

解説



リアルタイムシステム

1. リアルタイムシステムとは†

井原 廣一†

1. はじめに

計算機あるいはコンピュータは、その語源の意味する数値計算機能だけでなく、今やデータ処理、制御、通信、推論、イメージ処理、グラフィックスなどの機能を果たすようになってきているとともに、機能を遂行する処理速度の急速な向上により、リアルタイムシステムとしての利用が急である。ところでリアルタイムシステムとはどのようなシステムをいうのであろうか。JIS、あるいはハンドブックの類は、「処理時間に制限をうけるオンラインシステムをリアルタイムシステム」と定義づけているが、より厳密にシステムのもつべき属性を考察し、分類してすることは大いに意義のあることであろう。そしてこれらの考察を通じて、リアルタイムシステムが要求する技術的課題が明らかになるように思える。

筆者は1960年初頭より制御用計算機の開発に携わってきた。それからの30年間のコンピュータ技術の革新と、応用の拡大は目覚ましいものであり、今やほとんどのコンピュータシステムはリアルタイム性を当然もっているとの前提で考えられている。これは、半導体技術の進展にともなう処理速度の向上と故障率の低下が、大きく寄与した結果といえよう。

世界的なリアルタイムシステムの開発動向は文献を参考にさせていただき^{1),2)}、以下、主に我が国における制御用計算機、コントロールコンピュータ、プロセスコンピュータと呼ばれるイベント指向システムとトランザクション処理指向システムを中心に、リアルタイムコンピュータシステムへの実用化の歩みを簡単に振り返り、リアルタイム

システムの定義を試み、技術的問題点をまとめる。

2. リアルタイムシステムの開発と背景

電子式計算機の開発の動機は大砲の弾道計算であり、また流体計算であった。ノイマンらによるストアードメモリコンピュータは開発の当初から部品故障に悩まされた。これは高速で、しかも故障しないコンピュータへのニーズが最初から存在していたことを示している。しかしながら当時のエレクトロニクス技術はこれらの要求に応えることができず、リアルタイム性をさほど強く要求されない分野の科学技術計算および事務処理計算として、利用されることから始まった。1950年代にはオートメーションなる概念が紹介されコンピュータはその中核として位置付けられたが、当時、故障間隔平均時間は12時間程度であり12人の専門保守員が故障のつど8時間もかけて修理したと言われている。稼働率60パーセント程度では、オートメーションの実現に程遠いものであった。まず比較的時定数の遅い連続プロセスである化学工業への適用から始まったからであろうか、プロセスコンピュータとよばれた。

表-1に、リアルタイムコンピュータの発展経過と技術背景と代表的システムを挙げる。

1960年代は、半導体材料がゲルマニウムからシリコンに変わることによって、トランジスタの信頼性も演算速度も格段と向上した時期であった。また、それまでの固定ヘッド磁気ドラムメモリは、浮動ヘッドによる大容量磁気ドラムメモリに変わるとともに、可動部分のない磁気コアメモリ装置が実用化されて、アクセス時間を短縮し、信頼性向上と高速化に寄与している。ソフトウェア技術においては、プログラムの検証を容易にする高級言語フォートランおよび優先順序制御機能

† A Review of Real Time Systems by Hirokazu IHARA (Hitachi Medical Corp.).

†† (株)日立メディコ

によってアプリケーションプログラムの制御を司るプロセスモニタ,あるいはエクゼクティブの出現により,リアルタイムシステム要件を満たしはじめた.1960年代半ばに通信,計算,制御(3C)の高度な発展によってオートメーションが実現されるとの見通しが発表されたが,実用化には多くの問題を抱えていた.このような背景で,この年代末期には科学技術計算機と事務処理計算機との統一も行われたが,一方,TRW,GE,IBM,日立,東芝,三菱電機,富士通などから制御計算機が発売され,当時我が国の経済成長をになった鉄鋼,自動車の組立,折りからのテレビ民間放送の発足によって番組の自動編成送出,火力発電化に対応して火力発電の自動化,流動の拡大に対応して貨物ヤード制御など先進分野に導入されはじめた.しかし,計算機制御システムに許容される上限費用は制御対象コストの3%程度との当時の常識によって導入は限定され,信頼性向上のための冗長システム構成は困難であった.

1970年代は集積回路幕開きの時代であった.部品点数,接続点数の大幅な減少により,コンピュータシステムの性能,信頼性およびコストは,飛躍的に改善され,ミニコンピュータの出現をみるようになった.工業計器に代わるものとして,ミニコンピュータによるDDC(direct digital control)概念が提案されたことは,制御コンピュータが制御機器として考慮できる水準に達しつつあったことを示していたといえよう.経済の拡大と大量生

産,大量消費時代の到来は,製品原価の縮減,交通,通信網の整備への要請であった.新設工場,設備,施設の多くに制御用コンピュータが導入されたが,リアルタイムシステム技術の未熟さにより,やはり限定された範囲への導入に留まっていた.我が国においては,作業員の教育水準の高さと,優秀な製造技術によって信頼性の高い製品の製作技術(フォールトアボイダンス技術)に注力されたため,システム停止時の危惧は残っており,操作員がシステムの異常時に必要なサービスが維持できるかどうかによって,リアルタイムシステムの導入がきめられた.電電公社の電子交換機,国鉄の座席予約システム(MARS)および新幹線運行管理システム(COMTRAC),NHKの自動番組編成送出システム(TOPICS)は人間がとって代わることができないので冗長系を取り入れた数少ない例である.

1980年代にはマイクロコンピュータとLSIが主役に踊り出たことにより,制御システムは広くデジタル化されソフト化された.システム構造に生物システムを範とした自律分散概念が提案された.これらの技術はハードウェアの価格を引き下げるとともに信頼性を飛躍的に改善したので,無停止を目指した冗長システムや応答性向上のため分散システムが容易に取り入れられ,列車,自動車の運転制御や航空機の飛行制御のような人命に直接関わるオンボードシステムにも取り入れられた.情報分野でも米国においてトランザクション

表-1 技術背景

| 年代 | システム | ハードウェア | ソフトウェア | システム例 |
|--------|---|---|---|---|
| 1960年代 | 集中 スタンドアロン | トランジスタ 磁気ドラムメモリ (固定ヘッド) 磁気コアメモリ | アセンブラ プロセスモニタ | RW-300 TRW-330 |
| 1970年代 | 階層分散 スター結合 TSS | IC 磁気コアメモリ 磁気ドラム(浮動ヘッド) | フォートラン PDL MS-DOS VOS DB | GEPAC 4040, 4060 IBM-1800 PDP-8 DG HIDIC |
| 1980年代 | 水平分散 ループ結合 イーサネット 負荷分散 パイプライン構造 | LSI 磁気ディスク フロッピディスク マイクロプロセッサ ASIC ゲートアレイ 光伝送 | C言語 仮想OS Unix OS マルチウインド LISP, Prolog OSI RDB | VAX/DEC Apollo Tandem Stratus SPARC Intel-X 86 Ethernet |
| 1990年代 | クライアント-サーバ データフロー マッシュパラレル 多重構成 | RISC プロセッサ 64ビット CISC フォールトトレラントチップ WSI マルチチップ実装 光バス | 目的指向言語 AI 言語 ADA WINDOW C++ | TRON Alpha-AXP PA-Risc IBM-POWER Hypercube |

処理の単機能商用フォールトトレラントコンピュータの出現があり³⁾、金融の如き世界をおおう通信システムが構築されるようになった。実時間通信が本質の電話システムもマイクロコンピュータなしには考えられぬようになり、電子郵便も日常化された。8ビット組込マイクロコンピュータ、16, 32ビット情報・制御マイクロコンピュータが、幅広く取り入れられた。我が国独自のトロンコンセプトによるシステムも開発された。一方、ハードウェアの高性能化は人工知能技術の実用化を促しリアルタイムシステムに組み込まれ始めたが、応答性と信頼性に新たな問題を提起している。

1990年代に入って、縮小命令コンピュータ(RISC)とULSIの利用が進み、ネットワーク化されたパーソナルコンピュータ、ワークステーションが、ダウンサイジングと呼ばれる潮流を作りだし、システムコンセプトを根本から変え始めた。64ビットマイクロコンピュータも出現した^{4), 5)}。ソフトウェアがハードウェアに代わって主役を演ずることになり、情報システムも、リアルタイム性を前提に建設されるようになった^{6), 7)}。米国で進められている大統領大プロジェクトは、テラフロップスの処理能力、ギガバイトの通信能力、ペータバイトの記憶能力を持つ情報ハイウェイの建設により、画像情報も含めた科学技術医学経済などの処理をリアルタイム化しようとするものである。我が国においても同様な計画がすすめられている^{8), 9)}。

3. リアルタイムシステムとは

現在われわれが目にする情報・制御システムが、すでにリアルタイムで利用されており、改めてリアルタイムシステムの定義を求められると当惑するほどであるが、リアルタイムシステムの定義を調べてみよう。

1960年代の J. Martin によれば、『リアルタイムコンピュータシステムは、データを受信してこれを処理し、その時の周囲条件に十分な速さで反応し、あるいは結果を返送したりすることによって、周囲条件を制御するシステム』と定義される¹⁰⁾。1988年の電子通信ハンドブックによれば『リアルタイム処理とは処理要求あるいはデータが発生したつど、直ちにその処理を行う処理形態

をいう。リアルタイム処理では即時に処理することが要求されるため、一般にオンライン処理と同様な設備形態になっている』としている¹¹⁾。そして1993年になると、P. A. Laplante は『リアルタイムシステムは明示された応答時間制限を満たさなければならないとともに、障害も含めた過酷な条件に対応しなければならない』と障害時の条件を付加している。そして『故障したシステムとは公表された仕様に述べられている一つあるいはそれ以上の要求項目が満足できないシステムである』と定義づけ、実用的なシステムはすべてリアルタイムシステムであるとしている¹²⁾。ここでは初期コンピュータの利用形態であったバッチ処理は、システム分類から外されている。筆者は、上記に、フォールトトレラント性、オンライン拡張性、オンライン保守性を追加する。

リアルタイムシステムとは広義の信頼性(ディペンダビリティという)と密接に関係した応答時間で規定されたシステムということである。そして、応答時間制限を満足できないときに、システム障害状態になるものをハードリアルタイムシステムと言う。システムの取り得る状態は予測可能であり、確定的であり、故障間隔時間がミッション時間に対して十分長くなければならないということになる。この例としては、航空機、自動車、原子力発電所、宇宙機¹³⁾など人命や環境などに直接関係するシステムであろう。応答時間制限のある低い確率で越えても許容されるシステムをファームリアルタイムシステムと言う。これは、システムがまれに通常条件から外れた処理を行うときには、短時間その能力低下を許すということであり、システムは確定的であるが、稼働率の高いことが必要である。生産システム、銀行情報システム、通信システムなどがこの範疇に入る。そして、ワードプロセッサの類はソフトリアルタイムシステムと言い、応答時間に明確な制限がなく、外界あるいは操作者の能力に依存する非確定的なシステムである。

これに対し非リアルタイムシステムは、平均ターンアラウンド時間、スループット、サービスの公平性、利用率をシステム指標とするものである。

以上の議論から、演算時間の高速化、システムの高信頼化を追及してきたこれまでのコンピュー

タ開発の歴史はリアルタイムシステム実現への道であったと言ってもよからう。

4. リアルタイムシステムの評価

リアルタイムシステムの主要な評価値は処理速度と信頼度となる。計算速度を表す指標として、クロック周波数、各命令時間、命令ミックス、ベンチマークがある。クロック周波数は演算ステップの基準であり、処理速度を物理的に決定する。命令の出現頻度の平均値の百分率と命令時間を乗じて算出するのが、命令ミックスであり、事務処理用、制御用、科学計算用などが提案されている。典型的なプログラムを走らせてその時間を測定するのがベンチマークテストである。特に、スーパーコンピュータではベクトル化の程度が計算時間に大きく影響するので、ベンチマークプログラムでの測定が行われる。信頼性の指標として、主としてハードウェアに関して確率論から稼働率(Availability)と平均故障間隔時間(MTBF)、平均修理時間(MTTR)の算出がよく行われるが、新しい尺度として性能稼働率(Performability)、機能稼働率(Functionability)が提案されている。なお、知能処理が、実用化されてきたことにより、上記指標とともにプログラムの検査、検証、機能の耐故障性、情報の秘匿性、システムの安全性を広く包含して、ディペンダビリティ(Dependability)として議論が進められている¹⁴⁾。

現実に確認することが難しい 10^{-10} /時のシステムの信頼性を評価する方法として、マルコフ過程、ペトリネットによる予測技法やシミュレーション技法が開発されている¹⁵⁾。

5. リアルタイムシステム設計上の留意点

前章で述べた指標によって、リアルタイム性を表現できることが分かった。処理性能と信頼性は一方を強化すると他方が低下するように互いに相反する効果をシステムに与える。

5.1 処理性能

処理性能を決めるものとして、演算部の演算速度、メモリへのアクセス時間がある。前者に関するものとして、素子のスイッチング速度、配線距離、演算方式、タスク起動方式、例外処理オーバーヘッド、処理アルゴリズム、プログラム言語などがある。

演算速度の重要な要素として、電子の移動速度で決まる一連の論理のスイッチング時間と論理素子間の配線伝送遅延時間の合計があり、システム同期クロックパルス間隔を決める。これは加工の微細化、有機材料の利用で年々高速化されている。一連の論理鎖での伝播時間がパルス間隔を超えるならば、並列処理化して論理段数を減らす工夫をすとか、クロックパルス間隔時間を大きくして演算速度を遅くすることになる。

演算処理部分においては、演算命令方式、演算命令数、内部レジスタ数、マイクロ命令方式、内部バス幅、演算実行方式などが演算時間を決める。代表的なものとしてフォンノイマン型では複合命令(CISC)コンピュータ、縮小命令(RISC)コンピュータと呼ばれるものがある。多段の命令先取りや命令キャッシュメモリによる演算実行制御、並列処理方式としてはプロセッサ組合せによるパイプライン型、マトリクス型、シストリック型、ハイパキューブ型などが提案されている¹⁶⁾。

システムの総合的な管理を行うオペレーティングシステム(OS)は、ハードウェアを効果的に使用して効率よくタスク実行制御とシステム制御を行うが、システム処理時間に占める割合が問題になる。外界に応じてタスクを起動する方式としてハードウェア主体の優先割込み方式と監視プログラム主体の方式があり、外部事象がデジタル的な変化には前者が、連続的な変化には後者が採用される。メモリ装置や周辺装置との情報交換については、サイクルスチールによるダイレクトアクセス方式と演算レジスタによるプログラム制御方式の使いわけも必要である。

処理アルゴリズムについては、メモリアクセス回数、処理のダイナミックステップ数が、そして使用するプログラム言語については、コンパイル変換効率が処理速度を決める。

リアルタイムシステムは、システムの異常事態のすべてに迅速に対応しなければならないから、システムの目的を達成するための機能のほかに、例外処理を行う機能をもっている。この例外処理のためのハードおよびソフト量は、システムのかなり大きな割合を占めることになり、また、処理時間のオーバーヘッドを増大させ、実質的な応答時間の増大と処理能力の低下をきたす。

5.2 信 頼 性^{2), 17)~19)}

システムの信頼性に関係するものとして、部品の故障率、部品点数、接続点数、環境条件、システム構成、プログラム品質、マンマシン機能、などがある。信頼度の算出は各部品、接続点の故障率と点数から計算され、ハードシステムの故障間隔時間 (MTBF) を統計的に表現してきた。部品の故障率は部品の特性によって寿命モードが異なっていること、環境に大きく影響されることで、絶対数値の算出には実用的に問題がある。しかし、システム間の相対的な比較値としてみれば、有効である。半導体技術の進展にともない故障率、点数、接続点数は劇的に改善されて、故障間隔時間は大きくなった。しかし、LSI 技術により、スイッチング速度が速くなり、電力効率が改善されたとはいえ、部品実装密度が大きくなることは単位体積あたりの自己発熱量を大きくする結果になり、冷却能力が故障間隔時間に影響するようになる。さらにスイッチング速度をあげるためにはより多くの電力消費が必要になる。また、経年変化の影響を避けるための部品のディレイティング使用は、部品数を増やすし、部品特性の耐経年時間変化と耐環境温度変化に有効なクロック時間のマージンは、演算速度を低下させる方向である。パイプライン化や並列計算化などシステム構成による演算速度高速化は、部品点数の増大をもたらす。

ハードウェアの信頼性が着実に向上してきたこと、システムの規模が拡大してきたこと、リアルタイム使用が当たり前になってきたことにより、ソフトの信頼性に大きな関心が寄せられてきた。

処理性能をあげるため使われていたアセンブリ言語は潜在的バグが残ったり、検査が難しいなど信頼性を落とす要因が多く、コンパイラの技術が進むにつれて、高級言語に移行してきた。プログラムの保守性を第一の目標として米国国防省によって開発された高級言語 Ada は、利用が限定されており、C言語が主流となっている。

以上はフォールトアボイダンス技術に属するものであり、フォールトトレランス技術として以下がある。システム内部における異常処理はハード、ソフトにおけるコンピュータ異常にかかわる処理とリアルタイムシステムの機能に含まれる例

外処理がある。前者はハードと OS のもつ必要機能であり、メモリ装置、インタフェース装置などには誤り検出誤り訂正コード、論理回路にはスキャンインスキャンアウトテスト、コンカレントチェック、ビルトインテストなどが、取り入れられている。OS における異常処理は、オーバヘッドを増加させ、処理低下をきたす。後者はアプリケーション技術の範囲に属するもので、あらゆる例外処理をプログラム化することは困難で、状態情報を提示して操作者の介入を仰ぐことにすることが多い。プログラム技術としてカバリブロック、ロールバック、Nバージョンプログラミングなどが、システム構成においては、各種の多重化方式、過去の正常時点まで遡って再処理を行うシステムフォールバック、同一外部仕様で複数のグループが独立にシステムを組みあげ、これらを同時に動作させて多数決の結果を採用するデザインダイバシティ技法がある。

システム停止時間を極力短縮するために、通常のオンライン処理を行いながら、故障部分の交換や、機能の変更追加削除が行えるよう、機能サブシステムレベルのアイソレーションを行う。保守センタの監視コンピュータが通信路を經由してシステムの稼働状態を常時監視することも行う。

システムの規模が大きくなり、複雑になって、操作者に任せられる例外処理が増えてくると、システムに関わる人間の誤操作による事故が重大な問題になる。操作者の思考を助けるヒューマンシステムが要求され、人間工学、ユーザ工学などの助けが必要になる。ベクトルコンピュータ、パラレルコンピュータ方式においては、耐故障技術、特に故障部分を除いて復旧するシステム再構成技術が重要になってくる。

知識工学、ファジィ工学、ニューラルネットが提案されてから、これまでの対象システムの解析とモデル化による確定論的技法に加えて、不確定要素を含んだシステムのディペンダビリティとしての配慮が必要になっている^{20), 21)}。

6. あとがき

コンピュータがリアルタイムシステムとして機能するだけの水準に達しない状態では、人間が全てを決定しコンピュータは人間の補助的存在であった。ついでコンピュータがリアルタイム性をも

ってくるに従い人間と共同で意思決定を行うようにならた。そして最近ではコンピュータが人間の性能信頼性を遥かにしのぐようになり、人間にとって代わる場面が多くなっている。加藤教授はこれらの状態を航空機操縦の例で詳細に論じ、自力本願にたいして、機力本願という言葉を出された^{22),23)}。言いえて妙である。最近、ここで述べてきたリアルタイムシステムのうち、並列/分散システム環境においてレスポンスシステムと呼び、体系化する試みが始まりつつある²⁴⁾。

筆者はコンピュータ開発の初期からリアルタイムコンピュータシステムに関わってきた。プラントの制御、新幹線の運行制御、地下鉄の制御、自動車交通の制御、宇宙システム、宇宙ロボットを経て、今、人命に直接関係する診断治療システムを対象にしている。高齢化の進展にともなう福祉社会において重要な意味をもつリアルタイム医療システムは、今後より高い性能と信頼性を必要とするとともに、知識処理を含めたディペンダブルシステムの開発が焦点になっていくものと思われる。

参 考 文 献

- 1) Avizienis, A. et al.: The Evolution of Fault-tolerant Computing, Springer-Verlag (1987).
- 2) 当麻喜弘編: フォールトトレラントシステム論, 電子情報通信学会 (1990).
- 3) 井原廣一: フォールトトレラントコンピュータ, 電気学会誌, Vol. 105, No. 8, pp. 23-26 (1985).
- 4) Site, R.: Alpha AXP Architecture, Communication of the ACM, Vol. 36, No. 2, pp. 33-44 (1993).
- 5) Hof, R.: Inside Intel, Business Week, June 1, pp. 48-54 (1992).
- 6) 井原廣一: 自律分散システム, 計測と制御, Vol. 26, No. 1, pp. 33-38 (1987).
- 7) Lui Sha and Sathaye, S.: Architectural Support for Real-Time Computing using Generalized Rate Monotonic Theory, 計測と制御, Vol. 31, No. 7, pp. 744-756 (1992).
- 8) 水田 浩: マンダラ計画, 化学装置, pp. 43-48 (1993).
- 9) Office of Science and Technology Policy: Grand Challenges 1993; High Performance Computing and Communications, NSF, U. S. A. (1992).

- 10) Martin, J.: Design of Realtime Computer Systems, Prentice-Hall, Inc., New Jersey (1967).
- 11) 電子通信ハンドブック: (1988).
- 12) Laplante, P. A.: Real-Time Systems Design and Analysis, IEEE PRESS (1992).
- 13) 金川信康他: 宇宙機搭載用コンピュータ, 電子情報通信学会誌, Vol. 73, No. 11, pp. 1209-1214 (1990).
- 14) Laprie, J. C.: Dependability: Basic Concept and Terminology, Springer-Verlag (1991).
- 15) Arlat, J.: SURF-2; A Program for Dependability Evaluation of Complex Hardware and Software Systems, FTCS-23, pp. 668-673 (1993).
- 16) 井原廣一: 超高速演算技術の展望, 計測と制御, Vol. 24, No. 8, pp. 687-692 (1985).
- 17) 井原廣一: システムの高信頼化技術, 情報処理, Vol. 23, No. 4, pp. 327-334 (Apr. 1982).
- 18) 井原廣一: フォールトトレラントコンピュータのシステム構成技術の展望, 電子情報通信学会誌, Vol. 73, No. 11, pp. 1136-1141 (1990).
- 19) Gray, J. and Siwiolek, D.: High-Availability Computer Systems, Computer, Vol. 24, No. 9, pp. 39-48 (1991).
- 20) 井原廣一: 知識工学の産業への応用, 電気学会誌, Vol. 103, No. 3, pp. 204-208 (1983).
- 21) 井原廣一: ファジィとニューロは宇宙航空へ適用されるか, 計測と制御, Vol. 30, No. 1, pp. 928-931 (1991).
- 22) 加藤寛一郎: 墜落, 講談社 (1990.8).
- 23) 加藤寛一郎: ニアミス, 講談社 (1992.4).
- 24) Kopetz, H. and Kakuda, Y. (eds.): Responsive Computer System, Springer-Verlag, Wien, New York (1993).

(平成5年7月15日受付)



井原 廣一

昭和12年1月17日生。昭和34年信州大学工学部通信工学科卒業。同年(株)日立製作所入社。日立工場、大みか工場、システム開発研究所副所長、宇宙技術推進本部主管技師長、同社技師長を経て平成5年(株)日立メディコに転属。現在、同社取締役技術研究所長、計算機制御システム、計算機応用の研究開発に従事。工学博士。IEEEフェロー、AIAAシニア、電気学会、計測自動制御学会、航空宇宙学会各会員、IFIP 10.4 委員など。