解　説

リアルタイムシステム

4. ハードリアルタイムシステムの応用例†

竹　垣　盛　一††

1. はじめに

リアルタイムシステムとは，なんらかの形で外界とインタラクションをもつ計算機システムであると言えよう[1]．プラント制御システム，航空管制システムなどがその代表的なシステムである．外界からはセンサをとおして情報を入力し，外界へは，相手が制御対象の場合，アクチュエータへの指令信号の形態，また，相手が人間の場合，GUI などをとおしての情報提供という形態で出力する．システムには，外界の変化を常に検知し，その変化に対して定められた時間（デッドライン）内に反応することが要求される．これまでリアルタイムシステム設計の現場では，システム技術者の長年の経験に基づいた半ばアドホックな方法でこれらの問題を巧妙に解決してきたが，システムの大規模・複雑化にともない，システムの時間的挙動を予測可能 (predictable)[2] にするシステム設計技術の重要性が認識されだしてきた．リアルタイムシステム応用分野は広範囲におよび，限られた紙面で網羅することは困難である．本稿では，リアルタイムシステムとしての特徴が明確で，その機能が比較的理解しやすいという観点からいくつかの事例を紹介することにする．

2. リアルタイム応用システムの概要

リアルタイム計算機は，種々の入出力装置を介して，プラントおよびネットワーク，周辺装置と相互作用し，それらとの入出力処理にリアルタイム性をともなうところに特徴がある．コントローラや小規模の組込みシステムでは必ずしも OS を搭載しているわけではないが，多くの周辺装置を制御する必要がある計算機システムの場合には OS の機能は不可欠である．ここでは，リアルタイム計算処理に要求される機能と実現のための技術課題について整理しておく．

（a） ハードリアルタイム処理

ハードリアルタイムシステムは外界の変化に対して定められた時間以内に反応して特定の情報処理（タスク）を完了させなければならない．外界の変化は同時並行に進展するため，一般に複数のタスクの並行処理（マルチタスキング）が必要である．そのためには，OS/ハードウェアレベルでの入出力管理，タイマ管理，などの高速割込み処理，およびマルチタスクのリアルタイムスケジューリング（通常，プライオリティベースのスケジューリング）機能が必須である．タスク群のデッドラインが守れるかどうかはタスクのスケジューリング方式に大きく依存する．いくら割込み処理時間やタスクスイッチング時間が短くても，タスクスケジューリング方式が不適切であれば CPU の能力に余裕があってもデッドラインが守れなくなってしまう．たとえば，ラウンドロビン方式や FIFO (First In First Out) 方式は，並行タスク群のデッドライン保証に適していない．処理の緊急度合いに応じてタスクを切り替えるためには，プライオリティベースのタスクスケジューリングが不可欠である．ただし，予測可能なシステム設計のためには，どのようにプライオリティを設定するか，あるいは制御するかが問題である．また，リアルタイムシステムでは資源の相互排他制御（タスク間の同期制御）やネットワーク通信処理，ディスク上のファイルアクセスなどにも特別な配慮が必要である．本特集の「分散リアルタイムシステムのための OS アーキテクチャ」の解説にもあるように，近年のリアルタイムコンピューティング技術，特に，スケジューリング理

† Hard Real Time Application Systems by Seiichi TAKEGAKI (Mitsubishi Electric Corporation, Industrial Electronics & Systems Laboratory).

†† 三菱電機(株)産業システム研究所プラントシステム開発部

論の成果は，これらのリアルタイムシステム設計の問題にきわめて有力なソリューションを提供すると考えられる．また，これらのOSレベルでのリアルタイム機能に加えて，アプリケーションレベルでのプログラミング支援環境を充実させることが重要な課題になっており，リアルタイムシステム構築のためのミドルウェア，CASEツールの開発に期待がかけられている．

（b） リアルタイムデータインタフェース

プラント監視制御などのリアルタイムシステムでは，大量の入出力データ(数千〜数万点)のリアルタイム処理が要求される．まずプロセス入出力装置(PIO)で第一段階の処理(アナログ入出力処理，デジタル入出力処理，割込み信号処理)が施され，その後段で計算機との間でデータの授受が行われる．計算機側でのプラントデータの直接の入出力処理はOSレベルで実行される．通常，OS側ではカーネル内ルーティンとデバイスドライバが入出力処理を司るが，大量データサンプルのためには（カーネルに切り替えることなしに）ユーザプログラムから直接I/O空間にアクセスできる機能(Memory Mapped IO[3])も必要になる．大量のプラントデータの状態変化を漏らさず検出するために，数ミリ秒オーダの分解能のデータサンプリングが要求され，計算機がすべての必要な処理を確実に実施できるように，PIO側でもある程度の前段階データ処理と保持機能が必要になる．特に，プラント事故時などには一度に大量の割込み信号が入ってくるが，計算機側で一挙に割込み信号のハンドリングを完遂することは困難であるため，PIO側で割込み信号をキューイングし，それを計算機側が順次処理していけるようなメカニズムも必要になる．

（c） システムの高信頼化

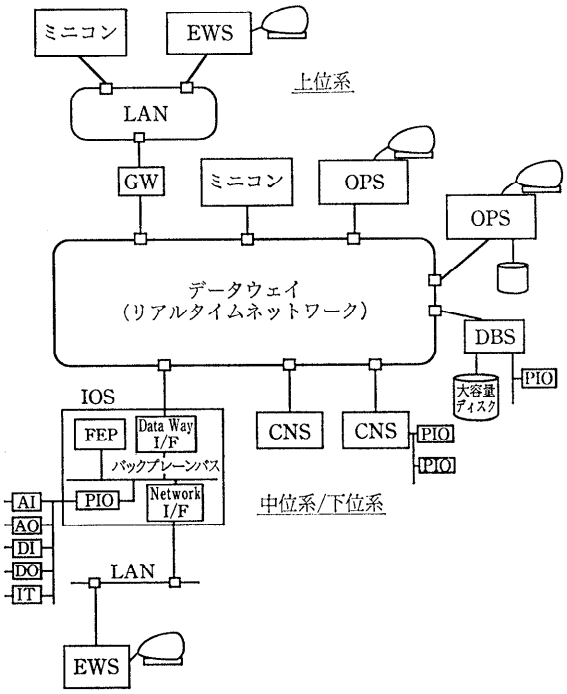
多くのリアルタイムシステムにとって，システムの信頼性はハードリアルタイム処理性能とならんで重要な根本的要件である．特に，フォールトトレラント機能の構築は重要な課題である．フォールトトレラント機能の開発においては，コンポーネント・ハードウェアレベル，システム・アーキテクチャレベル，ソフトウェアレベルでそれぞれ技術課題があるが，リアルタイムシステムにおいては，特に，システム・アーキテクチャレベルの技術が未成熟である．すなわち，故障検出（誤り検出），診断，システム再構成，システム回復などのフォールトトレラントシステムを実現するための要素機能を，リアルタイム制約下においていかに実現するかが重要な技術課題になっている．

3. ハードリアルタイムシステムの応用例

本章では，プラント監視制御分野および宇宙・航空分野におけるハードリアルタイムの実例，実用を目指した試作システム例，ケーススタディを紹介する．

3.1 プラント監視制御システム分野

プラント監視制御システムの典型的なシステムアーキテクチャを**図-1**に示す．統合監視制御システムの全体は，上位の計算機システムからプラントデータの入出力処理装置にいたるまでの数多くのモジュールで構成され，それらが種々のネッ



OPS：オペレータステーション
CNS：コントローラステーション
IOS：I/O ステーション
FEP：フロントエンドプロセッサ
DBS：データベースサーバ
PIO：プロセス入出力装置
AI/O：アナログ入力/出力
DI/O：デジタル入力/出力
IT：割込み信号
GW：ゲートウェイ
EWS：エンジニアリングワークステーション

図-1 プラント監視制御システムの典型的アーキテクチャ

トワークを介して接続されている．大まかには，生産管理や設備管理などの情報管理を司る上位系，運転監視，操業支援，計算機制御などの情報制御を司る中位系，プラントデータ入出力処理，DDC (Direct Digital Control)，シーケンス制御などの計装制御を司る下位系に分類できる．プラントの生のデータは下位系で吸い上げられ，中位系で加工されて上位系に伝わる．また，上位系からの指示が中位系を介して下位系の制御タスクに指令値として伝わる．すべてのタスクにはなんらかの意味でリアルタイム性が要求されるが，そのグレードは各階層でまったく異なってくる．すなわち，上位系には，高度な情報処理が多いが，要求されるリアルタイム性は緩く，なるべく早く応答してほしい（ソフトリアルタイム）という程度である．中位系には，リアルタイム監視などの上位系よりは仕様のきついソフトリアルタイム処理（数百ミリ〜秒オーダ）と CRT オペレーション（CRT からのマニュアルオペレーション）やリアルタイムデータベース管理などに必要なハードリアルタイム処理（処理にデッドラインがある）が混在している．また，プラントデータや制御信号の伝送のためのデータウェイ（リアルタイムネットワーク）にはデータ転送遅延時間の上限値を保証すること（リアルタイム通信）が要求される．下位系はリアルタイムデータの入出力処理，リアルタイム制御など，中位系のそれよりもさらに厳しいハードリアルタイム性（数ミリ〜数十ミリ秒オーダ）が要求される．また，処理すべきリアルタイムデータはシステム全体で数千〜数万点，一台のマシンで数百〜数千点のデータを取り扱うこともある．プランド監視制御の対象は，鉄鋼プラント，上下水処理プラント，化学プラントなど多岐にわたり[1]，その形態もさまざまであるが，ここでは高信頼なリアルタイムシステムの実用例として原子力プラント・ディジタル計装制御システムを，また，プラント監視制御分野での今後の新しい形態のシステムコンポーネントであるリアルタイムデータサーバの試作例を紹介する．

3.1.1 原子力プラント・ディジタル計装制御システム

原子力発電プラント（加圧水型炉；PWR）の監視制御システムの構成は図-1 とほぼ同等であり，統括管理／監視操作／計装制御の3階層よりなる．ここでは監視操作レベルに位置するプラント計算機制御システム (PCCS) と計装制御レベルに位置するディジタル制御システム (DDC) について紹介する．

（1） P C C S[4),5)]

現在の PWR の PCCS は複数の計算機（リアルタイム OS 搭載）による機能分散型システムであり，データ収集計算機 (DASP)，エンジニアリング計算機 (ENGP)，CRT プロセッサ (CRTP) および中央制御室警報監視盤 (ANC) などより構成される．これらは二重化構成のユニットバスに結合されデータの送受信を行っている．DASP は割込み入力（10 ミリ秒分解能でタイムスタンプ），ディジタル入力（0.5 秒周期でサンプル），アナログ入力（0.8 秒周期でサンプル）の PIO 入力処理と事故時データ収集（事故発生入力をトリガにして最高 100 ミリ秒分解能でアナログ入力を記録）の処理を行う．ENGP では，警報印字（警報発生を時系列順に印字），各種エンジニアリング計算（解析計算，など），各種設定情報の入力などを行う．また，CRTP では，CRT 画面更新（1秒周期），表示用トレンドデータの収集・記録，警報入力のチェックと編集・表示，オペレータからのリクエスト入力処理（3秒以内の応答）などを行う．これらの処理の中で実時間性を保証しなければならないハードリアルタイム処理は，DASP の PIO 入力処理，事故時データ収集と CRTP の画面更新，リクエスト入力処理および ENGP の事故時対応に必要な応用計算である．PCCS 全体では 300 本程度のタスクが走っているが，その中でこれらのハードリアルタイムタスクは，DASP, CRTP, ENGP でそれぞれ 20〜30 本程度になる．PCCS では，機能の重要性の観点から，これらのハードリアルタイムタスクに高い優先度を与えたリアルタイムシステム設計を行っている．さらに，どのような事態（事故時など）になっても上記のハードリアルタイムタスクの実時間性を確保できるように，特に，割込み入力多発時の挙動，過負荷時の対処，排他制御における異常時処理などに重点を置いて設計・検証がなされている．また，処理能力をオーバする場合にはプラント運用からみた重要度に応じて機能の縮退を行う必要があり，システム設計者にはプラント挙動に対する深い理解が要求される．

（2） 原子炉制御系 DDC[5]

PWR の原子炉制御系は従来よりコンポーネントごとの分散制御システム構成をとっており，DDC においても従来の分散構成を踏襲している．言うまでもないが，原子炉の制御を司る DDC にはきわめて高い信頼性が要求されるため，何よりも耐故障性を重視したハードウェア設計と制御プログラムにバグを潜ませないようにビジブルでシンプルなソフトウェア設計がなされている．まず，ハードウェア的には待機冗長二重化方式を採用し，異常発生時にはマイクロプロセッサ内蔵の自己診断機能により主系から待機系に自動的に切り替わる．一つの故障モードに対して複数の診断機能で検出し，また，出力信号のリードバックなどの診断により検出率を高めるなど自己診断機能を充実させることにより制御システム単体での MTBF 100 年を達成している．また，二重系の両系故障時には自動的に手動に移行するフェールセーフ設計となっている．制御ソフトウェアは，定周期性と実証性を重視した設計となっている．不確定な挙動を極力排除することが要求されるため，DDC では OS を用いず，割込みを使用しないシングル・サイクリックタスク方式となっている．これにより，プログラムは周期的に起動（500 ミリ秒周期）され，全パスの動作検証が可能な構成となっている．なお，原子炉が制御系のように極度に高信頼性が要求されるケースは別として，最近のコントローラにはリアルタイム OS が採用されることが多くなっている．

3.1.2 プラント監視制御用リアルタイムデータサーバ[6]

近年はプラント監視制御分野においても AI やデータベースなどの高度な情報処理や高度な GUI 機能が要求されるようになっており，これらの情報処理機能を得意とするワークステーション（WS）の導入が試みられている．しかしながら，通常の UNIX 系 WS はリアルタイム仕様にはなっていないので，直接，ハードリアルタイムの世界とつなげることは難しい．そこで，図-1 の左下部にあるように，リアルタイムデータハンドリングを担当するフロントエンドプロセッサ（FEP）を介して WS を接続する方法がよくとられている．ここでは FEP の一つの形態であるリアルタイムデータサーバの試作例について述べる．試作されたシステムは，プラントから定期的にサンプルされたデータにタイムスタンプを付けて保持すると同時に，ネットワークに接続された外部 WS 上のクライアントから送られてくる要求によって現在時刻のデータを周期的に要求元に返したり（周期要求），指定された期間内のサンプルデータを時系列データとして要求元に返す（非周期要求）などの機能をもつサーバステーションである．データの定期的サンプル，ディスクへの定期的保管およびクライアントへの定期的データ転送はハードリアルタイムタスクであり，クライアントの非周期要求に対する時系列データ転送は非リアルタイムタスクである．**図-2** にリアルタイムデータサーバの構成を示す．本システムは分散型実時間プログラミング言語 RTC++[7] で提唱されているアクティブオブジェクトをモデルにしており，1 プロセス上のスレッド群によってタスクシステムを構成している．一定周期ごとのプラントデータの（主メモリ上の）リングバッファへの格納およびリングバッファ上のデータのディスクへの格納が常時実行されているが，クライア

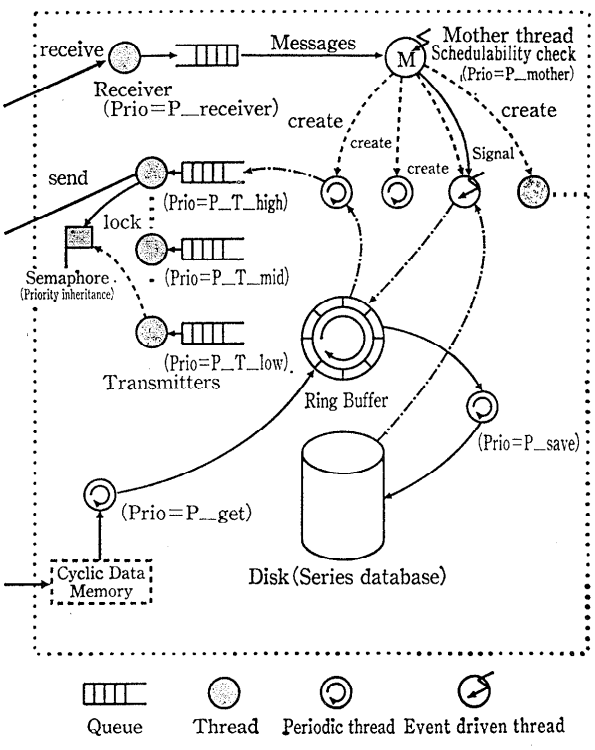


図-2 リアルタイムデータサーバの構成

ントからの周期要求，非周期要求に対しては，マザーと呼ばれるスレッドが要求メッセージの内容に応じてあらかじめ登録されている関数（メソッド）を周期あるいは非周期スレッドとして生成する．各スレッドのプライオリティは基本的に rate monotonic scheduling (RMS)，すなわち，周期のより短いタスクにより高いプライオリティを与える方式をベースに設定している．スレッド生成時にマザースレッドは，あらかじめ登録されている各メソッドの最悪実行時間とプライオリティから，RMS 理論のスケジューラビリティ公式[2]に基づいてスレッドのスケジュール可能性を判定している．リングバッファおよびディスク上のファイルに関するスレッド間の排他制御は，全体をロックせずに，現在書込みを行っているレコードのインデックスに対してセマフォを設け，それに基づいて衝突を回避するように各スレッドがデータにアクセスする楽観的方式を採用し，さらに，優先度逆転[8]が生じないように優先度継承方式[8]でそのセマフォを管理している．このような排他制御上の工夫により各タスクの動作はほぼ独立とみなすことができ，RMS 理論[2]によりリアルタイム性能を精度よく予測することが可能となっている[6]．

3.2 アビオニクスシステム分野

アビオニクス (avionics) という言葉は航空機エレクトロニクス制御の分野の総称に用いられている．アビオニクスは大きく航空機制御と地上管制システムに分けられる．航空機制御システムは高度に自動化の進んだシステムの一つであり，同時に高度の信頼性が要求される．航空機制御システムには，最も内側に機体の姿勢安定化のためのループ，その外側に地上からの情報などを基に機体の位置を判断し，目的地に移動させるための航法のループ，さらに最も外側に想定外の事情（気象の急変など）による飛行計画変更や緊急事態に対応するためのループよりなる．最後のループは地上の管制官とパイロットの対話によって形成される．また，管制官は地上管制システムの支援を受けてパイロットに対するガイダンスを行う．ここでは，航空機制御システムの典型的なタスク構成の概要，および，管制システムの例として NASA ミッションコントロールのリアルタイムデータシステムについて紹介する．

3.2.1 航空機制御システムの概要[9]

ここでは，IBM でアビオニクスシステム設計のケーススタディに用いられている汎用アビオニクスプラットフォーム (GAP) のモデルを基に航空機制御システムの概要を紹介する．GAP は，米国海軍機のミッション制御をモデルにしている．ミッション制御システムは一般に，センサ，アクチュエータ，ミッションプロセッサがシリアルデータバスを介して接続されたマルチプロセッサ構成となっている．ミッションプロセッサのソフトウェアは，ミッションの完全な実行を保証するために，各サブシステムを統合する種々の機能を実行している．GAP でモデル化されている主なサブシステムの機能は次のようになる．

○ナビゲーション：機体の位置，姿勢，速度を計算．

○レーダ制御：目標物体（ターゲット）の位置を計算．

○レーダ警報受信：警戒情報を提供．

○兵器制御：起動によりミサイルなどの弾道計算を実行．

○ディスプレイ：パイロットのスクリーン情報を更新．

○トラッキング：ターゲット情報の更新．

○データバス：ミッション制御計算機と外部デバイス（センサ，アクチュエータ）間の通信．

また，これらのサブシステムのタスクにともなう一般的なリアルタイム制約としては，

○ナビゲーション：航行精度に対する要求から，20 Hz (50 msec サンプル）のタスク周期実行が必要．（マッハ 2 のスピードでは 90 フィートを 50 msec で通過する）

○ディスプレイ：状況変化を視覚的に認識するために 80 msec～100 msec の表示更新（メータ類など）が必要．（キー入力処理は 200 msec 周期のポーリングで実現）

○弾道計算：ターゲットの位置精度を保証するために，タスクが起動されてから 5 msec 以内に処理を完了することが必要．

○センサ制御：ハードウェア制御ループに応答する必要がある．レーダアンテナサーチに対しては 100 msec，電子監視装置に対しては 1 msec 以内の応答が必要．

などがある．ディスプレイシステムのキー入力に

より一連の処理が起動される兵器制御システムを除いて，各サブシステムは周期タスクで構成されている．

これらのサブシステムはデータバスを介して密接に連携している．たとえば，ナビゲーションの更新処理はディスプレイシステムのキー入力処理やレーダサブシステムからの出力を入力とするし，トラッキングサブシステムの出力であるターゲット情報はディスプレイシステムや弾道計算処理の入力となる．さらに，誤り発生時の回復処理のためのデータを各サブシステムが再び利用できるように各サブシステムは通常その入力データを得るデータサーバに計算結果を送り戻す．

ところで，従来よりアビオニクス分野ではシングルサイクリックタスク方式あるいはタイムラインエクゼクティブ（時間軸上に固定的にタスクを割り振る）方式がとられてきた．しかし，対象の規模が拡大し，システムが複雑になってくるとタイムラインエクゼクティブのような固定的なタスク割当方式ではロバストなシステム設計が非常に困難になり，システムの拡張性，保守性の面で問題があることが認識されてきた．GAP のプロジェクトでは，Ada のマルチタスキングによって上述のようなアビオニクスシステムを設計することが検討されている．特にそのリアルタイム性能が RMS 理論を用いて詳しく解析されており，優先度継承などのリアルタイムコンピューティング手法を駆使することにより高い CPU 負荷率（典型的なテストケースでは約85%）においてもすべてのリアルタイム制約を保証できることが示されている．

3.2.2 NASA ミッション制御用リアルタイムデータシステム[10]

NASA ではこれまでメインフレーム主体であったミッションコンピュータの種々の領域に WS を導入することが積極的に行われている．チャレンジャーの事故以降，スペースシャトル運航の意思決定に重要な役割を果たしているリアルタイムデータシステム (RTDS) がその一つにあげられる．RTDS はさまざまな用途に応用されているが，ここではスペースシャトルの着陸ガイダンスへの適用について紹介する．この場合，RTDS のハードリアルタイム処理はリアルタイムデータ（中継衛星から送られてくるスペースシャトルのテレメトリ情報など）を WS に取り込むデータ収集系にあたる．RTDS の構成を**図-3** に示す．リアルタイムデータを最初に取り込む WS ではハードリアルタイム処理が要求されるためプロセスの優先度付け可能なリアルタイム UNIX が搭載されているが，この WS におけるハードリアルタイム処理を最小にするために，基本的なテレメトリ処理は専用のテレメトリプロセッサで実行される．ここで抽出されたパラメータは，DMA 転送により約 10,000 Byte/sec で WS に送られる．テレメトリプロセッサボードには 2K バイトのテレメトリ用 FIFO バッファが設けてある．このバッファが溢れないように WS にテレメトリ情報を転送することが第一のハードリアルタイム処理になる．リングスタッファと呼ばれるタスクが DMA コントローラを制御して，このテレメトリ用 FIFO から WS の主メモリ上に設けられたリングバッファに順次データを転送する．リングスタッファが最高のプライオリティをもつ．次に，バッファスタッファと呼ばれるタスクがリングバッファからタスクを読み込んで，他のアプリケーションからアクセスされるフレームバッファ（共有メモリ上に設定）にそのテレメトリ情報を書き込む．バッファス

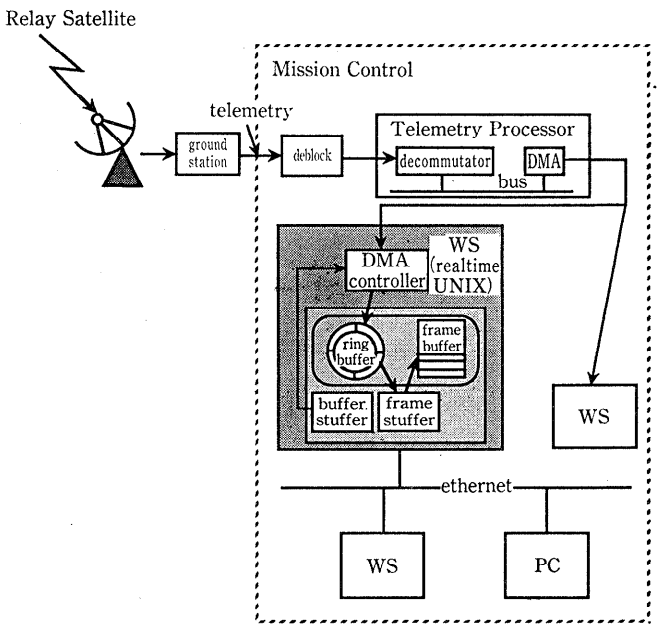


図-3 NASA リアルタイムデータシステムの概略構成[10]

タッファはリングスタッファよりは低いが，他のアプリケーションよりは高い優先度が設定されている．リングスタッファとバッファスタッファは共有メモリを介してリングバッファ上のアクセスしている位置を互いに通信している．前述のリアルタイムデータサーバの場合とは異なり，バッファスタッファが読み込んでいるリングバッファ上のセグメントをリングスタッファが上書きするような状態では，リングスタッファはスペアのバッファにDMAでテレメトリ情報を書き込む方式をとっている．フレームバッファにはさまざまなアプリケーション（監視画面表示，診断・ガイダンスなど）が専用ライブラリを介してアクセスしている．また，バッファスタッファはすべてのパラメータのログをディスク上にとっている．RTDSはリプレイモードをもっており，このモードではデータ収集は停止し，代わってリプレイスタッファがこのログデータをフレームバッファに書き込んでいく．フレームバッファにアクセスしているすべてのアプリケーションはこの仕掛けによりプレイバック動作させることができる．LUNAのランディングに使われたミッション制御センタでは，大型のメインフレームで1秒間に処理できるパラメータの数は1,000個以下であったが，RTDSでは一つのWSで約4,000個のパラメータを処理してアプリケーションに提供することができるなど，ミッション制御の分野に大きな改善をもたらしている．

4. おわりに

本稿ではリアルタイムシステムの特徴や最近の技術動向を分かりやすく説明することに主眼を置いて応用事例を紹介してきた．タスク構成を中心にシステム構築のイメージが見えるように解説したつもりである．リアルタイムシステムの本質である予測可能なシステム設計に少しでも興味をもっていただければ幸いである．

なお，本稿はサーベイという点ではリアルタイムシステム全般からみてやや偏りがあり，他の多くの重要な分野（FA・ロボット分野，通信分野など）やマルチメディア（連続メディア）応用などの興味深い事例についてはまったく触れていない．たとえば，参考文献1）の第7章には広範囲にわたる応用事例が紹介されているし，参考文献11)～13）にも最近の事例がいくつか紹介されているので参照していただきたい．

参考文献

1) 三巻，桑原編著：制御用計算機におけるリアルタイム技術，コロナ社（1986）．
2) Sha, L. and Sathaye, S.S.: Architectural Support for Real-Time Computing Using Generalized Rate Monotonic Theory, Journal of the SICE, Vol. 31, No. 7, pp. 744-756 (1992).
3) Laplante, P. A.: Real-Time Systems Design & Analysis, An Engineer's Handbook, IEEE Press (1992).
4) 犬房，他：将来プラント向け総合計装制御システム，三菱電機技報，Vol. 64, No. 3, pp. 211-215 (1990).
5) 岡崎，松宮，路次：大飯3・4号機向け計装制御システム，三菱電機技報，Vol. 64, No. 3, pp. 202-210 (1990).
6) 水沼，他：リアルタイム・マルチスレッド・プログラミング支援ツールの開発，電子情報通信学会技術研究報告，CPSY 92-80, pp. 1-8 (1992).
7) 石川，徳田：分散実時間プログラミング言語RTC++，コンピュータソフトウェア，Vol. 9, No. 2, pp. 28-47 (1992).
8) Rajkumar, R.: Synchronization in Real-Time Systems—A Priority Inheritance Approach, Kluwer Academic Publishers (1991).
9) Locke, C.D. et al.: Building a Predictable Avionics Platform in Ada—A Case Study—, Proc. 12th IEEE Real Time Systems Symposium, pp. 181-189 (1991).
10) Muratore, J. F. et al.: Real Time Data Acquisition at Mission Control, CACM, Vol. 33, No. 12, pp. 19-31 (1990).
11) Schiebe, M. and Pferrer, S.: Real-Time Systems Engineering and Applications, Kluwer Academic Publishers (1992).
12) Lawson, H. W.: Parallel Processing in Industrial Real-Time Applications, Prentice Hall (1992).
13) Goldsmith, S.: Real-Time Systems Development, Prentice Hall (1993).

（平成5年4月30日受付）

竹垣　盛一

1953年12月31日生．1981年大阪大学大学院基礎工学研究科卒業．工学博士．1981年三菱電機(株)入社，中央研究所を経て，産業システム研究所勤務．研究テーマ：リアルタイムシステム，分散システム，フォールトトレラントシステム，インテリジェントコントロール．著書「知的制御システム」(海文堂)他．計測自動制御学会，システム制御情報学会各会員．