解　説

TRON プロジェクトの現状と展望

4. CTRON サブプロジェクトの現状と展望
——リアルタイム性能とポータビリティ†——

大久保　利　一††　曽根岡　昭　直†††　花　沢　　満††††

1. まえがき

CTRON は，超多重リアルタイム処理を要求される交換システムに代表される通信機器向けの OS インタフェースであり，伝送装置制御機能，情報通信処理など広範なアプリケーションへの適用を狙いとしたものである．同時に，数百万ラインを越える通信機器制御用のソフトウェアの生産性向上のためには，ソフトウェアの再利用を図ることが重要であり，このためにポータビリティ向上をも狙いとしている．また，CTRON は完全にオープンな仕様として公開されており，世界中のベンダ／ユーザが自由に利用することができる．

本稿では，まず CTRON のアーキテクチャと特徴について概観し，その中でリアルタイム性とポータビリティとを満足させるための特徴的な事項について述べる．その後にリアルタイム性の評価手法について述べ，リアルタイム OS 製品の性能の把握の考え方，他の OS との性能比較の考え方などについて述べる．また，ソフトウェアポータビリティの向上を目的として2年間にわたって実施された“ソフトウェア流通性評価実験”の成果について述べる．最後に，通信機器向けリアルタイム OS としての今後の課題などについて述べ，今後の展望に替える．

2. CTRON アーキテクチャの概要と特徴[1)~4)]

2.1 CTRON アーキテクチャの概要

（1） アーキテクチャ構成

最近では UNIX やパソコン用 OS においても階層化の議論がなされているが[5),6)]，CTRON では検討の当初から2階層のインタフェース構成を取り，下位の OS インタフェースを基本 OS インタフェース，上位の OS インタフェースを拡張 OS インタフェースと呼んでいる（図-1）．

下位の OS インタフェースは，プロセッサ，メモリ，タイマなどの機能を論理的なモデルで表し，ハードウェアの個々のアーキテクチャに依存しないインタフェースを提供することを狙いとしている．これにより，アプリケーションプログラムのみならず拡張 OS 自身をも流通可能とすることができる．

（2） サブセット規定とプロファイル

CTRON では，インタフェースのサブセット規定を明確に行い，システムの必要とする規模／機能に対応した最適なインタフェースセットを採用することが可能となっている．たとえば基本 OS でのカーネルサブセットでは，フルセットでは 140 個のシステムコールに対して最も小さな μC サブセットではわずか 48 個のシステムコールとなる．

さらに，サブセット間の上位互換条件を規定することにより，システムの成長にともなって上位ソフトウェアへのポータビリティが保証されてい

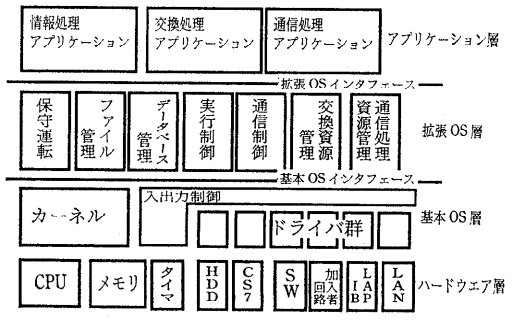


図-1　CTRON 参照モデル

† The Present and Future of CTRON Sub Project—Realtime and Portability Feature—by Toshikazu OHKUBO (NTT Network Service Systems labs.), Terunao SONEOKA (NTT Service Engineering Headquarters) and Mituru HANAZAWA (NTT Information and Communication Systems labs.).

†† NTT ネットワークサービスシステム研究所
††† NTT サービス生産本部
†††† NTT 情報通信網研究所

る．また，各機能単位ごとのサブセットを適用分野ごとの標準的な組合せとしてガイドラインを示し，これをプロファイルと呼び，標準プロファイル間でのポータビリティの確保に当てている(図-2)．

なお，CTRON は他の TRON 系 OS とほぼ同様の概念定義を用いており，教育面でもどの TRON 系 OS から入門しても理解が容易となっている．

また，インタフェース規定にあたっては，国際標準にある通信プロトコルや言語仕様は，これに準拠することとし，リアルタイム性や通信処理に関連するもので国際標準にないものを中心に規定している．

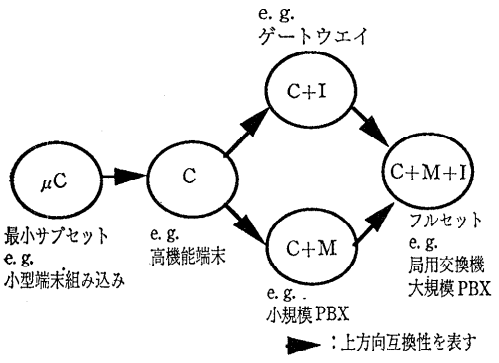
2.2 超多重リアルタイム性確保のための特徴

通信システム，特に多数の加入者を取り扱う公衆ネットワークシステム向けの機器においては，機器組込みのロボット制御などのシステムと異なり，桁違いの多重度制御が要求される．

具体的には，交換システムでソフトウェアでのタスク多重度としては，数千以上のタスクが必要となる．一方，ロボットや家電製品などでは高々対象となるセンサやアクチュエータの数程度であり，実際には数個から数十個の多重度が要求されるに過ぎない．

超多重資源環境でのリアルタイム性確保のため OS インタフェースが考慮すべき特徴的な点について以下に述べる．

（1） 対象資源の限定と事前処理



μC：最小サブセット単位
C：標準機能サブセット単位
M：タスク間通信拡張サブセット単位
I：メモリ管理拡張サブセット単位

図-2 CTRON でのサブセット定義例
（カーネルインタフェースでのサブセット定義）

OS 内ではきわめて多数の資源がシステム内に存在する環境下で，リアルタイム性を確保して特定資源の検索／操作を行う必要がある．

図-3 に示すように資源状態をきめ細かく規定しておくと，超多重資源下でもリアルタイム性を確保した，インプリメントが可能となる．

たとえば，CTRON を含む，TRON 系 OS インタフェースでは，タスクの状態に Suspend 系の状態を定義している．この状態をインプリメント上では，操作対象の範囲外とすることが可能となり高速化に資する．

また，タスクの生成／削除機能は，タスクの起動／停止などと比べて処理量を要するが，あらかじめこれらの生成処理を実施しておき，サービス開始時にはタスクの起動処理のみを行えば高速に処理に取り掛かることができる (Dormant 状態の定義)．

（2） 内部アルゴリズムの自由度の保証

システム内にきわめて多数の資源を定義し操作するためには，システムの条件に対応して最適な内部アルゴリズムを取る必要がある．

資源の管理方式としては，トラフィックが定常的なシステムでは静的な管理方式が，トラフィック条件が幅広く他の資源とのバランスが大きく変動するような場合は，資源を動的に確保し割り当てる方式が好ましい．このような条件への適用を図るためには OS インタフェース規定においては内部アルゴリズムを規定しないことが必要である．

CTRON での例としては，メモリ管理を静的／動的な管理のいずれにも依存しないインタフェース規定としている．

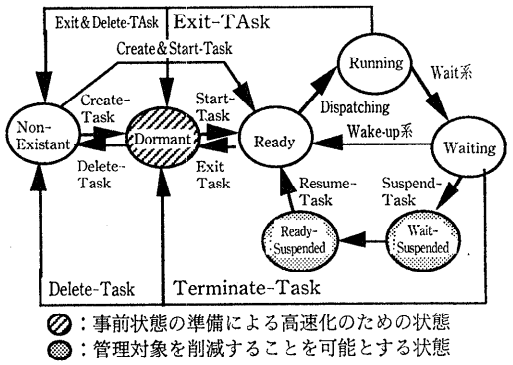


図-3 CTRON でのタスク状態遷移図

2.3 ポータビリティ確保のための特徴[7]

リアルタイム処理ソフトウェア（アプリケーションおよび拡張 OS）のポータビリティの確保は CTRON での重要な特徴の一つである．

CTRON は，その規定にあたってポータビリティに対する配慮として以下の施策を取った．

（1） 使用するマイクロプロセッサのアーキテクチャによらず，どのベンダのハードウェアにも適用可能なインタフェース規定とするため，論理的な資源モデルを構築し，これに基づいて OS インタフェースを規定する．

（2） OS 仕様に準拠した製品のコンフォーマンス（検定）技術を合わせて研究し，製品の機能の過不足を検出する検定システムでの検定による認証を実施している[8]（図-4）．

3. リアルタイム性評価技術

3.1 リアルタイム性評価の課題と方針

OS の性能比較の指標には，割り込み応答時間，タスク切り替え時間，タスク間通信時間などが用いられている[7]．CTRON の適用領域においては資源の多重度が，数千を越える大きな所での処理時間の上限が保証されているかを評価する必要がある．また，評価時のハードウェアおよびコンパイラ環境によって評価データは大きく異なり，OS 自体を比較評価する場合には，これらの条件を正規化する必要がある．さらに重要なことは，ユーザが構築しようとするシステムにおいて，各 OS 機能要素評価のうちどの要素を重視し，総合的にどの OS を採用すべきかを示すシステム評価尺度を確立することである．

3.2 OS 機能要素の評価

OS 自体を評価するためには，OS やハードウェアにプローブを挿入することなく，外部から OS へ入力を与えてそれに対応する出力までの処理時間を測定するベンチマーク手法が通常用いられる．OS のリアルタイム処理性能を測定しようとするものとしては，Rhealstone 測定法[9]などや，SSCL ベンチマーク[10]がある．しかしながら，これらは，対象とする資源の負荷が変化する状況下でのリアルタイム性能を測定するものではない．

CTRON[11)~13)]では，負荷（資源の多重度）を変化させて，各システムコールの応答時間を測定する CTRON ベンチマークプログラムを開発し，7 社の CTRON 製品を対象に，CTRON カーネルのすべてのシステムコール対応に処理時間を測定した．図-5 に示すように，ほとんどの CTRON 製品は，資源の多重度が増大してもシステムコールの処理時間がほぼ一定である．CTRON 仕様はこのような性能条件を満足するインプリメントが

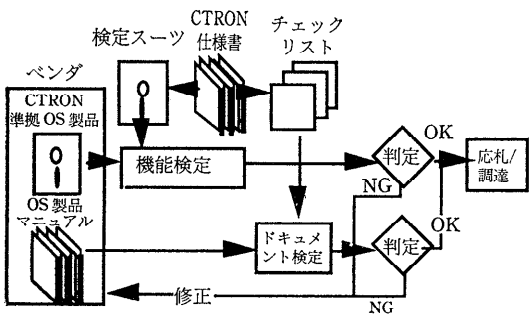


図-4 CTRON 検定方式の概要

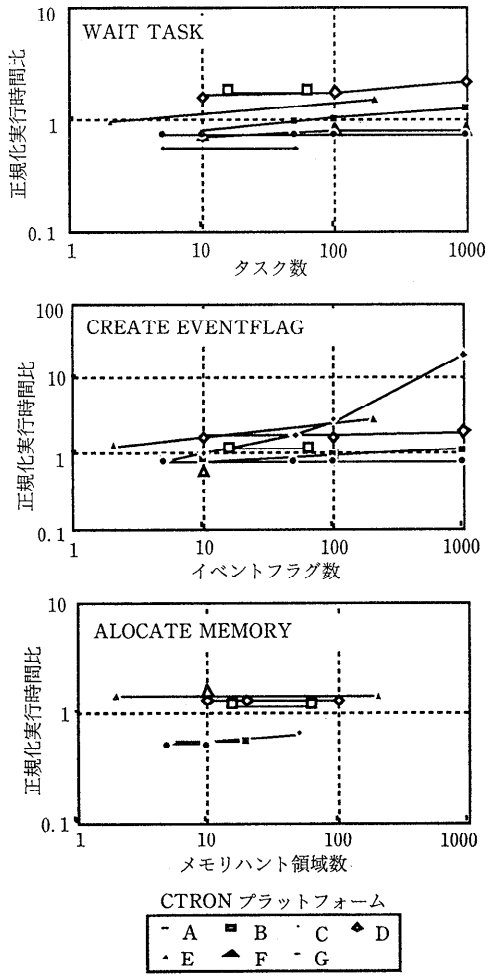


図-5 CTRON の OS 機能要素の測定

可能であることを確認した．なお，図-5は各プラットフォームの使用プロセッサ，コンパイラが異なる．これらの差を補正するため，Dhrystoneベンチマーク値[14]により正規化(Dhrystoneベンチマークのループ回数で補正)し，さらに各プラットフォームの平均値で正規化したものである．

また，リアルタイムUNIXとの比較結果を**表-1**に示す．表-1の値も，Dhrystone値で正規化した値である．CTRONはリアルタイムUNIXに比べてどの領域においても高速であることが分かる．

3.3 リアルタイムシステムの評価

CTRONの適用分野における代表的な処理モデル（交換処理など）を設定し，ベンチマークプログラムから得られたOS機能要素データからアプリケーションシステムでのOS性能を推定する手法について検討している．

アプリケーション処理モデルaで発行する各システムコールの回数を要素とするベクトルをMaとし，ベンチマークプログラムにより測定されたOSbの各システムコールの処理時間を要素とするベクトルをPbとすると，これらの内積Ma・Pbにより，アプリケーション処理モデルaでのOSbのOS処理時間が算出される．

たとえば，交換処理モデルでは，タスクでの処理分担のパターンを二通り作り，システムトータルな評価に用いている．このモデルを用い，CTRON準拠のOSとリアルタイムUNIXベースのOSとの比較をこの指標を用いて比較した結果を**図-6**に示す．

リアルタイムUNIXのOS処理時間はCTRONに比べ約数倍以上大きいことが分かる．CTRONがリアルタイムUNIXより性能が良くなる理由は，表-1での性能差の要因に示したように，交換モデルAでは，タスク状態遷移処理およびタスク間通信の性能の良さが，交換モデルBではタスク状態遷移処理の性能の良さが効いていることが分かる．

このようにMa・Pbの要素を個々に比較することにより，適用アプリケーションを想定した場合に，OSの性能のボトルネックと改善項目の抽出が可能となる．

4. リアルタイム処理ソフトウェア流通技術

4.1 ソフトウェアポータビリティ評価実験

CTRONでは，ソフトウェアポータビリティに関するCTRON仕様の達成度を定量的に評価するため，トロン協会主催で1990年から1992年まで足掛け3年間にわたり，東京大学理学部情報科学科坂村研究室との共同研究として，ソフトウェアポータビリティ評価実験を実施した．実験には8社が参加し，移植対象ソフトウェアの規模はC

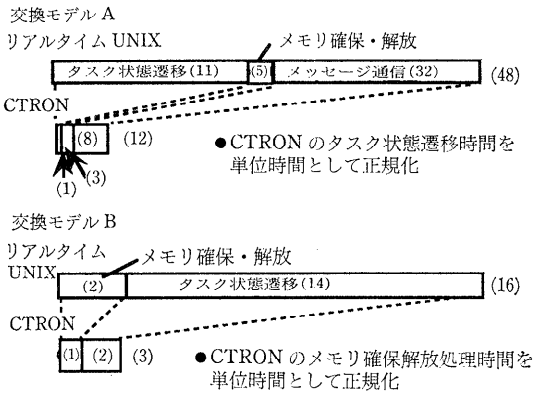


図-6 OS処理時間の分析

表-1 リアルタイムUNIXとの比較結果

評価項目		CTRON (μsec)	リアルタイムUNIX (μsec)	性能差の要因
タスク状態遷移処理（タスク生成・削除，コンテキスト切替え時間）		137	1500**	・タスクのメモリ常駐化 ・タスクコンテキストの軽量化
メッセージ通信		68	278*	ロケードモード(ポインタ渡し)
メモリの確保・解放		47	77*	固定サイズ・メモリブロック
割込み応答時間	旧コンテキストが割込みハンドラ	6	13*	ハンドラコンテキストの軽量化
	旧コンテキストがタスク	85	175*	タスクコンテキストの軽量化

ドライストーンベンチマークで正規化
* SSCL ベンチマーク値　** Task: Thread で評価

言語で約 600 k 行である[15)~21)].

（1） 実験の概要

CTRON は OS インタフェース仕様のみを規定しており，個々の製品のインプリメントは規定せず，インプリメントに委ねている．一般的に仕様書だけではポータビリティを保証することにはならない．このため，CTRON 仕様に準拠して開発された拡張 OS およびアプリケーションプログラム (AP) を，他社のシステム上に実際に移植する実験を行い，定量的にポータビリティを確認することにした．実験の目的は，ポータビリティ阻害要因の抽出と仕様へのフィードバック，移植の有効性の評価であり，次の評価項目を用いた（図-7）．

1） 移植に必要となるソース修正規模

2） 移植に必要となる工数

実験は，参加各社が自社内で開発し，自社の基本 OS 上で走行する拡張 OS を，実験場に設置されている他社が開発した異なるハードウェア上で走行する基本 OS 上に移植して，正常走行することを確認する．また，通信制御プログラムのポータビリティの確認は，ISDN 公衆網で相互接続された基本システム（ハードウェアと，その上で走行する基本 OS）間で，正常に通信が行われることを確認することにした（図-8）．

（2） 実験結果と分析

ソフトウェアポータビリティ評価実験でえられた結果を以下に述べる．

1） ソースプログラムのポータビリティ

ソフトウェアポータビリティ評価実験では，プログラムのポータビリティを評価する尺度の一つとして，

ソース修正率(%)＝(A/B)×100

A：移植に必要となるソース修正規模（行）

B：移植対象プログラムの母体規模（行）

を用いている．実験結果では，移植プログラム数：16，移植規模：600 k 行で，ソース修正率は，平均で約 4% であった．これは，母体ソースプログラムの 96% が流用可能であることを示しており，CTRON 仕様準拠拡張 OS のポータビリティが高いことが分かる（図-9）．

また，実験結果に基づき，仕様不備はもちろん，インプリメント依存部についても仕様化の検討を進めている．一方．仕様化できないものは移植性を向上させるガイドラインの整備を進めており，現状ではソース修正率 2% が達成できていると推定され，一層のポータビリティの向上が図られているものと考えられる．

2） 移植の有効性

移植の効果を判断する上で，移植工数を評価するのは重要である．移植の有効性の評価尺度として，同一ソフトウェアを新規に作成する場合の工数と，移植に要する工数を比較し，百分率で表現

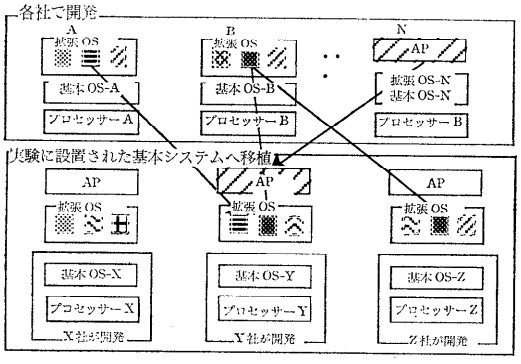


図-7 移植実験のイメージ

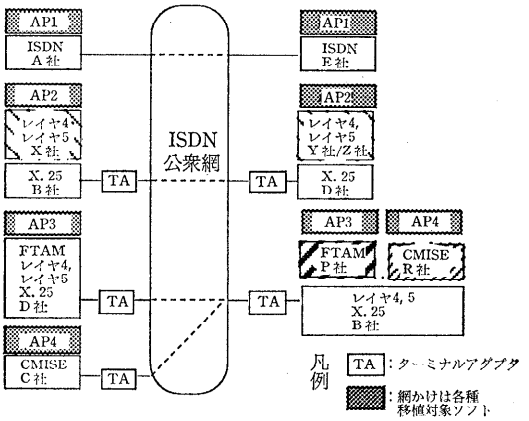


図-8 通信実験の構成

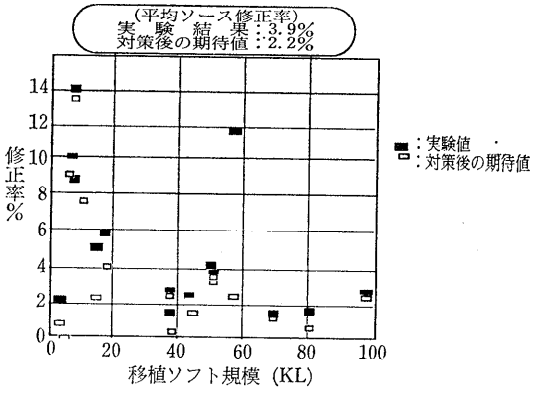


図-9 ソース修正率

することにした．移植工数の評価にあたっては，公平な比較を行うために以下の前提条件の下に評価を行った．

- ソフトウェアを移植する者（移植者と呼ぶ）は，当該ソフトウェアを開発した者（開発者と呼ぶ）と異なる．このため移植工数にはプログラム理解のための工数を含む．
- 移植者と開発者のプログラミングスキルは，同レベルとする．
- 移植したソフトウェアの確認試験は，通常のソフトウェアプロダクトの確認試験と同レベルとする．

実験結果では，移植工数は新規開発工数の26%であり移植の有効性が確認できた（**図-10**）．

また，開発者が移植を行う場合の移植工数は，移植者が開発者と異なる場合の半分で済むことが判明している．これは，ドキュメント上に移植時に必要な情報が提供されているかどうかが工数削減に重要であることを示している．

4.2 ポータビリティ確保のための考慮点

CTRON仕様に基づきインプリメントする場合にポータビリティを向上するための考慮すべき事項を詳述したものが，“原典 CTRON 大系　設計ガイダンス—ソフトウェアポータビリティ編—”[2]として出版されているので参考にされたい．以下に主要項目について述べる．

（1） 内部データ形式をもつプログラムの流通

OSは対象とする資源の管理や制御をするため一般に内部データ（制御表と呼ぶ）をもっている．この内部データが，複数の流通単位から共通に参照されると問題が発生する．たとえば，一般プログラム管理機能は，システムプログラム以外のプログラムを制御する仕様を規定したものであり，プログラムの構造，再配置情報，制御移行先のアドレス情報などが必要となる．これをプログラム制御情報と呼び，コンパイラ／リンカなどとプログラム管理が共通に参照する情報である．CTRONでは，アーキテクチャに依存する内容が多い，アーキテクチャを活かしたインプリメントを阻害するなどの理由からプログラム制御情報の構造／内容を規定していない．しかしながら，アドレス情報などのアーキテクチャに依存する機能部については，一般プログラム管理支援機能としてカーネル仕様として規定することにより基本OS内でアーキテクチャを隠蔽するとともに，標準的なインプリメント方法を例示している．

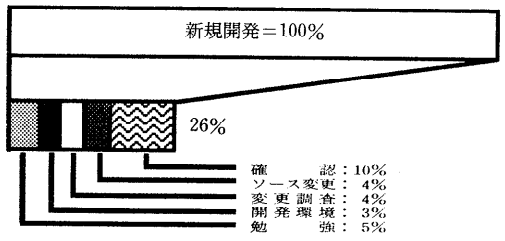


図-10 新規開発時との工数比較

（2） C記述プログラムの流通

CTRONでは流通用の言語仕様（ISO 9899 C仕様[22]）を規定しているが，高級言語で記述すればプログラムが流通するとは限らない．ポータビリティの高いプログラムを作成するには，以下の注意が必要であり，プログラミングガイドの整備を進めている[23]．

1） アーキテクチャに依存するコーディング

たとえば，変数サイズをプロセッサの暗黙のサイズと仮定したコーディングは避けるべきである．サイズを明示的に意識する場合は，sizeof 文により確認されたサイズでの使用とすべきである．

2） コンパイラ依存事項

たとえば，言語仕様では，コンパイラが受理する量的制約値（受理量）の最小値を定めているが，実際のコンパイラの量的制約値（受理量）はインプリメントにより異なるため，言語仕様の制御値以内で使用すべきである．

4.3 ポータビリティ向上サポートツール

プログラムのポータビリティは究極的にはプログラマのスキルに依存するところが大きい．このためには，プログラミングガイドを作成するだけでなく，ソースプログラム上で流通上問題になるコーディングがないか自動的にチェックするツールがあれば有効である．

実験結果では，確認工程が平均で移植工数全体の約40%を占めており，移植をスムーズに行うには，確認試験をいかに効率的に行いうるかが，キーポイントであり，機能確認用自動化ツールができれば工数削減効果が大きい．

このようなツールの研究も進められている．

5. あとがき

CTRONインタフェースに準拠した製品はすでに，20数件発表されており，これらは検定を受け，認定製品として具体的な高性能通信システムへの適用が進められつつある．

CTRONインタフェースは，現状においても通信システムの中核として適用可能であるが，今後も継続的に検討を進め，一層の適用性の向上を図るべく以下の課題について研究を進めていく．

（1） CTRONインタフェースはサブセッティングの概念で整理されているが，さらに適用性を向上させるためには，所要機能のプロファイルの一層の整理を図り，スリムな適用からフルセットの適用までを，性能と互換性を確保しつつ適用可能とする．

（2） 現在の言語適用条件は，国際標準準拠のC, CHILL言語などを対象としているが，今後のオブジェクト指向言語への対応を図り，ソフトウェアのモジュール化，部品化へ資する．

（3） 分散OSとしての適用性の向上を図るため，オブジェクト指向概念を用いた分散制御インタフェースの拡充を図る．

（4） リアルタイム性の評価指標の体系化を図り，他のOSとの比較のための正規化手法を含めた一般的なOS機能要素，システムとしての評価方式を確立する．

（5） ソース修正率0%に向けたポータビリティ向上のため仕様の詳細化（サービス依存性があるものは選択的仕様化）を進めるとともに，設計ガイドおよびプログラミングガイドの整備に取り組む．

（6） ポータビリティの確保のためには，検定技術やポータビリティ確保のためのツール群の整備が必須であり，これらの技術の確立と，ツールの整備を図る．

参考文献

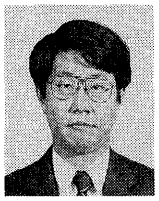
1) 坂村監修：原典 CTRON 大系　第1～第10編 CTRON 概説，カーネルインタフェース編，入出力制御インタフェース編他，(社) トロン協会編，オーム社 (1994).
2) 坂村監修：原典 CTRON 大系　別冊1，2編 設計ガイダンス“フォールトトレランス編”，“ソフトウエアポータビリティ編”，(社) トロン協会編，オーム社 (1994).
3) 野口監修，大久保著：図解 CTRON 入門，オーム社 (1990).
4) Ohkubo, T. et al.: Configuration of the CTRON Kernel, IEEE MICRO, pp. 33-44 (Apr. 1987).
5) ヴァーホル，P.D.：MS や IBM が模索するマイクロカーネルの姿，NIKKEI BYTE MARCH 1994, pp. 158-170.
6) 特集“いま始まる仮想マシン OS の時代” NIKKEI BYTE APRIL 1994, pp. 94-122.
7) 和佐野他：マルチベンダ指向リアルタイムソフトウエアプラットフォーム，電子情報通信学会論文誌 B-I, Vol. J76-B-I, No. 3, pp. 274-282 (1993).
8) Takenaka, I. and Oda, H.: The CTRON Interface Validation System, TRON Project 1989, Springer-Verlag, pp. 171-181 (1989).
9) Furtht, B. et al.: Real-Time UNIX Systems Design and Application Guide, Kluwer Academic Publishers (1991).
10) Low, K. et al.: Overview of Real-Time Kernels at the Superconducting Super Collider Laboratory, IEEE Proc. of Particle Accelerator Conference (May 1991).
11) Takahashi, T. et al.: Performance Evaluation by CTRON Kernel Benchmark, Proc. of the 10th TRON Int. Symp., 1993, pp. 156-164, IEEE CS Press.
12) Yamamoto, K. et al.: A CTRON Kernel Benchmark, Proc. of the 9th TRON Int. Symp., 1992, pp. 185-193, IEEE CS Press.
13) Kurosawa, H. et al.: An Evaluation Method of Kernel Products Based on CTRON, TRON Project 1990, pp. 191-200, Springer-Verlag.
14) Weicher, R.P.: Dhrystone Benchmark: Relational for Version 2 and Measurement Rules, ACM SIGPLAN Notices, Vol. 23, No. 8 (1988).
15) Ohta, T. et al.: Evaluating Software Portability in CTRON, IEEE 2nd Int. Workshop on Software Reusability (1992).
16) Ohta, T. et al.: CTRON Software Portability Evaluation Experiment, TRON Project 1990, pp. 133-148, Springer-Verlag (1990).
17) Ohta, T. et al.: Software Portability in CTRON, Proc. of the 8th TRON Project Int. Symp.,IEEE CS Press, pp. 86-92 (1991).
18) Ohtaka, M. et al.: Portability Evaluation for CTRON General File Management, Proc. of the 8th TRON Project Int. Symp., IEEE CS Press, pp. 93-102 (1991).
19) Nishimura, C. et al.: Portability Experiment for CTRON General Management, Proc. of the 9th TRON Project Int. Symp., IEEE CS Press, pp. 137-145 (1992).
20) Mochida, Y. et al.: Portability Experiment for CTRON Communication Control (Transport Layer and Session Layer), Proc.: The 9th TRON Project Int. Symp., IEEE CS Press, pp. 146-153 (1992).
21) Wakano, M. et al.: A Study on the Portability of CTRON FTAM-CCL Interfaces, Proc. of the

9th TRON Project Int. Symp., IEEE CS Press, pp. 154-161 (1992).
22) ISO: Programming Languages-C ISO/IEC 9899 (1990).
23) Iwata, N. et al.: Porting Classes and Coding Rules for C Programs on CTRON, Proc.: The 9th TRON Project Int. Symp., IEEE CS Press, pp. 144-155 (1993).

(平成6年3月31日受付)

大久保利一（正会員）

1946 年生. 1969 年早稲田大学理工学部電気通信学科卒業. 1971 年同大学院修士課程修了. 同年日本電信電話公社（現, NTT）電気通信研究所入所. 以来, DIPS オペレーティングシステム／通信制御ソフトウェア, DEX オペレーティングシステムの研究開発に従事. 現在リアルタイム OS インタフェース／OS 性能評価技術／コンフォーマンス技術の研究に従事. 電子情報通信学会, IEEE 各会員.



曽根岡昭直（正会員）

1957 年生. 1980 年東京大学電子工学科卒業. 1982 年同大学院修士課程修了. 1989 年工学博士. 1982 年日本電信電話公社(現, NTT) 横須賀電気通信研究所入所. 以来, 分散システム, 通信ソフトウェア開発環境の研究開発に従事. 現在NTTサービス生産本部にてマルチメディアサービス開発に従事. 1989～1993 年米国コーネル大学計算科学科客員研究員. 電子情報通信学会, IEEE, ACM 各会員.

花沢　満（正会員）

1945 年生. 1969 年電気通信大学電子工学科卒業. 同年日本電信電話公社(現, NTT) 電気通信研究所入所. 以来, DIPS オペレーティングシステム及びリアルタイム OS の研究開発に従事. 現在, ソフトウェアポータビリティの研究に従事. 電子情報通信学会会員.