クロスデバッガ用グラフィカル・ユーザインタフェースの構築

-- VDB-GUIDE(Graphical User Interface for VDB Debugging Environment) --

1K−5

村山 直樹*, 須藤 さゆり*, 平松 健司*, 高橋 真希*, 岩本 雅幸**, 古城 隆*

*日本電気(株) C&C共通ソフトウェア開発本部 基本システム開発部

**日本電気マイコンテクノロジー(株) 共通ソフトウェア部

1. はじめに

近年, 組み込みソフトウェアの開発において, クロス開発のホストマシンとしてワークステーション(EWS)を使用するケースが増えつつある. その背景としては, 従来のパソコン(PC)をホストとした開発に対し, EWSを利用した場合には分散ファイルシステムの機能によって開発ファイルをネットワーク上のマシンで共有でき, 大規模のソフトウェア開発にも適していること, また, ネットワーク機能により分散した環境での開発が可能であること等が挙げられる.

このような状況において, 我々はEWSが持つもう一つのファシリティであるグラフィックス機能に着目し, グラフィカル・ユーザインタフェース(GUI)を利用したデバッグ機能の実現法及びその効果について研究を行っている.

本発表では, 我々のグループで開発中のクロスデバッガ用グラフィカル・ユーザインタフェースについて報告する.

2. クロスデバッガにおけるGUIの可能性とその効果

2.1 GUI実用化の背景

次のような状況から, 今後クロスデバッガにおいてGUIが実用化されていく可能性がある.

(1) ウィンドウシステムの普及と高性能化

X Window[1]に代表されるようなグラフィック・ウィンドウシステムがEWSに標準的に装備されるようになりつつあり, 一方, ウィンドウ対応の端末の低価格化も進行している. また, 処理性能面でもかなりの高速化が実現されてきている.

(2) ホストマシン-ターゲットマシン間の通信の高速化

GUIにおいては多種の情報を同時に表示するため, ターゲットとの高速な通信が必要であるが, デバッグ対象である組み込みシステムにおいて, 近年, ネットワーク機能を組み込むケースが増加する傾向にあり, そのようなシステムの場合, ホストマシンとターゲットシステム間の通信が従来のシリアル回線接続に比して格段に高速化される.

2.2 GUI利用により期待される効果

(1) グラフィカルな表示による視覚効果

複雑な構造を持ったデータを視覚的に見易くレイアウトしたり, カラーやアイコンによって情報を認識しやすく表示することができる.

(2) ポインティング・デバイスによるオペレーション

デバッグ作業において関心のある部分を画面上でダイレクトに指定することができる. また, コマンドの実行も画面上のボタンやメニューの操作でおこなえるので, コマンド名やオプションについて覚える必要がない.

(3) 関連ツールとの併用

画面上に他の開発ツール(エディタや性能解析ツール等)のウィンドウも同時に開いて作業できるため, 使用中のツールを終了せずに関連ツールを使用できる.

3. VDB-GUIDEのGUI機能

現在我々はマルチタスクデバッガVDB[2]対応のGUI(VDB-GUIDE)をEWS上のX WindowとOSF/Motif[3]を使用して開発中である. 本節では, VDB-GUIDE開発にあたっての基本的な方針について説明する.

3.1 デバッグ対象オブジェクトのビジュアル化

3.1.1 デバッグ対象オブジェクト

プログラムの変数とかタスク間通信に使用するセマフォなどのようにターゲットプログラムを構成している「物」のことを「デバッグ対象オブジェクト」と呼ぶことにする.

デバッグ対象オブジェクトは, ターゲットプログラムとの関係で見た場合, それぞれが状態(値)をもった部品であり, それらの全ての状態の組合せがターゲットプログラム全体の状態と考えることができる.

一方, デバッグ作業の観点から見た場合には, デバッグ対象オブジェクトは表示や設定の際の最小単位であり, デバッグ作業は全てのオブジェクトについて値の正当性を確認すること, あるいは不正な状態を持ったオブジェクトが存在した時にその状態を引き起こす原因となったプログラム・テキストを発見することであると捉えることができる.

図1は, VDB-GUIDEが対象とするデバッグ対象オブジェクトの一覧である.

プログラムオブジェクト	OSオブジェクト
ファイル	タスク
関数	セマフォ
システムコール	イベントフラグ
変数	メイルボックス
	メモリプール
マシンオブジェクト	**(デバッガオブジェクト)**
メモリ	ブレークポイント
レジスタ	デバッガステータス
I/O	

図1 デバッグ対象オブジェクト

3.1.2 オブジェクトのビジュアル化

グラフィックスによる視覚効果を生かすためにVDB-GUIDEにおいては, オブジェクトの表示に対して以下のような表示形式を提供している.

On Graphical User Interface for Cross Debugger.
Naoki MURAYAMA*, Sayuri SUTOU*, Takeshi HIRAMATSU*, Maki TAKAHASHI*, Masayuki IWAMOTO**, Takashi KOJO*
*NEC Corporation.
**NEC Microcomputer Technology,Ltd.

[1] X Windowは, 米国マサチューセッツ工科大学の登録商標.
[2] NEC 32ビットVシリーズ・マイコン用のクロスデバッガ.
[3] OSF/Motifは, OSFの商標.

(1) 構造化データのグラフィックス表示
構造体や配列はもとより，それらがツリー状やリスト状にリンクされている場合にでも，それらのデータの各要素を2次元的にレイアウトして表示する形式である．

(2) サマリ表示
ターゲットシステム内に同一種類のオブジェクト(例えばタスク)が複数存在する場合に，それらのオブジェクトを一覧表形式にして表示するものである．表中には，オブジェクトの主要な情報のみを抜粋して掲載する．
図2は，タスクサマリ表示の例である(各タスクの状態(state)は，色と形の異なるアイコンで表示し視覚的に判別しやすいようにしている)．

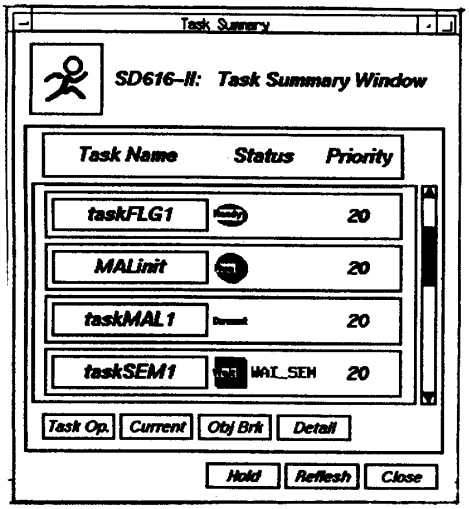

図2 タスクサマリの表示例

3.2 デバッギング・オペレーション

開発中のVDB-GUIDEでは，デバッグ操作はできるだけマウス・オペレーションで実行できるようにすることをひとつの開発方針にしている．デバッグ時の主な操作は，表示関連の操作，設定関連の操作，実行関連の操作の3つに大別できるが，ここではVDB-GUIDEの表示関連と設定関連の操作法について述べる．

3.2.1 表示オペレーション

VDB-GUIDEでは表示オペレーションに2種類の状況を想定している．

ひとつは，あるオブジェクトの状態を観察している時に，それと関連している他のオブジェクトについても状態を表示したくなる場合であり，例えばセマフォの状態を表示してみたらその中にタスクが待ちキューに繋がれている様子が表示されていたので，次にそのタスクの状態を表示したくなった，等のような状況である．VDB-GUIDEではそのような状況に対応するため，ある表示ウィンドウ中での別オブジェクトの選択操作は，そのオブジェクトに対する表示指示であると解釈するようにしている．例えば上記の例では，キューイングされているタスクはそれぞれがMotifのボタン・ウィジェットに対応しており，そのボタンをプレスすることによってそのタスクの情報を表示できるようになっている．

もうひとつの状況は既に表示中のオブジェクトについて別の形式の表示を行いたい場合であり，例えば表示されているソーステキストに対応するディスアセンブルを表示したいといった状況がこれに該当する．VDB-GUIDEでは，表示されている対象オブジェクトのウィンドウ上で表示変換機能を選択(ボタン/メニュー)することにより，別形式の表示を行えるようにしている．

3.2.2 設定オペレーション

設定オペレーションとは，ある変数の値を変更するとか，あるタスクを強制終了するなどのように，オブジェクトに対してある種の状態変化を引き起こすデバッグ操作のことである．実行可能な設定オペレーションの種類はオブジェクトの種類によって異なるが，VDB-GUIDEでは，

対象オブジェクトの選択 → 設定オペレーションの選択

というオブジェクト選択優先の手順で設定を行うようにしている．実行可能なオペレーションは各オブジェクトの表示ウィンドウにボタンまたはメニューとして割り付けている．

4. 実現上の問題点

4.1 処理速度/レスポンスの高速化

VDB-GUIDEではオブジェクトの表示ウィンドウ上に，関連オブジェクトの表示を導くためのボタンや，オペレーションを指示するためのボタンなどの様々なボタンを配している．そのため，多い場合には1ウィンドウを構成するウィジェットの数が数十個になることもあり，表示の準備にかなりの時間がかかってしまうことがある．そこで我々のシステムでは，GUI処理の空き時間(例えばターゲットへのダウンロードの間やターゲットのイニシャライズ処理が実行している間)を利用して，生成に時間のかかるウィンドウは表示要求がある前に予め準備しておくように設計を行っている．

4.2 表示ウィンドウの管理

VDB-GUIDEでは，約30種類のウィンドウが表示可能であるが，画面上に表示されたウィンドウの数が多くなり過ぎて見易さを損ねるのを防ぐために，以下のような方針により表示ウィンドウの管理を行っている．

- 同一種類のウィンドウは，同時に2つ以上表示しない．
- 表示時に，すでに同一種類のウィンドウが表示されていた場合には，そのウィンドウ上に上書きする．
- ただし，状態比較のために古い表示内容を保存したい場合を想定し各ウィンドウにホールド(貼付け)機能を備える．
- ホールドが指示されない限り，表示は実行に合わせて更新する．

5. おわりに

今回発表のシステムでは，スタティックな情報表示が中心であるが，今後はダイナミックな状態表示(具体的にはOSオブジェクトの状態変化表示やターゲットの動作のサマライズ表示など)の実現についても検討も進めていきたいと考えている．また，現在は32ビットVシリーズ・マイコンをターゲットにして開発を行っているが，今後は各種チップ/各種組込みOSへの展開も進めていく予定である．

参考文献

[1] 橋本他,"32ビットVシリーズ用ソフトウェア開発環境",情報処理学会第56回マイクロコンピュータとワークステーション研究会,(1989).