

ターゲットに依存しない操作性をもつデバッグ環境の構築

6R-5

橋 晋哉, 上田 裕一, 渡邊 豊

(富士通株式会社)

1. はじめに

我々は、プログラム構成図から構成定義～ソース編集～翻訳結合～デバッグの作業を、ツールを意識せずに自然なビューでプログラム開発ができるCプログラマ向けの環境——「Cプログラマブラウザ環境」^{1,2)}と呼ぶ——を構築した(図1参照)。これは社外及び社内のユーザの開発に適用してもらっており、高い評価を得ている。

ところで、Cプログラム開発においては、プログラムを複数のターゲット・システムへ移植する作業が多い。通常ターゲット・システムが異なると、ツールの操作性も異なり、特にデバッグ作業ではその差が大きく、プログラマに非常に負担がかかる。ここでは、Cソースレベルのデバッグ作業が対象である。

本稿では、上記の問題を解決するために、ターゲット・システムに依存しない操作性をもつデバッグ環境を、我々の構築したCプログラマブラウザ環境において実現できたので紹介する。

2. ターゲットに依存しない操作性のデバッグ環境

ターゲット・システムに依存しない操作性のデバッグ環境の実現に当たって、我々は、先ず流通している主なデバッガを調査・分析してみた。その結果、これらのデバッガは、基本的に次の4つの機能を共通にもつことに気がついたが、これらの操作性は、当然のことながらデバッガごとにバラバラであった。

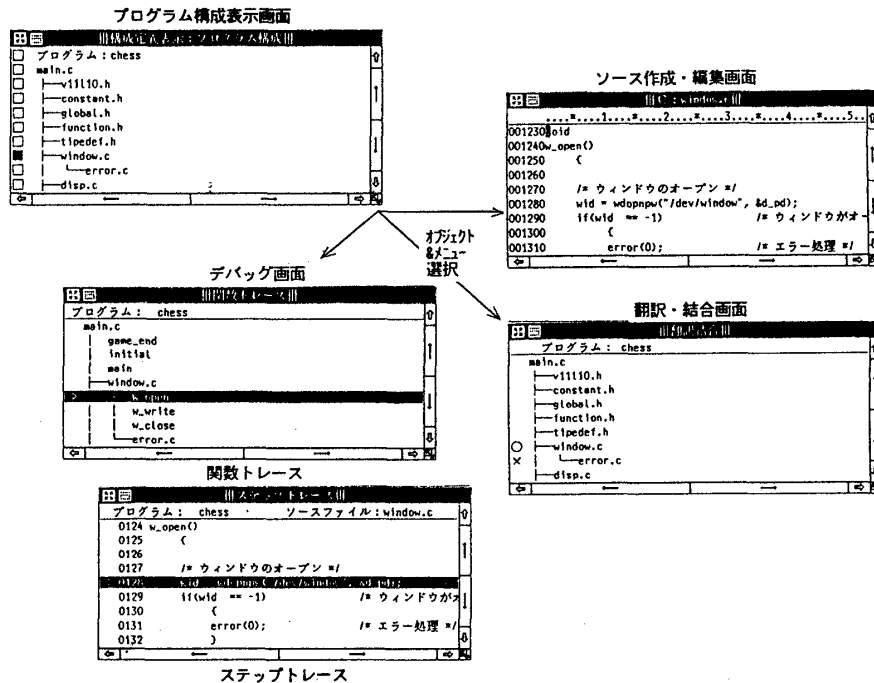


図1. Cプログラマブラウザの概要

Debugging Environment Independent of Target System

Shinya Tachibana, Yuichi Ueda, Yutaka Watanabe

Fujitsu, Ltd.

- ①ソースの任意の行位置で中断できる。
- ②ソースの任意の箇所を実行できる。
- ③ソースの任意の箇所を表示できる。
- ④任意の変数値を表示/変更できる。

しかしながら、これらは本来ターゲット・システムに依存しない操作性を実現できるものである。一方、デバッガは、どうしてもターゲット・システムに依存する部分をもつ。

そこで、ターゲット・システムに依存しない操作をプログラマに提供する共通な部分と、ターゲット・システムに依存する部分から成る開発環境を構築することにした(図2参照)。そして、徹底してターゲット・システムに依存しない使い勝手をもつ共通な操作性を実現した。

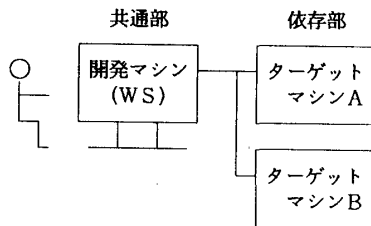


図2. Cプログラマブラウザの動作環境

3. デバッガの構成

我々のデバッガは、次に示す2つから構成される。

共通部: デバッガの操作を制御する部分で、ターゲット・システムに依存しない。

依存部: 各ターゲット・システム上の被デバッグ・プログラムを制御する部分で、ターゲット・システムに依存する。

ここで重要なのは、共通部と依存部のインタフェースをターゲット・システムに依存しない共通なものにすることである。例えば、前述した基本的な機能に対しては、図3のようにした。

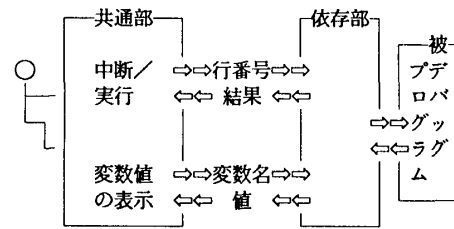


図3. 共通部と依存部のインタフェースの例

ここで難しいのは変数値のインタフェースであるが、変数値を表示形式で受けとることにより、共通部が変数の型などを意識しなくて済むように工夫した。

4. 効果

ターゲット・システムが違ってもプログラム開発環境の操作が共通であると、その環境に慣れたプログラマが、容易に他のターゲット・システムに対する開発ができるため、プログラマの負担が軽減し、開発効率が向上する。特に、移植時には、その効果が大きい。

また、現在新しいターゲット・システムへの対応中であるが、その対応作業は依存部のみであり、短期間で作業が完了する見込みである。

5. 今後の課題

我々の環境は、当社の独自ウィンドウ上で動作しているが、環境の拡大を目指してX-Window上へ展開中である。

また、現在対象としている言語はCとY P S (Yac II Programming System)であるが、C++の追加を考えている。

参考文献

[1]上田, 笠原, 渡邊:「プログラム構成図をベースとしたプログラム開発環境」, 第37回情報処理学会全国大会論文, (1988)

[2]山本, 笠原, 黒木, 渡邊:「アシスト機能を備えたプログラム開発環境」, 第39回情報処理学会全国大会論文, (1989)