

## マイコン系基本ソフトウェア開発支援系—SAFEシステムの実現

小田 一博、吉田 信博、中原 康、北村 順次、高木 正彦、木下 恂\*

(株)東芝 青梅工場、\*東芝ソフトウェアエンジニアリング(株)

従来、基本ソフトウェアの開発は機種毎に異なった開発形態をとっていることが多く、それぞれの担当者間での技術交流もなかった。そのため共通の開発環境のもとお互いに切磋琢磨し、各種ノウハウの交流、技術力を高めあう機会が少なかった。

そこで、どの計算機上でも同一の開発環境下で基本ソフトウェアの開発を可能とするという基本理念のもとに、SAFE (Software And Firmware Development Environments) システムを企画し、現在構築中である。本稿では、特にマイクロコンピュータ(マイコン)系ソフトウェアを対象とするシステム(SAFE/MPと呼ぶ)について報告する。本システムは、ファームウェアの開発支援系としても利用できる。

Software Development Support System of Basic Software for  
Microcomputers--Implementation of the SAFE system

Kazuhiro ODA, Junji KITAMURA and Shun KINOSHITA et al.

Toshiba Co., Toshiba Software Engineering, Ltd.  
2-9 Suehiro-Cho, Ome-City, Tokyo 198, Japan

Up to this time, development of basic software has been done on different machines and by different support systems. Therefore, engineers could not communicate each other on the same technical bases.

And also, little chances has been given to each engineers for competing and know-how transfer with another engineers.

We are now developing the SAFE(Software And Firmware Development Environments) system which gives same development environments for software engineers.

In this paper, we reports SAFE system for microcomputers. We named this system as SAFE/MP(MicroComputers).

This system can also be used for the development of firmwares.

## 1. はじめに

従来、基本ソフトウェアの開発は機種毎に計算機の命令語が異なり、開発ツールや使用方法なども、それまでに発展してきた歴史も異なるため、それぞれ独自の道を歩んできた。それぞれの技術者がお互い議論することはあっても、根本的には理解しえないところが残ってしまうことが実態であった。そのため、共通の開発環境のもとでお互い切磋琢磨し、各種ノウハウの交換、技術力を高めあうということができなかったきらいがある。

SAFE/MPシステムでは、計算機が異なっても同一環境、同一ツールで基本ソフトウェアの開発ができることを目標に現在開発および評価中である。

また、ツールのみ揃えてもその使用方法や、ソフトウェア開発の手順などを知らないはどうしようもないことから、技術者の躰に関してもマニュアルを作成し徹底をはかっている。

## 2. 考え方の背景

開発環境を共通化できるようになってきた背景として、以下のことが挙げられる。

- ・プログラミング言語Cの普及
- ・UNIX風のオペレーティングシステム(OS)の浸透
- ・強力なパーソナルコンピュータの出現
- ・ソフトウェア技術者が今まで以上に世の中から注目されるようになった  
(開発期間の短縮、品質向上など)

### (1) ソフトウェアバスとしてのC言語

計算機アーキテクチャの統一とファミリ化により機種が整理される一方、基本ソフトウェア開発部門でもC言語のような共通言語が広く使われる時代になってきた。いわばC言語というソフトウェアバス(ハードウェアのバスになぞらえて)を通じて、ソフトウェア作りの共通の土壌を設けることができるようになってきた。

しかし、C言語でプログラムを記述すればどんな機種にも簡単に移植できると、考えては間違いである。最近のプログラム移植の経験から得られた結論は、「移植できるように作ったプログラムは移植できる」という極めて当り前のことであった。この当り前のことが出来ていないのが現状でありそれだからこそ何らかの改革が必要である。

### (2) UNIX流開発手法

UNIX環境は、ソフトウェア生産の場としては極めて秀れていると言われている。しかし、我々の今までのソフトウェア作りの環境とはあまりにも異なるので、素人にはなじめない、あるいは理解できないという反論を聞くことが多い。一部の玄人はうまく使いこなしているが、そのノウハウを広め誰でもが使いこなせるようにしむける努力が今まで欠けていたと言える。

### (3) 強力なパソコン

パソコンの分野でもCPUが強力になり、主記憶と補助記憶装置の拡充とあいまって、今ではオフィスコンピュータの上位に肉薄する程になってきている。マンマシンインタフェースなど使い易さの点では、UNIXツールも対抗できない程である。

このようなことから、従来のクロス開発環境を見直し、パソコン自身の上でソフトウェアを開発することが重要になってきている。

## 3. SAFE/MPの基本方針

このような背景のもとで、基本ソフトウェアの開発、あるいは今後益々増大していくであろうファームウェアの開発を、共通の環境下で管理し、統合し、効率良く進めるために考えられたのがSAFE/MPシステムである。

UNIX流の階層構造のファイルシステムを持つOS下なら、どんなOSの元でも実現できるように配慮されている。すなわち、SAFEシステムではホストOSを持たないシステムである。

逆に言えば、リホストの容易なシステムを、目指している。この環境下にUNIXとパソコンの持つそれぞれの特徴を生かしたツール類を集め、融合させ、ソフトウェアツール装備率向上を図ることにより、素手で計算機を扱うのではなく、道具を用いて(あるいはその道具からまた新しい道具を作りそれを用いて)ソフトウェアを開発することができるようにする。

また、特にパソコンのスケールが大きくなってきた実態を反映して、従来のクロス開発という考え方から、実機開発という現状に即した環境に切換えることにより、工期短縮を図ることを目指す。

SAFEシステムを移植するためのターゲットシステムに求められる環境条件としては、以下のものがある。

- (1) UNIX流の階層構造を持つファイルシステム  
(UNIX、MS-DOS、C-DOS等)
- (2) C言語が使用できる。
- (3) 補助記憶装置として、ハードディスクが使用できる。

あるSAFEシステム下で、ソフトウェアを開発する場合、そのソフトウェアが他のSAFEシステム上でも容易に動作することを目標に開発する必要がある。そうすることにより、ソフトウェアの移行性を実現するばかりでなく、同じ様な環境が機種毎に保証されているので、異なるシステム間でのソフトウェア技術者の移行も容易となり、要員の効率的運用が可能となる。

#### 4. SAFEシステムの具体化と運用

上記方針にもとづき、SAFEシステムを各実機にリホストしてみた例を表1に示す。ホストOSと実機のカテゴリ別に4つのシステムに分類できる。

生産性向上のためには、SAFE環境下でどのようにソフトウェアを作るかを明確にする必要がある。ベテランのプログラマの頭の中のみあった、いわゆるコツを整理統合化し、一般のプログラマにノウハウとして教え、普及させることも、SAFEの重要なテーマの一つになっている。

図1、表2に、ソフトウェアの開発の手順とその時使用すべきツールを示す。

SAFEシステムでは、ツールの他に文献[1, 2, 3, 4]で示すように各種マニュアルを作成し、教育によってこれらの考え方を徹底させることにより、移植性の高い、寿命の長いソフトウェアを、効率良く作れるようにした。

#### 5. おわりに

以上、SAFEシステムの概要を紹介した。今後はハードウェア、ソフトウェア技術の進歩に歩調を合せつつ、さらにシステムの統合化を図るべく、ネットワーク環境をベースとした基本的な道具の強化・拡充およびノウハウの洗練を図っていく予定である。

#### 参考文献

- [1] 木下、「Cプログラミング作法」、サイエンス社 1986.12
- [2] 木下、「移植しやすいプログラムの作り方」 Computer Today 1987/4別冊、pp11~31
- [3] 東芝社内資料、「C's beautiful programmer」 1987.3
- [4] 東芝社内資料、「貴方のCプログラムを長生きさせる法」 1986.6

図1. 開発工程とSAFEシステム

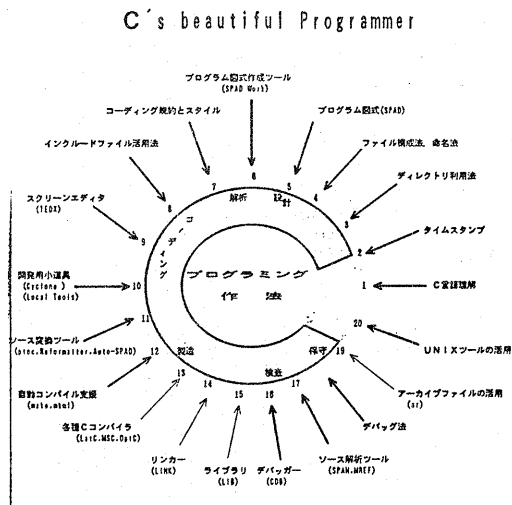


表1. 現在利用できるSAFEシステム

名 称	ホストOS	実 機
SAFE/T	英語MS-DOS	T2100 (HD付) T3100, T3500
SAFE/P	日本語MS-DOS	Pasopia1600 (HD付)
SAFE/J	日本語MS-DOS 日本語C-DOS	J5030 J3100 (日本語MS-DOS)
SAFE/UX	OS/UX	UX-300 UX-700 UWS

- \* UNIXは米国ベル研究所が開発し、所有権を有するオペレーティングシステムです。
- \* MS-DOSは米国マイクロソフト社の登録商標です。
- \* C-DOSは米国デジタルリサーチ社の登録商標です。

表2. 基本的道具とノウハウ

工程	項番	キーテクニック (ツール名)	SAFE	SAFE	SAFE	SAFE
			/T	/P	/J	/UX
共通	1	C言語理解	共通			
	2	タイムスタンプ(TOUCH)	共通			
	3	ディレクトリ利用法	共通			
	4	ファイル構成法	共通			
設計	5	プログラム図式(SPAD)	共通			
	6	プログラム図式作成ツール (SPAD Work)	-	-	○	-
製	7	コーディング規約とスタイル	共通			
	8	インクルードファイル活用法	共通			
	9	スクリーンエディタ(TEDX)	○	-	○	-
	10	開発用小道具(cyclone) (local tools)	○	-	-	-
	11	ソース変換ツール(REFORM) (Auto-SPAD)	○	○	○	○
		(PTOC)	-	○	○	-
	12	自動コンパイル支援 (make.mknf)	○	○	○	○
	13	Cコンパイラ	○	○	○	○
	14	リンカー	○	○	○	○
	15	ライブラリ	○	○	○	○
造	16	ソースデバッガ(CDB)	○	-	○	-
	17	ソース解析ツール(SPAN) (MREF)	-	-	○	-
	18	デバッグ法	○	○	○	○
	19	アーカイブファイルの活用 (AR.ARC)	○	○	○	○
	20	UNIXツールの活用 (SCCS 等)	○	○	○	○

○: 利用可能 - : 利用不可

図2. プログラム図式 (簡易PAD)

