

OS インターフェース標準化の動向

越 田 一 郎

東京工科大学 工学部

近年、ソフトウェアのポータビリティを高める手段として、OS インターフェースの標準化に対する関心が高まっており、各種の標準化作業が進行している。本報告では、現在提案されている各種の標準化案（POSIX, MOSI, X/OPEN, SAA等）について、その目的、基本的な考え方、および標準化の現状を概観する。特に、わが国がISOにNWIとして提案しているSSIについては若干詳細に述べる。

併せて、先頃バリで行われた ISO/TC 97 SWG on POSIX, SSI and Related Matters の会議の報告も行う。

STANDARDIZATIONS OF OPERATING SYSTEM INTERFACE

Ichiro KOSHIDA

Department of Information Technology, Tokyo Engineering University
1404-1 Katakura-cho, Hachioji-shi, Tokyo, 19 Japan

The portability of application programs is enhanced by standardizing interfaces between application programs and operating system (i.e. operating system interface). In recent years, several standards of operating system interface have been proposed. This report is a survey of these standards (POSIX, MOSI, X/OPEN, SSI, etc.). Especially, the SSI which has been proposed by Japanese Member Body of ISO is closely discussed. In addition, a brief report of SWG meeting on POSIX, SSI and related matters is included.

1. はじめに

近年、OSインターフェースの標準化に対する関心が高まりつつあり、各所でその作業が進められている。我が国でも、情報処理学会のOSインターフェース専門委員会を中心として、昨年度、「SSI (System Software Interface)」をNWIとして、ISOに提案した。本稿では、このSSIを中心として、OSインターフェース標準化の現状について概観する。

2. 各提案の概要

本章では、現在提案されているOSインターフェースの標準化案 (SSIを除く) について簡単に説明する。

2. 1 P O S I X

P O S I Xは、IEEEが提案しているUn i xに基づいた標準化案である。標準化の基本的な考え方は、とにかく使用可能な標準を作ることを、最大の優先事項と進められている。したがって、de factoスタンダードであるUn i xに極めて類似しており、事実上Un i xの標準化だけが目的であると入っても良い。そのため、対象とする計算機も従来Un i xを使用してきたワークステーション～ミニコンが中心となろう。

また、アプリケーション・プログラムを記述するためのツールである、言語に関しても、現在は、C言語だけを対象としている。実際、資料 [1] では、通常のアプリケーション・プログラムが使用するであろうシステムコールを全て規定しているわけではなく、Cの標準化案 [2] に含まれるライブラリ関数を標準に含めることを前提としている。

さらに、標準化作業のスコープもかなり限定されており、たとえば、shellのようなコマンド言語、ネットワークプロトコル、グラフィックスインターフェースなどは対象外となっている。もっとも、これは必ずしも欠点というわけではない。いたずらに対象範囲を拡大し、標準化を遅らせるよりは良いと考えられよう。また、従来のUn i xで問題とさ

れていたいくつかの点についても、まだ検討中であるが、これも上記の理由によるものであろう。

P O S I Xを標準として採用する際に問題となる点の一つとして、ライセンスの問題がある。現在P O S I XはIEEEの、また、周知のように、Un i xはATTの商標である。IEEEは、ユーザに制限を加える、あるいはライセンス料を取るという意図はない、と言っているが [3]、Un i xに関してATTがどのような反応を示すかは、まだ不明である。

2. 2 X / O P E N

X / O P E Nは、ヨーロッパのメーカーが中心となって進めているUn i xの標準化作業であるが、現在ではATTをはじめ、米国のメーカーも参加している。P O S I Xと同様にUn i x SystemVをベースに作業を進めているが、P O S I Xよりは若干広い範囲を標準化の対象としている。すなわち、標準化する言語として、最初からC、C o b o l、F o r t r a nの言語仕様を定めており、ユーザーインターフェースやネットワーク機能、グラフィックスなども、標準に含まれる予定である。ただ、対象とする計算機のレベルとしては、P O S I Xと同様にP Cや大型機は対象外としている。現に、X / O P E Nに参加しているメーカーでも、大型機に関しては、独自のOS使用し続けることになっている。

作業方針は、P O S I X、あるいは後述のSSIとは異なり、X / O P E Nで新たな標準を作るという作業は行っていない。つまり、de factoスタンダード、あるいは既に定まっている国際標準を、SystemVに追加して、ビジネス用の計算機システムとして十分な機能を持たせるのが目的である。たとえば、ISAM機能などはRelational Database System INCの製品である“C-ISAM”を使用している。

以上のように、X / O P E Nで行っている作業はP O S I Xとは競合しない。実際に、X / O P E NグループはP O S I Xが国際規格になれば、P O S I Xを採用するとのことである。

2.3 MOSI

MOSIはマイクロプロセッサ向けの標準OSインターフェースで、POSIXと同様IEEEによって提案されている[4]。しかし、特に既存のOSをベースとして採用するのではなく、新たな規格を定めるという方針で作業が進められているようである。

マイクロプロセッサを対象としているため、規格に含まれるシステムの規模にはかなりの差がある。したがって、MOSIでは、各機能毎にモジュールと呼ばれるものを設定し、実際にインプリメントを行う際には、それらのモジュールのうち適当なものを選択する、という形になっている。たとえば、プロセスマネージメントの場合は、モジュール1として、シングルプロセスの場合を、モジュール2として、マルチプロセスの場合を扱っている。

2.4 SAA

IBMは1987年3月17日、IBM Systems Application Architecture(以下SAAと略す)を発表した。技術的な詳細は、まだ不明であるが、パーソナル・コンピュータから大型機まで(System/370, S/3X, PC)のプログラム・インターフェースを標準化しようとする、かなり大規模なものようである。現在分かっているところでは、SNAに基づいた通信サービス、コマンドプロセッサ

などのユーザ・インターフェース、データベース・アクセス、グラフィックなどの機能も含まれている。また、言語に関しては、C, Fortran, Cobol, C SPなど、複数の言語をサポートする。

2.5 その他

アプリケーション・プログラムに対するインターフェースの標準化案としては、以上が代表的なものであるが、これに関連する2, 3の標準化案について簡単に触れておく。

まず、ISO TC97/SC21で検討されているOSCRALであるが、これは、ユーザーインターフェース用のコマンド言語の標準化であり、コマンドプロセッサ自体はアプリケーションプログラムと見なせるので、OSインターフェースには、直接の関係はないと言えよう。実際、POSIXのようにコマンド言語を規格に含めないものも多い。しかし、計算環境を標準化しようとするならば、コマンドのパラメータをアプリケーションプログラムに渡す場合などのように、コマンドプロセッサがプログラムのポータビリティに影響を与える場合もあるので、OSインターフェースの標準化と、コマンド言語の標準化は、密接な連絡をとりつつ進める必要があると考えられる。

表1 標準化案の比較

	期間	視野	new or de facto	具体性
SSI	長	広い	new	具体的でない
POSIX	短	狭い	de facto	かなり具体的
MOSI	短	POSIXよりは広い	de facto	具体的
OSCRAL	長	(コマンド言語)	new	
SAA	?	広い	new?	具体的?
ODP	長	非常に広い	new	具体的でない
X/OPEN	短	POSIXよりは広い	de facto	具体的?
(OSS I)	短?	やや狭い	new	非常に具体的

つぎに、ODP (Open Distributed Processing) であるが、これは分散処理一般に対して基本的なモデルを作ろうという試みであって、非常に広い範囲を対象としている。将来、分散処理が一般的になるであろうことを考えれば、OS インターフェースの標準化でも分散OS について十分な検討を行わなければならないのは、事実である。しかし、現在のところ、ODP と OS インターフェースの標準化とは、直接の関係はないように思われる。

最後に、これは標準化案ではないが、興味深い試みとして、OSS I [5] がある。これは、Modula-2 で書かれたプログラムをポータブルにする試みで、Modula-2 用の OS に依存しない標準インターフェースを定めることが目的である。現在、Lilith, VAX/VMS, IBM PC, マッキントッシュ, Unix 上で OSS I は稼働している。

なお、表 1 にこれら標準案の特長をまとめた。

3. SSI

現在、我が国では情報処理学会 OS インターフェース専門委員会を中心として、SSI (System Software Interface) と呼ばれる OS インターフェースの標準化作業が進んでいる。本章では、SSI の目的、基本構想等について述べる。

SSI は、POSIX 等と同様にアプリケーション・プログラムに対する OS インターフェースを標準化しようとするものである。し

かし、POSIX が Unix マシンを、また、MOSI がマイクロプロセッサを対象としているのに対して、SSI では大型機までも含めた、幅広いスコープの標準化を目指している。

SSI の概念図を図 1 に示す。OS の提供する機能には、OS の内部だけで使用されるものと、アプリケーション・プログラム、ユーティリティ・プログラム等が使用するものの 2 種類に分けることができる。このうち、SSI が対象とするのは、後者だけであって、いわゆる OS の内部コールは規定しない。あくまでも、アプリケーション・プログラムのポータビリティを高めるのが目的であって、OS の内部構造には関心がないからである。

内部コール以外のシステムコールを機能的に分類すると、少なくとも、次のような 2 つの機能レベルに分けることができると考えられる。

- 1) システム・ソフトウェア・アプリケーション・インターフェース (SSAI) : エンド・ユーザーの作成したアプリケーション・プログラムとそれを提供するシステム・ソフトウェアとのあいだのインターフェース
- 2) システム・ソフトウェア・カーネル・インターフェース (SSKI) : ユーティリティ・プログラム間のインターフェース。そのうちのいくつかは、アプリケーション・プログラムも使用することがある。

さて、以上は機能レベルの問題であったが、システムコールのレベルとしては、もう一つ、

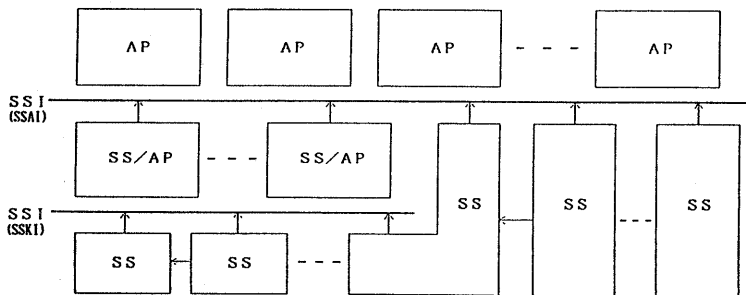
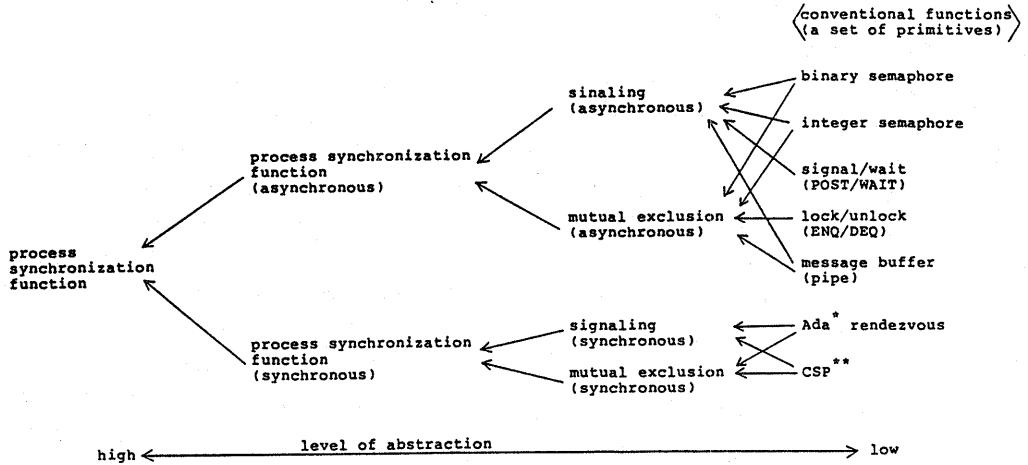


図 1 SSI の基本概念



* Ada is a registered trademark of the U.S. Government.
 ** Communicating Sequential Processes (by C.A.R. Hoare)

図2 抽象化レベル

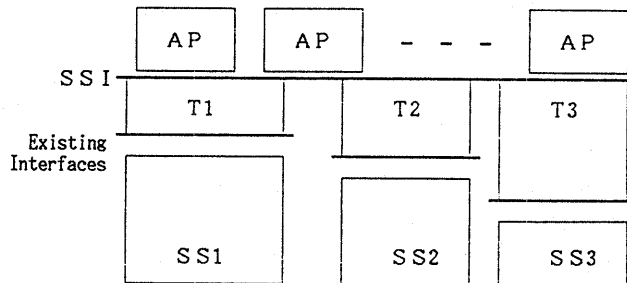


図3 SSIトランスレータ

抽象化レベルがある。基本的には、抽象化レベルを高めれば、その機能は一般的なものとなり、下位のレベルでは複数のプリミティブだったものを、1つのプリミティブに置き換えられる可能性がある。たとえば、プロセス同期プリミティブの例を図2に示す。

もちろん、抽象化レベルを高くすれば全て解決するわけではない。特に、SSKIではパフォーマンスが重要となるので、過度の抽象化を行ってパフォーマンスの低下をまねくことは避けるべきである。

SSIの仕様作成は次のような手順で行われる予定である。

- 1) 各種OSのサービスプリミティブの調査
- 2) サービスプリミティブの分類
- 3) 機能レベルの設定
- 4) 機能レベル間での作業の優先度の決定
- 5) プリミティブ間での作業の優先度の決定
- 6) 統一性のある設計方針の作成
- 7) 各クラスのプリミティブの設計
- 8) 言語バイディングの決定
- 9) SSIインターフェースを既存のシステム・ソフトウェアにマップするトランスレータの作成。

上記の 1) ~ 8) までは、特に説明は必要ないと思うが、9) について、若干の補足を行う。POSIXと異なり、SSIでは de facto スタンダードを採用する予定は現在のところ無いため、標準が固まったとしても、それをサポートするシステムが存在しないため、直ちにSSIに従ったソフトウェアが流通する可能性は少ない。従って、直接SSIのシステムコールをサポートするOSが一般的になるまで、多少の効率低下には目をつぶって、SSIのシステムコールを既存システムへマップするトランスレータが必要になろう(図3)。このようなトランスレータを主要なOSについて作成しておけば、他OSへのプログラムの移植も容易になると思われる。

なお、国内ではSSIの他にTRONプロジェクトも推進されているが、これについては、資料も豊富なので、ここでは割愛することにした。

4. 標準化作業の現状

以上述べてきた標準化案のうち、もっとも作業が進んでいるのはPOSIXである。これは、ISOのNWIとして提案され、各国の投票が行われた。ここでもっとも問題になったのは、トレードマークの問題であり、そのため反対投票をした国もみられた。しかし、Unixをベースにしているため、技術的にはそれほど問題があるとは考えられないので、NWIとなるのはほぼ間違いないと思える。また、X/OPENも、将来はPOSIXを採用すると予想される。

MOSIの作業状態もPOSIXとほぼ同様な状態であるが、こちらの方はまだ実績がほとんどない点が異なっている。MOSIはIECで国際投票にかけられていたが、反対投票はほとんどないので、IECで標準化されるのは確実であろう。

SSIもPOSIXとほぼ同時にISOのNWIとして提案を行ったが、NWIの定義が問題となった。しかし、反対投票を行った国でも、より厳密な定義がなされるならば、作業に加わってもよいという意志を示したと

ころもあった。

5. むすび

以上述べてきたように、POSIXあるいはMOSIなどは、かなり標準化作業が進んでおり、SSIとしても、どのように対応していくかが、重要な課題である。さらに、SSI規格原案を作成する必要があるが、これは、今年度よりWGを設置し、原案の本格的な検討を開始しつつある。

なお、SSIとTRONの関係については、詳しく触れなかったが、SSIの原案作成の有力な一材料として扱うというのが、基本的な方針である。

<参考文献>

- [1] Technical Committee on Operating Systems of the IEEE Computer Society : "IEEE Std.1003.1 Draft American National Standard IEEE Trial-Use Standard Portable Operating Systems for Computer Environments, John Wiley & Sons, 1986
- [2] C information Bulletin dated July 1985
- [3] Response from IEEE concerning the POSIX trademark, ISO/TC97 N1915
- [4] IEEE Trial-Use Standard Specifications for Microprocessor Operating Systems Interfaces, IEEE, 1985
- [5] Blagioni, Hinrichs, Heiser, Muller : "A Portable Operating System Interface and Utility Library", IEEE Software, Nov. 1986
- [6] システムソフトウェアの標準化に関する調査研究——OSインターフェース標準化の検討——報告書, 情報処理学会, 1987