

オンラインデバッグ支援システム “HITEST-DEMO”

Online Debugging Facilities for Industrial Computer System “HITEST-DEMO”

計算機制御ソフトウェアのオンラインデバッグ作業は、システム規模の拡大と複雑化に伴い困難さを増しており、システムの早期稼働を確保するためにも、有効な支援ツールの出現が待たれていた。

日立製作所ではこの要請にこたえるため、「オンライン環境での正確な稼働情報(機能・性能)を収集・解析し、見やすいドキュメントとして出力する」オンラインデバッグ支援システム“HITEST-DEMO”を開発した。

“HITEST-DEMO”は既に実システムに広く適用され、特に、組合せテスト以降の複雑なソフトウェアトラブルの原因究明と対策に必要なオンラインデバッグ支援ツールとして定着している。

大島啓二* Keiji Ôshima
林 利弘* Toshihiro Hayashi
本多 健** Takeshi Honda

1 緒 言

オンライン用計算機システムのソフトウェアのデバッグ作業では、「組合せテスト以降」に発見された不具合の原因究明と対策が最も難しいと言われる。システムの構成要素の数が増加するため、原因から現象に至るまでの道程が長く、かつ錯綜してくるからである。特に計算機制御システムではリアルタイム性能の確保のため、タスクと呼ばれるプログラムの単位を同時並行動作させることが普通であり、不具合原因の発生タイミングの特定すら難しいことも多い。

こういったオンラインリアルタイム環境下でのデバッグ作業では、従来から担当者個人の「現象の裏に潜む真の原因を探り当てる深い洞察力」と「長年にわたり積み重ねられた経験」が頼りとなり、他の作業(プログラミング、テストなど)ほど機械化・自動化が進んでいなかった。

しかし、制御レベルが高度化し、制御範囲が拡大している現在、デバッグ効率の低下は、システムの早期稼働に対する最大の隘路となるばかりか、「ソフトウェアに起因するトラブルによりシステムの操業率を低下させる」といった重大な問題を招きかねない。特にソフトウェア比率の拡大とともに、エンジニアがカバーしなければならない範囲が広がっている昨今、このようなオンラインデバッグを効果的に支援するツールの出現は強く望まれていた。

“HITEST-DEMO”(Hitachi Integrated Test System-Determine Evaluation, Monitoring and Output System)は、こういった要求にこたえるために、従来機械化が困難とされていた計算機システムのデバッグ作業の近代化を目指したデバッグ支援ツールであり、先に開発したテスト支援システム“HITEST”の1サブシステムとして位置づけられるものである^{1),2)}。

2 開発の基本思想と方針

ソフトウェアのオンラインデバッグ作業の特質と問題点は次に述べるとおりである。

(1) デバッグ作業は、テストの後段階にゆくほど難しくなる。

対象範囲が空間的に広がるだけでなく、不具合発生時点の特定という時間的要素が絡んでくるからである。

(2) 性能不具合の原因究明と対策が長期化する。また、見掛けの現象だけにとらわれた安易な対策が、思いもよらない副作用をひき起こすことがある。

(3) 操作ミス・確認ミスによる手戻り作業が多くなりがちである。運用開始間近で時間的制約が厳しく、作業者の精神的負担も増加する。

(4) システムを停止させてのデバッグは困難である。複数の人間が計算機を共有し、操業状態に近い形でテストを行なうことが多いからである。

以上のような「オンラインデバッグ作業」の特質に対し、現状のデバッグ支援ツールは、

- (1) 機能デバッグ用として取得できる情報が、メモリダンプ、簡単なトレースといった低いレベルにあり、
- (2) 情報収集・分析手順が複雑で、ミスを誘発しやすく、
- (3) 性能デバッグに有効なツールが少ない、

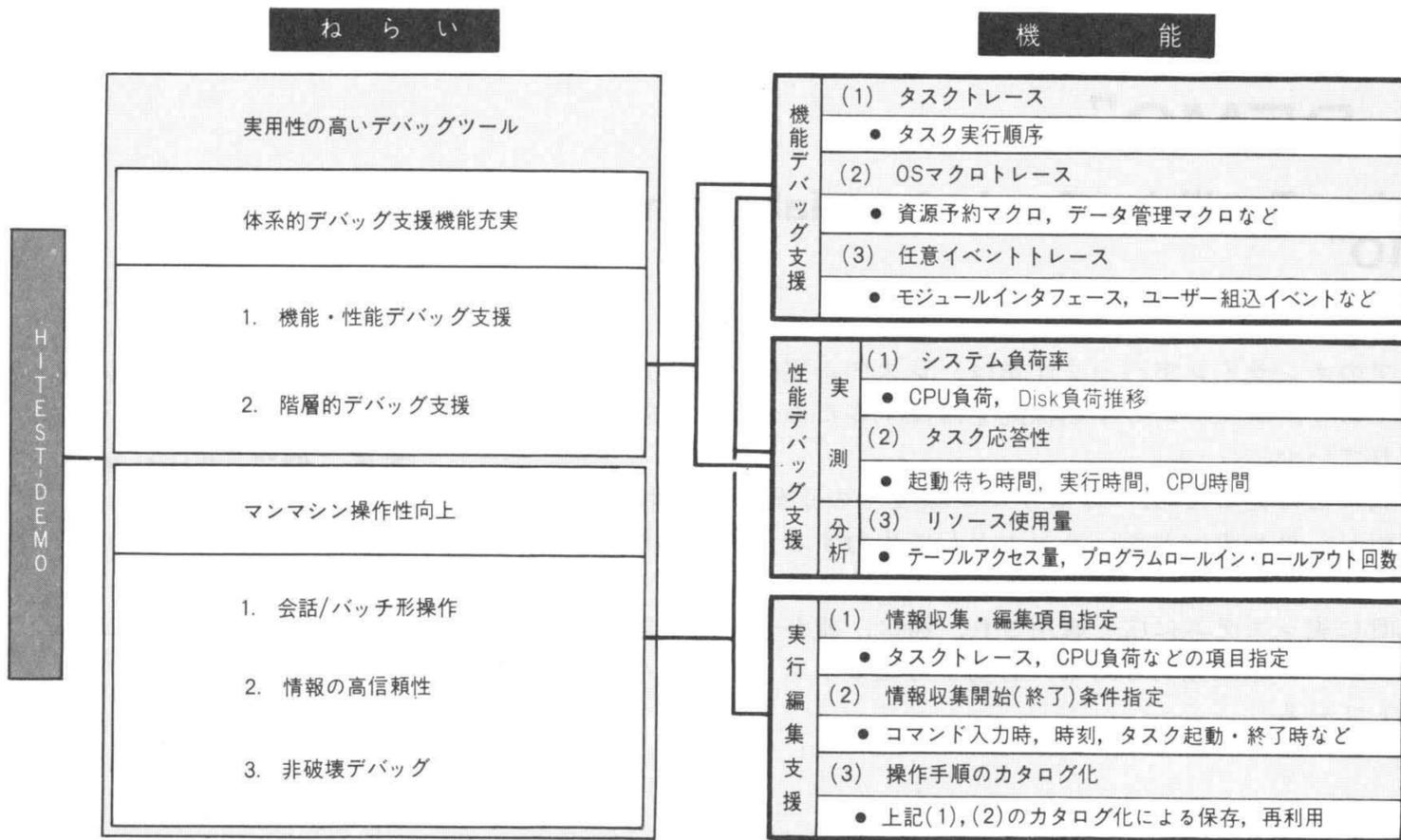
など、効率的なデバッグを行なう上で多くの問題があった。

以上の背景を踏まえ“DEMO”開発の基本方針を以下のように設定した。

- (1) 豊富なデバッグ支援機能を体系的に具備したツールとする。そのために、
 - (a) 機能デバッグ・性能デバッグ両者を支援できること。
 - (b) 状況把握(マクロレベル)から原因究明・対策立案(ミクロレベル)までを階層的に支援できる情報収集機能をもっていること。
- (2) マンマシン操作性に優れたデバッグ支援ツールとする。そのために、
 - (a) トラブルの内容・状況により「会話」、「バッチ」のいずれかの最適なオペレーションを自由に選択できること。
 - (b) 信頼性の高い情報が、見やすい形で出力されること。
 - (c) 「非破壊デバッグ」が可能であること。

デバッグ作業は、ソフトウェア開発で最も泥臭い部分の一

* 日立製作所大みか工場 ** 日立プロセスコンピュータエンジニアリング株式会社



注：略語説明 HITEST-DEMO(Hitachi Integrated Test System-Determinate Evaluation, Monitoring and Output System) OS(Operating System), CPU(中央処理装置)

図1 “HITEST-DEMO”のねらいと機能 “DEMO”は、「機能・性能デバッグ支援機能」を備えているだけでなく、これら機能を柔軟に使いこなすための「実行・編集支援機能」が充実している。

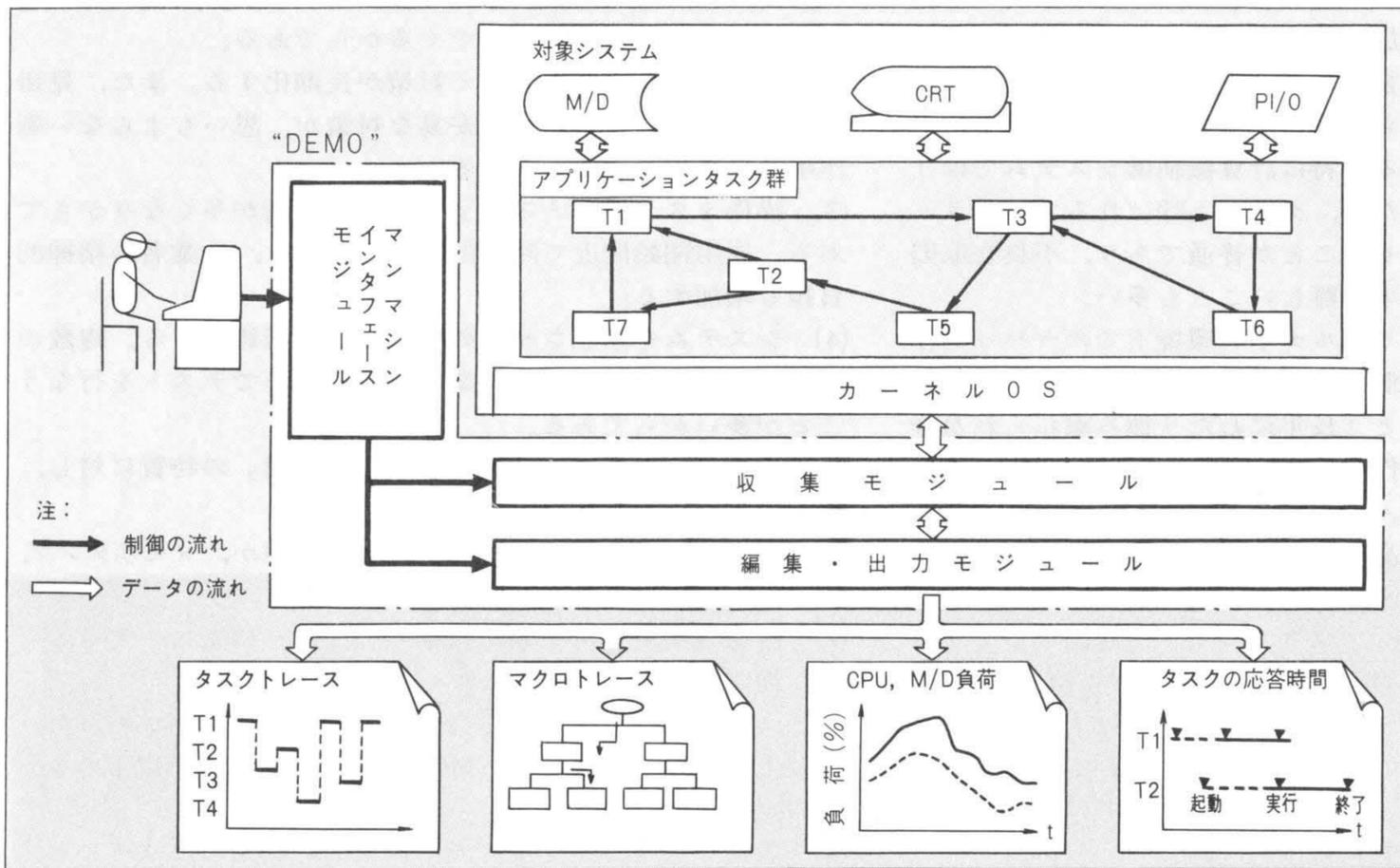


図2 “DEMO”のソフトウェア構成 “DEMO”は「収集モジュール」、「編集・出力モジュール」及び「マンマシンインタフェースモジュール」の三つから構成される。「収集モジュール」、「編集・出力モジュール」の最適処理分担により、“DEMO”稼働によるオンラインシステムへの影響を極小化している。

つである。手軽に、安心して使え、結果に対する信頼性が高いという「実用性」こそ、この種のツールが成功するポイントである。以上述べた“DEMO”のねらいと機能を図1に示す。

3 ソフトウェア構成

“DEMO”のソフトウェア構成を図2に示す。

「データ収集モジュール」は、オンライン稼働情報を収集し、収集バッファに蓄える部分である。情報収集時のオーバーヘッドを最小化するため、ここでは情報の加工は極力行なわないようにしている。

「編集・出力モジュール」は、収集バッファに蓄積された情報を加工し、簡潔な「機能・性能デバッグ支援レポート」を出力する部分である。

「マンマシンインタフェースモジュール」は、上記「データ

収集」、「編集・出力」に対する項目設定、条件指定をバッチ又は会話形で簡単に行なえるようにする部分である。「“DEMO”を使用しない場合のカーネルOS(Operating System)からの切離し」も本モジュールにより簡単な操作で行なえるようになっている。

4 支援機能

“DEMO”の支援機能一覧は図1に示すとおりである。

4.1 機能デバッグ支援機能

機能デバッグの支援のために“DEMO”は、「組合せテスト」以降のオンラインデバッグに焦点を当て次の3レベルの「トレース機能」を備えている。

(1) タスクトレース

並行動作するプログラム(タスク)群について、タスク間に

予期せぬ動きがないかどうかをマクロな視点で判断するのに有効である(図3)。

(2) OSマクロトレース

資源予約マクロの発行状況, I/O(入出力)機器のアクセス手順など, プログラムのソースリストからだけでは得られない「システムとしてのダイナミックな動き」が詳細に確認できる。

(3) 任意イベントトレース

標準パッケージとアプリケーションプログラムとのインタフェース部などに, 必要に応じてトレース用の割出し点を組み込んでおけば, インタフェース部の通過トレースを採取することができる。トレースの要否は簡単なコマンドにより, タスク単位に切り替えることができる。

(2)及び(3)を組み合わせて出力した例を図4に示す。

4.2 性能デバッグ支援機能

性能不具合に対しては, 「性能実測による現象の定量的把握」と, 「現象を構成する性能要素の分析」の二つをうまく組み合わせることが肝要である。

“DEMO”はこのことを踏まえ, 性能デバッグ支援として以下の機能を備えている。

(1) システム負荷率

CPU(中央処理装置)及びDiskなどの補助記憶装置の負荷率とその時間的推移を実測し出力する機能により, システム全体の性能余裕を精度高く確認できる(図5)。

(2) タスク応答性

マンマシンの応答性やプロセス追従性を確認するには, 「タスク応答性一覧表」(図6)が利用できる。これによれば, タスクの「CPU時間」だけでなく, I/Oアクセス時間を含めた

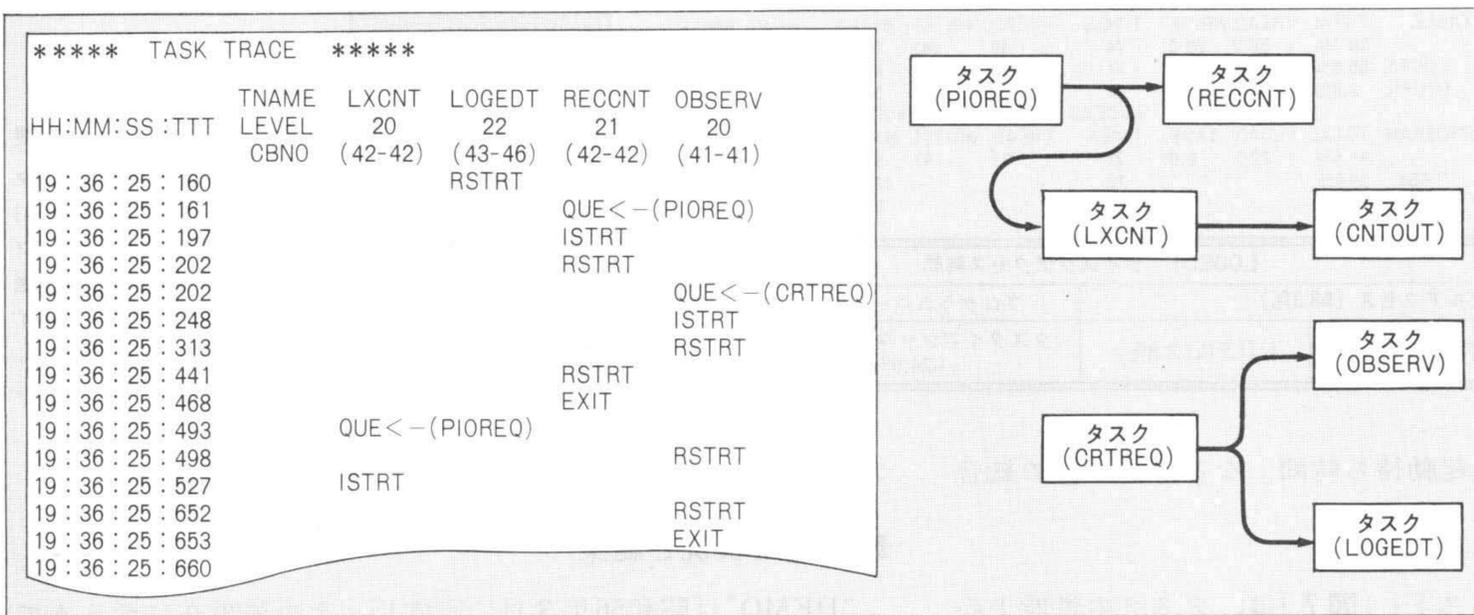


図3 タスクトレース
タスク優先レベル, 動作空間などにより, 並行動作するタスクの実行順序が決まる様子が視覚的に分かる。

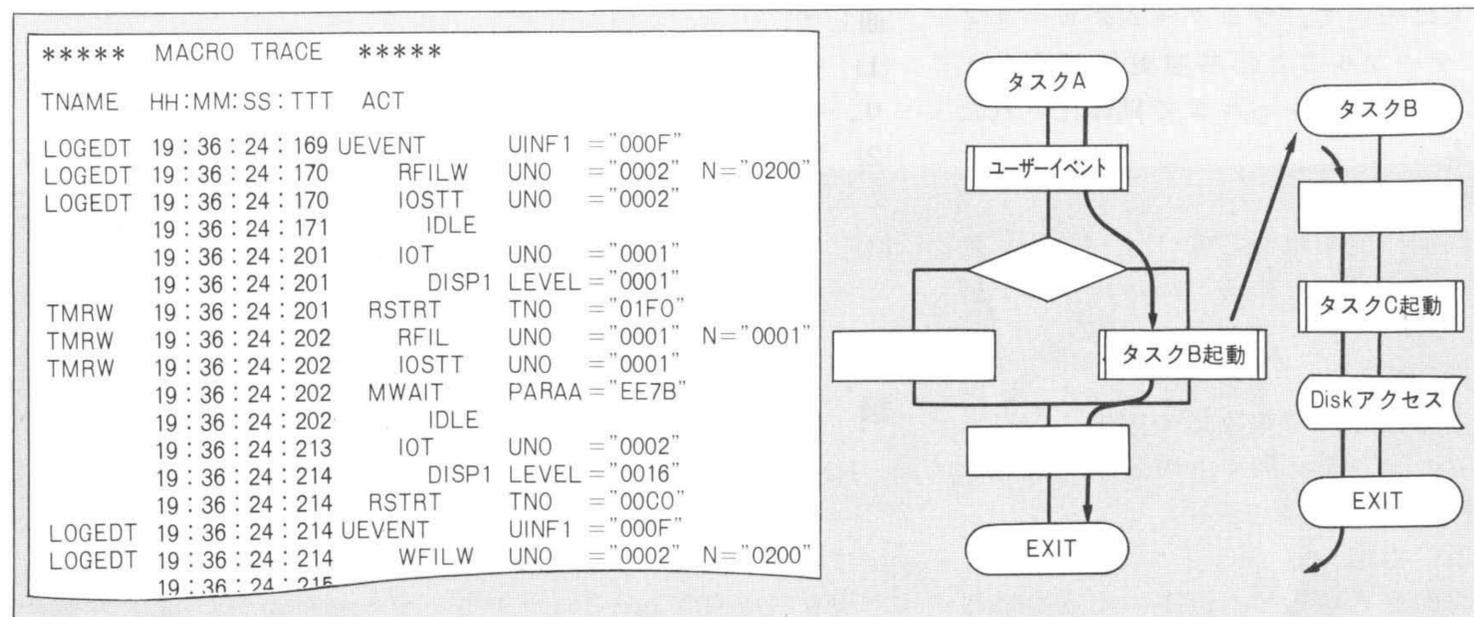


図4 マクロトレース・イベントトレース
ユーザー組込みイベント, タスク制御マクロ, データ管理マクロ, その他の区分によりインデントーションが付けられ, リスティングされる。タスクの動き, I/Oアクセスの動きを詳細に追跡する場合に有効である。

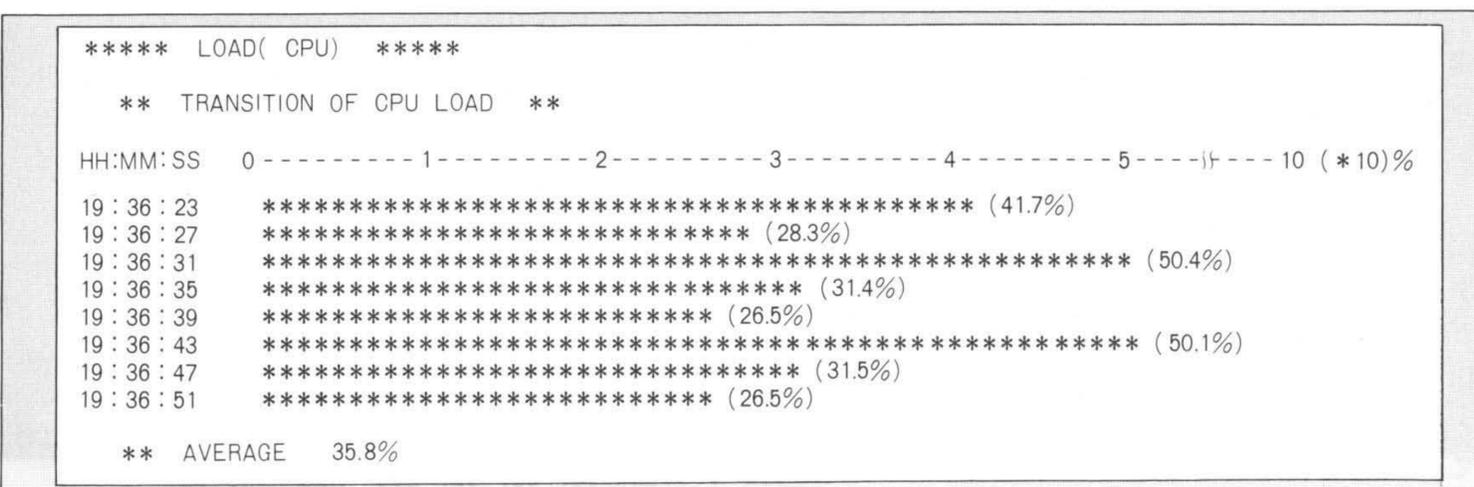


図5 CPU負荷 システムの負荷特性, 負荷余裕が定量的に分かる。補助記憶装置についても, デバイスごとに同様の出力ができる。測定時間帯, 刻みを任意に変えることができる。

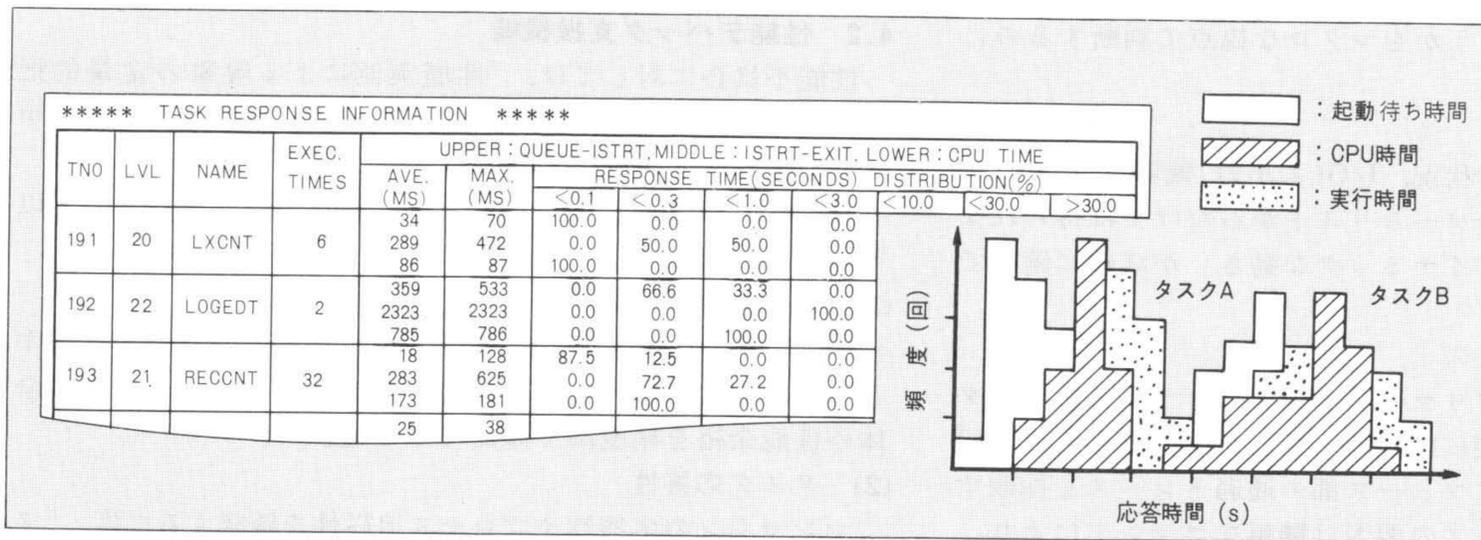


図6 タスク応答性一覧表
タスクごとに起動要求から初期起動、初期起動から終了に要した時間、及び所要CPU時間を出力する(平均, 最大, 分布)。

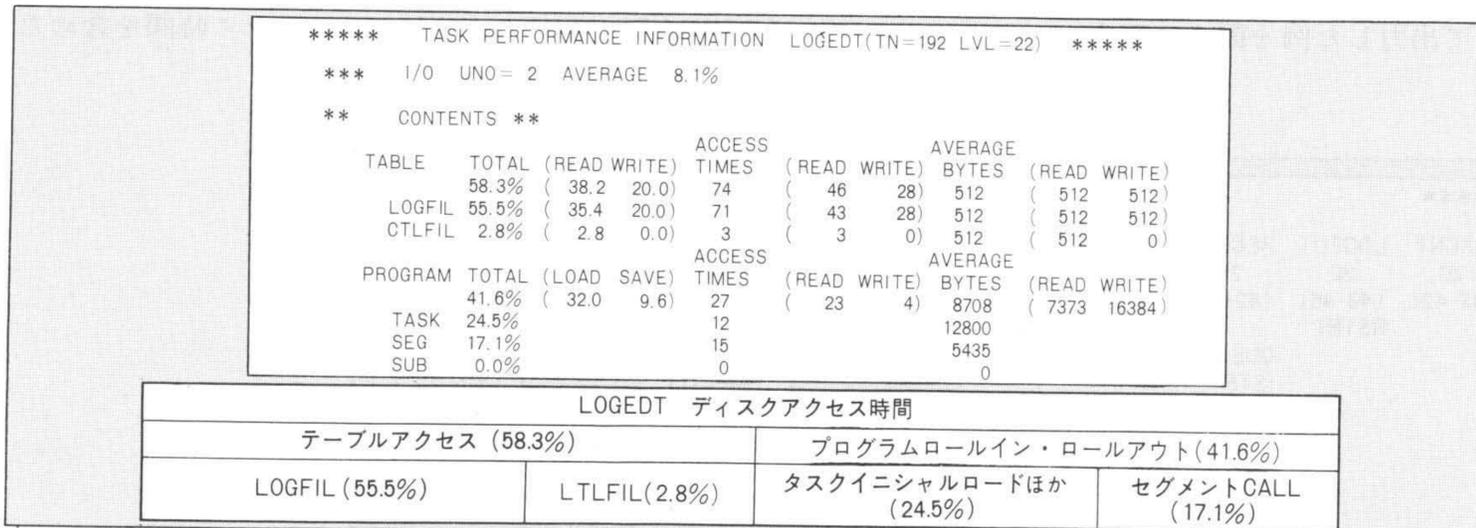


図7 タスクリソース使用量
タスクごとにそのタスクが使用したリソース(I/O)の使用量を出力する。テーブルのアクセス頻度、平均転送語数やプログラムのロールイン・ロールアウトの回数など、きめ細かな分析が可能である。

「実行時間」や、「タスク起動待ち時間」など、タスクの総合応答性能が鳥観できる。

(3) リソース使用量

「リソース使用量分析リスト」(図7)は、タスクの性能上の隘路を分析するために使用できる。本機能を用いれば、例えば磁気ディスクのアクセスについて、プログラムのロールイン・ロールアウト回数や、テーブルごとの参照更新頻度など「的確な改善策を講ずることができるレベルまで具体化された情報」を得ることができる。

4.3 実行・編集支援機能

先の4.1, 4.2に述べた個別機能を臨機応変に使い分け、効率的で信頼性の高いデバッグを可能とするため、以下の支援機能を備えている。

(1) 情報収集・編集項目指定機能

出力するドキュメントの単位で、情報収集や編集の項目を取捨選択することができるので、不要なドキュメントを出力せずに済み、また計算機時間も節約できる。

(2) 情報収集開始(終了)条件指定機能

デバッグ作業の種々の局面を考慮して、以下のような情報収集開始・終了条件の指定が可能である。

(a) 収集開始条件

コマンド入力時、指定時刻、指定タスク起動時、プログラムからの「開始マクロ」発行時、などの収集開始条件の設定ができる。

(b) 収集終了条件

コマンド入力時、指定時刻、指定時間経過後、指定タスク終了時、プログラムからの「終了マクロ」発行時、プログラム異常発生時、などの収集終了条件の設定ができる。

(3) 操作手順のカタログ化機能

使用頻度の高い「収集・編集項目指定」や「条件指定」の操作手順をカタログ化して、繰返し利用できるようになって

おり、デバッグ作業が効率的かつ信頼性高く行なえる。

5 適用状況と結果

“DEMO”は昭和56年3月に完成し、その後現在に至るまでに数多くのシステムに適用を行なってきた。これらの適用を通して、

- (1) 得られたデータの信頼性が高いので、試行錯誤がなくなり、安心して不具合原因の追究と対策ができるようになる。
 - (2) 特に性能不良については、対策の結果が定量的かつ即座に入手できるので、着実な改善実績が挙げられる。
 - (3) 将来の増設に対する性能余裕が正確に把握できる。
- といった良好な結果を得ている。最近では、デバッグ用だけでなく性能検証用としての使われ方も多くなってきている。

6 結 言

HIDICシリーズのオンラインデバッグ支援ツール“DEMO”について、その開発方針、構成と機能上の特長、更に適用状況について述べた。

現在“DEMO”は、「ソフトウェアの挙動を目に見える形に映し出す」という特長によって、ソフトウェアのオンラインデバッグに必要な不可欠のものとなっている。

今後は適用経験をフィードバックし、更に使いやすいツールとなるように、継続的な改良を加えていきたいと考えている。

終わりに、この“DEMO”の開発と適用に御協力をいただいた関係各位に対し、深く感謝の意を表わす次第である。

参考文献

- 1) 大島, 外: HIDIC V90/50ソフトウェア開発支援システム, 日立評論, 63, 12, 869~874(昭56-12)
- 2) 大島, 外: 制御用ソフトウェア機能一貫テストシステム“HIT EST/F”, 日立評論, 62, 12, 893~898(昭55-12)