ＰＳＳ：Ｃ言語プログラムの移植支援システム―概要

4Ｆ－7

西風　一*　原田　稔*　小林　純一*　前田　忠彦**　大浜　美雪*

*沖電気工業株式会社　**株式会社沖ソフトウェア中国

1．はじめに

近年オペレーティングシステムの標準化が進み、異機種間結合のシステム構築が日常的になるにつれて、アプリケーションプログラムの異機種間移植性の問題が注目されるようになった(1)。

ＵＮＩＸの統合や、各コンピュータメーカが発表しているアプリケーションアーキテクチャは、今後、移植性の低いプログラムが生産されないための歯止めをもくろんでいる。しかし、そのような意図を持たずに既に開発されてしまった膨大な量のソフトウェア資産が、オープンシステムへの移植を待っているという現状である。

ＰＳＳ(the Porting Support System)は、Ｃ言語で記述されたプログラムを、ＣＰＵアーキテクチャの異なる機種に移植する際に生じる問題の解決を支援する統合化された環境である。本稿では、ＰＳＳの全体構成、およびヒューマンインタフェースについて概観し、その成果について報告する。

2．移植性阻害要因

プログラムの移植性阻害要因とその対応策を表－1にまとめる。ソフト的な問題は静的解析による指摘によって、ソースそのものの移植性を高めることによって解決できる。この方法はプログラムの詳細を知らなくても作業できるため普及している。

一方、ハード的な問題は、ソフトウェアによってＣＰＵアーキテクチャをシミュレートすることによるバイナリレベルの移植で解決できる。しかし、作業工数が少なくてすむ反面、性能、拡張性で問題があり、あまり普及していない。このような方法を使わずに移植する場合、移植作業者が問題点を解析、修正するために品質面、効率面で大きな問題がある。

表－1　移植性阻害要因

分類	阻害要因	概略	対策例
ハードウェア	ｴﾝﾃﾞｨｱﾝ	2進ﾃﾞｰﾀをﾒﾓﾘ上に格納する際のﾊﾞｲﾄｵｰﾀﾞの差異	ﾊｰﾄﾞｳｪｱ ｼﾐｭﾚｰﾀ
	ｱﾗｲﾝﾒﾝﾄ	CPU高速化を目的とするﾃﾞｰﾀ語境界整列規則の差異	
ソフトウェア	OSの差異	ｼｽﾃﾑｺｰﾙのｲﾝﾀﾌｪｰｽ等の差異	XPG3 ﾍﾟﾘﾌｧｲｱ
	中間ｿﾌﾄの差異	画面、ﾌｧｲﾙ等I/O関連をｻﾎﾟｰﾄする共通ﾊﾟｯｹｰｼﾞの差異	移植用中間ﾊﾟｯｹｰｼﾞの開発
	ｺﾝﾊﾟｲﾗの差異	ｺﾝﾊﾟｲﾗの文法、解釈、制限の差異	lint、ANSICﾁｪｯｶ

3．ＰＳＳのソフトウェア構成

ＰＳＳは、前節で述べたさまざまな移植性阻害要因の解決を統合的に支援する。以下にそのソフトウェア構成と基本的アプローチを示す。

（1）チェッカによる大域解析

一般に、エンディアンの相違に起因するプログラムの問題箇所は、複数のコンパイル単位にまたがる解析によってのみ指摘され得る。ＰＳＳによる解析は、ソースファイル単位の局所診断と、各ソースファイル毎の局所診断情報を併合して行われる大域診断の２段階で行われ、問題箇所の指摘がなされる（図－1）。

（2）コンパイラによる阻害要因吸収

ＰＳＳは専用のＣコンパイラを用意しており、拡張構文（プラグマ指令）によって、エンディアンの相違やアラインメント違反の吸収を行う。この版のＰＳＳは、沖電気のワークステーション、ＯＫＩｓｔａｔｉｏｎ７３００をターゲット機としており、コンパイラもｉ８６０の目的コードを生成する。

（3）エディタによるナビゲート

ＰＳＳのエディタは、チェッカによる解析結果に基づき、複数ファイルにまたがる問題箇所をナビゲートしながら、作業者によるソース修正を促進する。

（4）操作支援による統合された環境

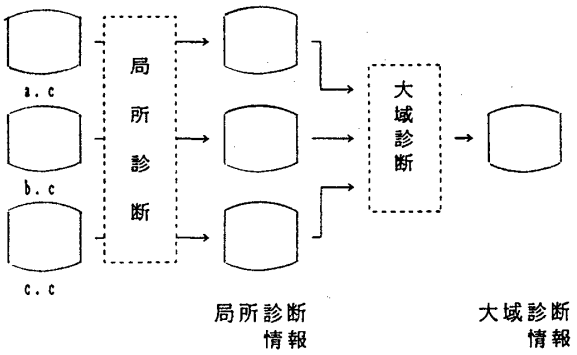
ＰＳＳ操作支援は、上記各ツールを統合し、ウィンドウベースの操作環境を提供する。操作支援は、この他に以下のような多様な補助機能を持ち、トータルな生産性向上を実現する。

- マクロ展開表示機能
- 版数管理機能
- 作業進捗状況表示機能
- マルチユーザ同時アクセス制御機能

チェッカ及びコンパイラについてはそれぞれ(2)、(3)で詳述される。本稿では、以下、エディタ･操作支援の環境下で移植作業者がどのように作業を進めて行くかについて報告する。

4．ヒューマンインタフェイス

本節では通信電文を編集する処理プログラム例（図－4）を考える。この例では send_messageで確保した出力バッファの一部を sel_itemで intとして扱っ



図－1　チェッカによる大域解析

UNIXｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑは、UNIX System Laboratories, Incが開発しﾗｲｾﾝｽしています.

PSS: The Porting Support System for C-programs - An Overview
Hajime NISHIKAZE, Minoru HARADA, Junichi KOBAYASHI,
Tadahiko MAEDA and Miyuki OHAMA
Oki Electric Industry Co., Ltd, Oki Software Chugoku Co., Ltd

ている。このため、この部分の実行結果はマシンのエンディアンに依存して異なってしまう。

ＰＳＳを使用した問題点の解決手順を以下に示す。

（1）ソース解析

操作支援からチェッカが起動され、エンディアン、アラインメントに関する問題点が収集される。

（2）問題点表示

エディタが起動され、問題点が発生した経路が表示される（図－2）。また、経路を選択すると対応するソースが表示される。

（3）問題点修正

移植作業者が問題点とソースを見比べ、修正の必要ありと判断した場合、修正位置と方法を指示する。これによりソースは自動修正される（図－3）。問題はある一点にあるのではなく経路にあるので、どこを修正するかは移植作業者にゆだねられる。

（4）問題点承認

修正の必要なしと選択した場合、承認済みマークが自動挿入される。これによって次回のチェック時、この問題点は指摘されない。

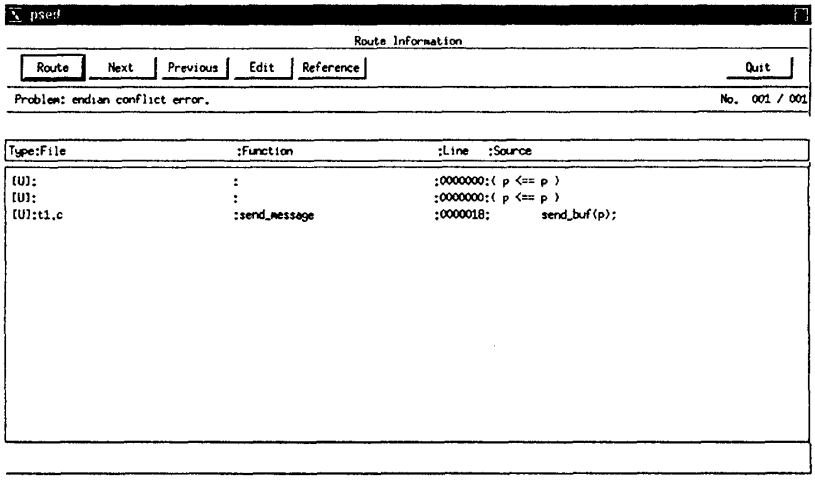
（5）コンパイル

チェック済みソースに対し、継承コンパイラが起動される。継承コンパイラはプラグマ指令を解釈し、エンディアン、アラインメントを意識した実行形式を生成する。

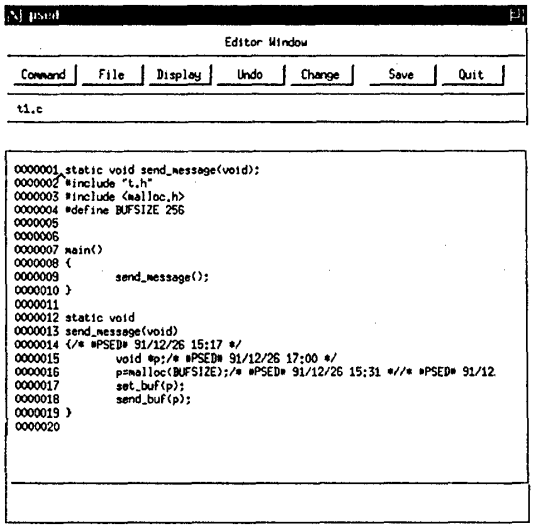
（6）作業進捗状況の把握

ＰＳＳは複数作業者による移植作業の進捗レポートが作成できる。

このように、ＰＳＳは移植作業のすべてのフェーズに対し、一貫したサポートを行っている。



図－2　経路表示ウィンドウ



図－3　編集用ウィンドウ

5．評価

ＰＳＳは現在、第1版の開発がほぼ完了したところであり、実際のソフトウェア移植プロジェクトで使用された実績は無い。しかし、試用の結果次のような意見が出ている。

・チェッカによる指摘が細かすぎ、膨大な量になってしまうことがある。
・類似の目的に対し異なる操作が必要など、操作性に弱い点がある。
・移植プログラム上の着目すべき点は漏れなく指摘されている。

6．おわりに

Ｃ言語プログラムの移植のための統合的な支援環境であるＰＳＳについて概観した。今後、さらに
・移植先・元について汎用性がある
・操作性が高く
・移植後の性能劣化が無い
システムへ向けて改良を重ねる予定である。

［文献］
(1)ﾍｲｽﾞ：「ｿﾌﾄの移行性確保には標準への準拠が不可欠」、日経ｺﾝﾋﾟｭｰﾀ、1991年6月3日号、pp. 75-77
(2)前田、他：「PSS:C言語ﾌﾟﾛｸﾞﾗﾑの移植支援ｼｽﾃﾑ-矛盾指摘ﾁｪｯｶ」、本予稿集、4F-06
(3)西川、他：「PSS:C言語ﾌﾟﾛｸﾞﾗﾑの移植支援ｼｽﾃﾑ-継承用ｺﾝﾊﾟｲﾗ」、本予稿集、4F-07

```
void
send_message(void) {
    void  *p;
    p = malloc(BUFSIZE);
    set_buf(p);
    send_buf(p);
}
```

```
void
set_buf(void *p) {
        ...
    set_item(p + OFFSET);
}
```

```
void
set_item(int *p)  {
        ...
    *p = data;
}
```

図－4　プログラム例