

4V-5

リアルタイムOSにおける  
母国語支援環境 (NLS) の開発

大島 輝洋, 武部 達明, 星 哲夫  
(横河北辰電機株式会社)

1. はじめに

筆者等は、HP1000 RTE-AリアルタイムOS上に日本語を包含する母国語支援環境(NLS, Native Language Support)を米国HP社と共同で開発したので報告する。NLSは、多国語支援環境とも換言出来、同時に最大7ヶ国語を支援する。現在支援可能な言語は、西欧系の言語13ヶ国語と日本語である。従来HP3000上で、8ビット・コードで実現可能な西欧系言語の支援を行っていたが、HP1000では、さらに16ビット・コードを使用する日本語の支援を可能にした。NLS実現上の問題は

1. メッセージ機能
2. コード体系に関連した問題
3. 言語・習慣に依存した問題

以上の3つに分類可能だが、以下にこの分類にしたがって実現手段を明らかにする。尚NLS化されたシステムは、一切ソース・コードを変更することなく他言語へ変換可能となる。

2. 開発方針

開発に当って、以下の方針を設定した。

- ・RTE-Aは、リアルタイムOSであることから、NLS化がシステムの性能に影響を与えないこと。
- ・RTE-Aが既開発のOSのため、NLS化OSは、後方向互換性を保証すること。
- ・他言語への拡張を容易にするため、テーブル・ドリブンとする。
- ・なるべくHP社の他機種との相互互換性を保つ。
- ・HP3000上の既開発の概念及びデータを極力利用して開発時間の短縮を図る。
- ・RTE-Aは、主記憶のみのシステムも許している。よってNLS化されたシステムも主記憶のみで動作すること。

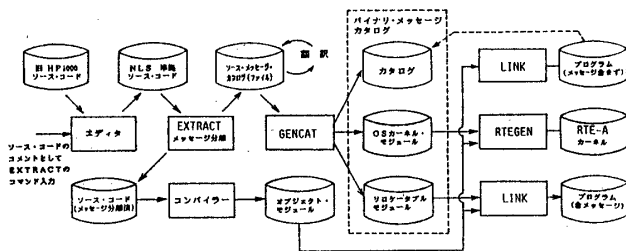


図3 コードとメッセージ分離の手順

3. メッセージ機能

ソース・コードとメッセージの分離

RTE-Aのメッセージは、一部のプログラムのみがメッセージのファイル化を実現しているのみで大多数のプログラムのメッセージはプログラム中にハード・コードの型で格納されている。そこでプログラムとメッセージの分離を行う2つのツールの開発を行った。これによりファイル化されたメッセージをエディタを使用して翻訳することによってソース・コードを変更することなく他言語の支援が可能となり、プログラマはソース・コードのみ管理すれば良い環境を構築出来る。

EXTRACT: ソース・コードからメッセージを抽出しファイルに格納する。又、メッセージ出力関数をNLS用の関数に置換する。

RTE-Aではメッセージを出力する手段が多岐に亘っているため自動変換・抽出はあきらめ、エディタでEXTRACTコマンドをコメントの型でソース・コードに記述させる方式とした。

GENCAT: EXTRACTを使用して抽出されたメッセージを実行時ルーチンが解釈出来る型に変換する。GENCATの入力はEXTRACTの出力ファイル、或は翻訳を行った後のメッセージ・ファイルである。

メッセージ格納先

- (1) リロケータブル・モジュール: メッセージはプログラムの1つのモジュールとしてリンクされる。
- (2) カタログ: メッセージはディスク上のファイルに格納される。
- (3) カーネル: カーネルの出力するメッセージは各国語毎にモジュールとしてまとめられる。多国語を同時に支援するので複数のモジュールをカーネルはリンクする。

以上3つの格納先を選択可能とした。格納先はEXTRACTの入力ソースプログラム作成時に属性コマンド(\$atb)を指定することにより、GENCATに伝達される。

RTE-AのNLS化は、ほとんどがリロケータブル・モジュールで実現された。理由は、パフォーマンスと主記憶システムを考慮したためである。又、この方式はディスク・エラーからも独立しており最も望ましい方式といえる。図3にコードとメッセージの分離の手順を示す。

Native Language Support on Real Time OS  
Teruhiro Oshima, Tatsuaki Takebe, Tetsuo Hoshi  
Yokogawa Hokushin Electric Corporation

メッセージ合成時の順序制御

メッセージは静的なメッセージと動的なメッセージに別れこれらは実行時に合成されて出力されるが、言語によっては動的メッセージの順序を逆転させる必要がある。以下の方式によりこの順序制御を可能とした。

・メッセージ中の!n (1 < n < 9)を動的メッセージ・パラメータPnで置換を行う。

例えば格納されたメッセージが"! 2 am a ! 1."の場合に、出力ルーチンが

```
er = NLwriteString(Output ..., P1, P2)
```

でP1="boy" P2="I"であったとすると結果は"I am a boy."であるが、日本語で出力するときには"!1は!2."としてP1="少年" P2="私"として"少年は私。"と出力することを可能とした。

言語の切換

言語の切換はプログラムNLSIDとPATHを組合せることで可能となる。

NLSID: 各ターミナルに対応した言語識別番号(LANGID)を変更する。

PATH: RTE-AはUNIX流の階層型のディレクトリ構造を持ち、通常プログラムが格納されるディレクトリは、/programsと決っている。PATHによってこのプログラム・ディレクトリの変更が可能である。この機能を利用して/programsを/programsxxxに変更する。尚、xxxは3桁の言語識別番号である。

メッセージの格納先をリロケータブル・モジュールにした場合、言語毎に異なるディレクトリにプログラムが格納され、上記のプログラムPATHによってディレクトリを変更して言語の切換を行う。メッセージの格納先をカタログとした場合、指定された言語識別番号によってメッセージの格納されるファイル名をプログラム・ディレクトリ同様に変更を行って切換えを行う。

NLS関連の各種関数は、言語識別番号をパラメータに指定させ言語の切換えを行う。

上記の機能を利用して、一般に使用者がログオンを行った時に言語を切換える。

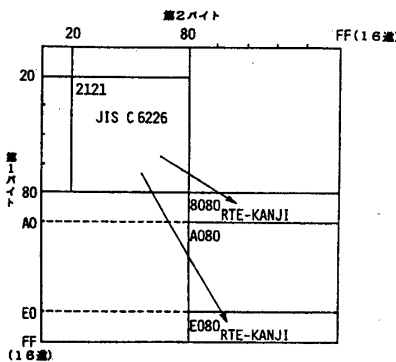


図4 RTE-KANJIコード

カスタム・フォーマット関数	文字ハンズリング関数
NI FmtClock	NI RepChar
NI FmtCustDate	NI ScanMove
NI FmtDate	NI Collate
NI FmtCalendar	NI Translate
NI ConvCalendar	NI KeyCompare
NI ConvClock	
Almanac	
NI Info	

図5 NLS用各種関数

4. コード体系に関連した問題

西欧系の言語は、USASCIIを8ビット・コードに拡張したROMAN8コード・セットによって処理される。ROMAN8は、ISOに準拠したHP社標準として各種周辺装置に使用されており、このROMAN8を使用する周辺装置をシステムに接続することによって英語以外の西欧系の言語の使用が可能となる。一方日本語は16ビット・コードを使用する必要があるが、RTE-AではシフトJISに似た図4に示すRTE-KANJIコードを内部処理コードとして採用することで、プログラムの変更なしで漢字が使用出来る様にした。仮名漢字変換は、当社で開発したターミナルで行う方式のためシステムの変更は必要としない。又、スクリーン・エディタもターミナルに依存したエディタの為、このターミナルを使用することでエディタの改造は不要であった。

5. 言語・習慣依存情報

言語及び習慣に依存した情報として特別に以下の項目を考慮する必要がある。

1. 日付・日時のフォーマットと表示文字列
2. Yes/No応答文字列
3. 小数点表示文字、千桁表示文字
4. 通貨記号とその位置、さらに通貨表示名
5. 照合順序(コレーティング・テーブル)
6. コード変換(ASCII → EBCDIC)

これらを言語毎に切換え可能とするために、図5に示す各種の関数を用意した。これらの関数を使用しているプログラムは言語が切換えた場合にも、ソース・コードを変更することなく指定された言語に対応した処理を行うことが可能となる。関数が必要とする各種の情報、例えば日付・日時のフォーマット等はすべてファイルLANGDEFでテーブル型式で管理されており、このLANGDEFファイルを変換することで容易に他言語への追従が可能となる。

6. 終わりに

以上NLSの実現手段を中心に述べたが、RTE-Aで実現された機能は、

1. RTE-Aのヘルプ、エラー、ガイドメッセージの母国語化(現在、翻訳を終っているのは日本語のみ)。
2. 日付、時刻表示型式の母国語化。
3. アプリケーション・プログラマーへのNLS各種関数の提供。
4. 同時に最大7ヶ国語迄切換え可能な環境。

以上である。これらの機能はシステムを世界市場に供給する場合不可欠なものと思うが、今後これらの機能が他のOS或は言語処理系に取込まれていくことを希望する。

【参考文献】

[1] Native Language Support Reference Manual, HP (Part No. 92089-90001)  
 [2] HP Journal, June 1985, Native Language Support for Computer Systems.