HITAC8400 を占有して使用する。1人1回あたりの使用時間を原則2時間として、黒板を用いて時間割り当てを行った。1回2時間ではデバッグに不足と思われる場合は、徹夜の使用もよく行われた。HITAC8400 の利用開始当初は、記憶パリティエラーがよく発生した。デバッグを行うものが自分でパリティエラーの発生箇所を同定し、主記憶装置の全領域を初期化する操作を行っていた。磁気ディスクを含めてハードウェアは使い込むにつれて、障害の発生頻度は格段に減少した。しかし、障害の発生は想定されるので、それに向けた対策は不可欠であった。また、運用の初期段階においては、ETSS 自身が異常を検出して、システムを停止させる事態も起こり得た。

6. ETSS の運用

6.1 外部運用

ETSS が芝分室内で稼働を開始したのは 1968 年 2 月である。この段階では、構内用のディスプレイ端末 8 台、遠隔地用のデータタイプライタ端末 8 台が分室内で使用可能となった。同年 5 月には、東京大学、電電公社電気通信研究所などにデータタイプライタ端末が設置され、専用通信回線を介した外部に対する ETSS の運用を開始した。土日休日を除き毎週 12:00~16:00 の間、同運用は 1971 年 3 月までの約 3 年間続けられた。提供されたプログラミング言語は当初アセンブラのみであったが、その後 ETSS 利用者によって BASIC 言語が開発された。この間、ETSS 利用者と開発者

による研究会が開催された。「ETSS 研究会」と呼ばれた会合には、遠隔端末が配置された東京大学の高橋英俊、森口繁一、穂坂衛、後藤英一、和田英一、元岡達の各研究室から参加があった。ETSS サービスに対する問題点の指摘を中心に、各研究地点での利用の現況についての報告と議論が行われた。

6.2 ファイルの信頼性

ETSS の運用に関してもっとも重要な点は、利用者に信頼性の保証をすることであった。利用者にとっての信頼性は、障害の発生によるシステムの停止を最小限にすることと、オンラインファイルを保護することである。前者については、HITAC8400 ハードウェアの信頼性として、中央処理装置の記憶パリティエラーと磁気ディスクのヘッドクラッシュなどの障害発生が問題であった。

利用者の作成したファイルを保護するために、運用終了後に磁気テープに格納した。毎日の格納は物理格納と称し、4 台の磁気ディスク装置をそれぞれ別の磁気テープ装置に割り当て、レコード単位に並列処理によって書き込んだ。当然ながら、磁気ディスク上のゴミも格納されるが、並列実行による時間短縮は有用であった。月に 1 度は論理格納と称して、ディレクトリ構造を辿りながら逐次的に磁気テープに格納した。ファイルの破損等の障害が発生した場合は、直前に格納した時点にすべてのファイルを復帰させる方法がとられた。なお、論理格納を行う過程で、使用されているオンラインファイルの特性を情報として収集した。ファイルを構成するレコードの長さ、数、利用頻度などを記録し、ファイルの利用特性の統計データとして解析、評価した。

7. ETSS の解析と評価

7.1 性能モニタによるデータ収集

ETSS が稼働を開始し、外部機関に利用環境を提供できるようになった段階で速度性能を改善するための評価を行った[5]。データ収集と解析による性能の改善が第一の目標であったが、オペレーティングシステムの一般的な解析評価手法を見出すことも重要な課題であった。システムプログラム(モジュール)に性能測定用モニタプログラム(性能モニタ)を組み込み、システムの局所的実行結果を時系列の性能データとして収集した。収集したデータは主記憶上に一時記録した後に、磁気ディスクに格納した。アセンブラ言語によるコーディングの技術的改良によって、個々のモジュールにおいて数倍に及ぶ著しい性能向上がもたらされた。その他、ETSS システムの全体設計と機能分割についても、幾つかの改良点が明らかとなった。システム開発途上での性能改良の方法、性能モニタの設計についての示唆も得られている。

7.2 性能の隘路

ETSS の外部運用を開始する以前つまり ETSS 開発途上においても、個々のシステムプログラムの製作過程で性能向上のための様々の改良が為されている。例えば、システムプログラム開発の初期段階で、プロセス管理モジュールが処理速度の点で使い物にならない事態が発生した。動的な記憶領域管理の機能を一般化し、プロセス管理モジュールがシステムコールによってその機能を利用すると、多発するシステムコールによるオーバーヘッドが極めて大きくなる。また、システムコールはそれだけで多大の実行時間(オーバーヘッド)を要するので、システムコールの多発を抑制することが必要であった。システムコールの発生を抑制するために、自分のプログラム内でサブルーチンコールにより所要の機能を代行させるプログラミングが要請された。このような性能の隘路を試行錯誤により発見し、速度向上に繋げる努力が不可欠であった。また、複数のプロセスが存在するとき、それらを効率よくスケジューリングして、システム全体として高性能が実現できるように設計する必要がある。このスケジューリングに際して発生するオーバーヘッドは、スケジューリングアルゴリズムの重さに比例する。現実的な使用環境では、スケジューリングは簡単で軽負荷なもので充分であるのに、複雑で負荷の重いアルゴリズムが実装されていたりする。

ETSS のシステム構成要素の相互関連にまで立ち入り性能上の隘路を見つけ、改良する研究を進めた。システム構築に際して導入されたプロセス、イベント、ファイルなどの基本概念の有効性、コーディング(プログラミング)技術の性能に及ぼす影響と改良効果などについて、データ収集と解析評価を行った。ETSS の評価結果として、オペレーティングシステムのような大規模ソフトウェアの開発においては版改訂が重要であり、当初からそれを考慮したシステム設計と開発行程の管理が不可欠であることが分かった。

7.3 トレードオフの見極め

例えば、オンラインファイルとして用いる磁気ディスク上でのトラック領域管理システムにおいては、領域要求と解除を実行する処理速度が速いことが要請される。一方で、主記憶に常駐させる管理情報を小さくし、主記憶の利用効率を高めることも重要である。これらはトレードオフの関係にあり、一般には主記憶常駐部分を大きくすると、処理速度を速くできる。また、オンラインファイルという 2 次記憶空間に対して、使用方法で専用化し特徴づければ、時間的・空間的により効率的なファイル管理システムが実現できる。通常、一般性が見通しのよさのために効率を落とし、特殊性は複雑さを受け入れて効率をよくする。多くのシステムプログラムで、一般性と特殊性が相克する場面が現れる。現実には、試行錯誤的な方法によって、適切なトレードオフを見つけたいかねばならなかった。個々のモジュールの機能と性能はシステム全体の中で位置付けられる必要があり、全体とのバランスという観点からの定量的な検討が重要で

あった。

8. 反省あるいは多少の後悔

8.1 版改訂

ETSS の開発に従事したものの多くは、プログラミング(コーディング)のスキルをシステム構築の過程で身に付けた。従って、ETSS が性能面でも信頼性の面でも実用的であるためには、組織だった版改訂による再構築が不可欠であった。個々のシステムプログラム(モジュール)をシステム全体の中で位置付け、その結果によって改良を加えることは短期間では困難があった。版改訂の過程でも、開発から離れた人が分担したモジュールのプログラムについては、いつまでも旧版が存続することになる。ETSS の解析評価を進める際、それぞれ機能を分担して開発したシステムプログラムに、他人が手をいれて改良することは容易ではなかった。一般に、各システムプログラム(モジュール)の分担者は、自分のプログラムを他人が変更することを好まない。知的所有権を振りかざす訳ではなく、ただ単に自分が手塩をかけた我が子のようなプログラムを、他人がいじり回すことに対する嫌悪感に起因するものと思われる。何れにしても、性能の隘路を克服しトレードオフを見極めるために、版改訂は開発の当初から計画されておかねばならない。

8.2 経験の交流と蓄積

日本では技術の伝承は長年の徒弟関係の中で行われ、門外不出の技芸(スキル)として同じ技術集団の中に囲い込むという伝統があった。本来、技術は個別的な経験を普遍化し、普遍的なアイデアを個別分野へ適用するというサイクルで発展していくものである。その技術の発展を促進するためには経験の交流が不可欠であり、研究グループ間の閉鎖主義を取り払う必要があった。計算機利用者の方でも、自分のプログラミングの経験を囲い込み、外部には公開したがる傾向が感じられた。オペレーティングシステムの技術的潮流を見ると、1970 年代後半から、米国ベル研究所が DEC 社の中型計算機 PDP11-45 上で開発した UNIX[6]が世界を席卷した。ベル研究所は、C 言語で記述された UNIX のソースプログラムを無償で公開した。それによって、大学・研究機関を中心とした UNIX の利用者が、システムを自由に改良しその経験を共有し蓄積することができる状況が創り出された。システムソフトウェアを記述する能力を C 言語が備えていたことと合わせて、ソフトウェア財産の共有という考え方が UNIX の世界的普及に大きく貢献した。その後、プログラムの利用と複製、改変、および改変したものの再配布を認め、それらの使用の自由を保証するフリーソフトウェアの考え方が普及している。このような発想はソフトウェアという技術文化が初めて可能にしたものであり、日本的なハードウェア至上主義の風土からは生まれなかったのではないかと考えられる。知的所有権は、自分達の事業的成功を存続させる

ための方でしかない。

8.3 ETSS の将来に関する議論

大型プロジェクトの枠組みの下で行われた ETSS 開発が終了したとき、外部への運用サービスを継続するか否かの議論があった。さらに実用化に向けて ETSS を再構築してはどうかとの提案もあった。運用サービスには非研究的な日常業務がとれない、多くの研究者にとってはサービス自体に研究的価値を見出せなかった。ETSS 再構築については、それをやるとすれば企業に委託すべきで、研究者である自分達の仕事ではないとする気運があった。ETSS 開発後の仕事は性能の改良と信頼性と利便性を高める運用の研究であり、開発に従事した研究者の関心と呼ばなかった。その意義や重要性を理解していても、研究の方向が後ろ向きと捉えられ、自分達自身の研究課題とはならなかった。ETSS の研究開発を立ち上げた管理職(複数)が、時期を同じくして組織を離れたことも理由の1つであったと思われる。結果的に、ETSS は 1968 年という早い時期に、第 3 者が利用可能な程度の完成度を実現していたにもかかわらず、その後の日本のオペレーティングシステム技術の発展に、目に見える形で直接的に寄与することはなかった。それらの理由を箇条書きにまとめれば、以下のようになるであろう。

1. 開発担当者に実用化研究への動機付けが不足していた。
2. 性能評価して改良することの研究的意義付けが共有できなかった。
3. システム運用を通じて、枯れたソフトウェアを実現する意欲が欠落していた。
4. ソフトウェアを磨き上げて、利用者に提供するという心がけが乏しかった。
5. 利用者に不可欠なコンパイラなどのソフトウェア開発が不十分であった。
6. ファイルを始めとして、システムの信頼性の保証が充分ではなかった。
7. 通信回線経由の遠隔端末からの利用は、応答速度と信頼性が不足していた。
8. 研究者たちが、大型プロジェクトの枠組みに長く拘束されることを忌避した。
9. 新しい大型プロジェクト「パターン情報処理技術の研究開発」が開始した。

9. ETSS 開発体制

9.1 開発を担当した組織

当時の電気試験所電子計算機部部長は野田克彦、計算機方式研究室長は西野博二であった。同研究室の分室として機械振興会館に芝分室が設置され、そこで ETSS の開発が実施された。芝分室では、ETSS 開発の総括的な管理を行った相磯秀夫、ETSS の基本設計とシステムソフトウェア技術全般について指導を行った淵一博を中心に、以下のように ETSS の開発を分担した。担当者はプログラムを実際に作製したものであり、実際には細部において入り組んだ協力体制が組まれた。相磯秀夫は、

「ETSS 特集号」[1]の中で、ソフトウェアの開発は経験が大事であること、仕事量が膨大で予測が困難であったことを述懐している。さらに、1968年7月の時点で、最大の課題はETSSの安定性であり、性能の改善、機能の拡張への努力が継続されているとし、開発に従事した若い研究員のチームワークと日夜の努力を評価している。

1. プロセス管理(斎藤信男)
2. 記憶管理(金田悠紀夫、古川康一)
3. ファイル管理(飯塚肇、弓場敏嗣、宮川正弘)
4. ディレクトリ管理(横井俊夫)
5. コマンド管理(鳥居宏治、杉藤芳雄、田村浩一郎)
6. ビデオ端末管理(真子ユリ子)
7. データタイプライタ端末管理(川合英俊、大東栄夫)
8. 磁気ディスクインタフェース(山本裕造)
9. オフラインファイル管理(有沢誠)
10. エディタ(真子ユリ子、古川康一)
11. アセンブラ(藤井狷介)
12. HITAC8400 システムシミュレータ(淵一博、田中穂積)
13. 性能の解析評価(淵一博、弓場敏嗣、古川康一)

9. 2 芝分室の雰囲気

一般社団法人情報処理学会の初代事務局長を務めた坂元真澄は、当時芝分室に在住した研究者 20 余名を対象に、1993 年にアンケート調査を実施している。その回答によると、芝分室の生活で一番印象に残っていることとして、「当時の管理職のお陰で、非常にのびのびした雰囲気で研究ができた」、「同年代の若い研究者が、グループを組んで1つのシステムを開発した」、「若い人達の研究グループで、好きなことを自由にやらせて貰った」、「リーダーが若く、よく指導して貰った」、「自由な発想と討論が可能な雰囲気があり、若手研究者の考え方を育む空気があった」、「当時はそうは思わなかったが、電総研を辞めてから、偉かったと思えるようになった仲間が沢山いた」などの言葉が残されている。なお、当時情報処理学会が芝分室と同じ機械振興会館にあったこともあり、芝分室をよく知る立場にあった坂元は、「この研究室の人間は、いつ来ていつ帰るのか見当もつかなかった。よく言えば自由闊達、悪く言うと勝手気侷な雰囲気の集団であった」と評している。

10. 50年の歳月を経ての「謝辞」

ETSS の開発には多くの支援者がいた。直接的な支援を得たのは、日立製作所神奈川工場の人達である。開発の初期段階では、エラーの原因がハードウェアかソフトウ

エアが分からない状況が出現した。HITAC8400 のハードウェア保守を担当された日立電子サービスの皆さんは、昼夜の区別なく保守作業を務めてくれた。遠隔端末およびそれを接続する専用通信回線については、電電公社技術局および電気通信研究所のお世話になった。東京大学の高橋英俊、後藤英一、森口繁一、和田英一、元岡達、穂坂衛の先生方およびそれぞれの研究室の皆さんには、学術的見地から ETSS 開発の意義を支えて頂いた。また ETSS の実験的運用に際しては、不十分なサービス環境にもかかわらず運用に参加し、利用者側から見た問題点を指摘して頂いた。当時の工業技術院電気試験所山田太三郎、百田恒夫の両所長には、大型プロジェクトという新しい研究開発制度の枠組みの中で、電気試験所らしい雰囲気の研究開発を遂行する環境を与えて頂いた。記して心からの感謝の意を表したい。ここに名前を挙げさせて頂いた多くの方々は、既に鬼籍に入られている。改めてこれらの皆さんの支援に感謝し、50 年前の研究開発の記憶を閉じたい。

■参考文献

1. 野田克彦、西野博二、相磯秀夫、淵一博、他：「ETSS 特集」、電気試験所彙報、32 巻、8 号、1968/8
2. Jerome H. Saltzer: Traffic Control in a Multiplexed Computer System, M.I.T. Project MAC Technical Report MAC-TR-30, 6/1966
3. Maurice Vincent Wilkes: Time-sharing Computer Systems, Elsevier Pub., 1975
4. Kazuhiro Fuchi, Hozumi Tanaka, Yuriko Manago and Toshitsugu Yuba: A Program Simulator by Partial Interpretation, Proc. 2nd ACM Symposium on Operating Systems Principles, 97-105, 1969
5. 淵一博、弓場敏嗣、古川康一、田中穂積、真子ユリ子：ETSS の解析(1)ーデータ収集・解析によるシステムの改良、電気試験所彙報、33 巻、12 号、53-74、1969/12
6. Dennis M. Ritchie and Ken Thompson: The UNIX Time-Sharing System, Comm. of ACM, Vol.17, No.7, 365-375, 1974/7

■資料 ETSS@芝分室の日誌

- 1966 年 4 月： 大型プロジェクト「超高性能電子計算機の研究開発」開始
- 1966 年 10 月： ETSS の研究開発開始
- 1967 年 2 月： 港区芝公園の機械振興会館内に芝分室を設置
- 1967 年 12 月： 単独利用者環境で実験的稼働

- 1968年4月： ETSS 運用開始
- 1968年8月： 電気試験所彙報「ETSS 特集」発行
- 1969年春～夏： ETSS の今後に関する議論
- 1970年7月： 電子技術総合研究所に改名、機構改革(組織替え)
- 1971年3月： ETSS 運用終了
- 1971年4月： 芝分室を閉鎖し永田町本部に移転
- 1971年4月： 大型プロジェクト「パターン情報処理技術の研究開発」開始
- 1971年12月： 工業技術院、筑波研究学園都市への移転決定

■ 著者紹介 弓場敏嗣 (ゆば・としつぐ)

1966年、神戸大学大学院工学研究科修士課程修了。(株)野村総合研究所を経て、1967年、通商産業省工業技術院電気試験所(その後、電子技術総合研究所に改名。現在、特定研究開発法人産業技術総合研究所)に入所。以来、計算機のオペレーティングシステム、見出し探索アルゴリズム、データベースマシン、データ駆動型並列計算機などの研究開発に従事。その間、計算機方式研究室長、知能システム部長、情報アーキテクチャ部長等を歴任。1993年より、電気通信大学大学院情報システム学研究科教授。2007年、同大学を定年退職、名誉教授。並列処理・分散処理の科学技術一般に興味をもつ。工学博士。情報処理学会、電子情報通信学会、各フェロー。米国ACM会員。

受理日：2016年4月5日