

MIAを実装したUNIX-OLTP環境下でのUNIX標準機能の使用についての一考察

7H-6

小山賢一 瀬尾紳一郎
NTT情報システム本部

1. はじめに

NTTではコンピュータ調達仕様としてMIA (Multivendor Integration Architecture)仕様を規定している。MIAは汎用機、UNIX機等の全てのオンライン環境を適用対象として、全ての環境間でのAPの移植を保証するために、APによる言語(C, COBOL, SQL, STD L)記述の使用について規定している。一方、UNIXには、UNIX標準として規定されベンダ環境間の移植性が保証されている機能(システムコール、C関数)が存在している。UNIX-OLTP上での動作を前提としたオンライン業務の実現においては、AP開発の効率化等の点から本機能を使用したいというニーズがある。しかし、システムコール、C関数は、OLTP環境下での使用可否についてベンダ毎の差異がありAPの移植性が損なわれる、等の問題が懸念される。本稿ではMIAを実装したUNIX-OLTP環境においてTPモニタ配下のAPからの使用が必要となるUNIX環境対応機能等について、一考察を示す。

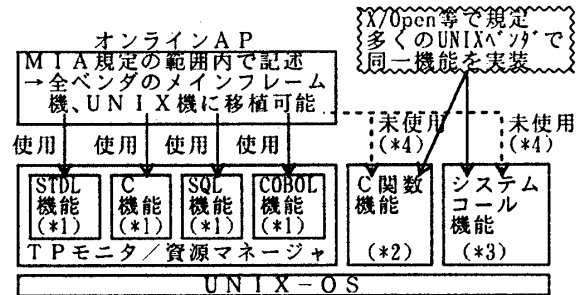
2. UNIXを導入するホスト環境の特徴

(1) MIAの特徴

MIAはメインフレーム機、UNIX機等の全てのオンライン環境をAP移植対象として、APIの規定を行っている。MIA規定の範囲内で記述されたTPモニタ配下のAP(以下オンラインAPと記す)は、全ベンダのオンライン環境に移植が可能である。(図1)

(2) UNIXを導入するホスト環境の特徴

NTTでのMIA導入によるオンラインシステムはホスト環境としてメインフレーム機を使用するケースと、UNIX機を使用するケースの2つに大別される。UNIX機をホスト環境としてオンラインシステムを構築する場合は、将来の機種更改の対象(開発済みAPの移植先環境)もUNIX機であるということ前提としている場合が多い。この場合、MIAで規定している機能/資源(SQLデータベース、Cストリームファイル、COBOLファイル)の他に、UNIX標準としてX/Open等で規定され、多くのベンダのUNIX機で実装されている豊富な関数/システムコールを使用して、オンライン業務APの開発の効率化、性能向上等を図り、同時にベンダ間の移植性の確保を図りたいというニーズがある。(図1)



- (*)1 MIA規定の各言語機能
- (*)2 MIAでは48個の関数が使用不可
- (*)3 MIAでは未規定(使用不可)
- (*)4 使用可ならAPの開発の効率化等が可

図1 オンラインAPの使用機能

3. オンラインAPからの使用が必要となるUNIXホスト環境対応機能

MIAを実装したUNIXホスト環境を用いて、NTTの代表的なオンライン業務を実現する上で必要となった機能(MIA規定機能及び資源の他に必要となった機能)の分類を以下に示す。

- (分類A)共有メモリ/セマフォをTPモニタ配下のトランザクション資源として組み込んで使用するための機能
- (分類B)UNIX-OS直下のプロセスとオンラインAPとの間の通信を行うための機能
- (分類C)C言語によるオンライン業務記述の効率化のために必要となったC関数機能(分類Aの実現のために補完的に必要となった関数も含む)

分類Bの機能は、各ベンダにおいてUNIX-OS直下のプロセスとしての動作(あるいは該プロセスからの使用)を前提に提供されている機能(各種通信に関する機能等)をオンラインAPから使用可能とするために必要となる機能である。

上記分類における必要機能詳細を表1に示す。

表1 オンラインAPからの使用が必要となるUNIX環境対応機能

分類	オンラインAPからの使用が必要となる機能
分類A	①共有メモリ使用のためのシステムコール (shmat, shmctl, shmdt, shmget) ②セマフォ使用のためのシステムコール (semget, semctl, semop) ③トランザクション制御機能(トランザクション開始/終了, 例外提起, トランザクション識別子取得, トランザクションウェイト/ウェイト解除, トランザクション終了出口)
分類B	①UNIX規定の「メッセージ」使用のためのシステムコール (msgctl, msgget, msgrev, msgsnd)
分類C	①メモリ操作/アクセス用関数(calloc, memcpy等 6種) ②ファイル操作/アクセス用関数(fopen, fclose, fread, fwrite等 10数種) ③文字列操作/変換/分類用関数(strcat, atoi, isalnum等 40数種) ④その他(qsort, sleep等 数種)

On use of UNIX standard functions for UNIX-OLTP environment on MIA
Ken-ichi KOYAMA, Shin-ichiro SEO
NTT Information Systems Headquarters

表2 標準化状況及びベンダ実装状況

分類	必要機能	標準化の状況 (X/Openでの規定)	ベンダ実装状況 *1	
			A社	B社
システム コール	共有メモリ	△	○	○
	セマフォ	*2	○	○
	メッセージ		○	○
T r 制御	Tr開始宣言機能	Txインタフェースで規定	○ *3	○ *3
	Tr終了宣言機能	Txインタフェースで規定	○ *3	○ *3
	例外提起機能	Txインタフェースで規定	○ *3	○ *3
	Tr識別子取得機能	未規定	△ *4	△ *4
	Trウェイト/ウェイト解除機能	未規定	△ *4	△ *4
	Tr終了出口機能	未規定	△ *4	△ *4
C 関数	メモリ操作/アクセス用	△	○	○
	ファイル操作/アクセス用		○	○
	文字列操作/変換/分類用	*2	○	○
	その他 (sleep 等)		○	sleep使用不可

*1 TPモニタ上のAPからの使用可否は未規定のみ。TPモニタ上のAPからの使用可否は未規定
 *2 UNIX-O/S直下を前提とした規定のみ。TPモニタ上のAPからの使用可否は未規定
 *3 MIA規定のトランザクション制御機能が使用可 (Txインタフェースは未サポート)
 *4 機能はあるがAPIはベンダ固有

分類Aの機能の実現のためには、共有メモリ/セマフォ使用のために直接的に必要なシステムコール機能の他に、これらの資源をトランザクション資源として組み込んで使用するためのトランザクション制御関連のAPI機能が必要であった。

4. 標準化状況及び実装状況

表1に示した各機能に対する標準化状況及びベンダ実装状況を表2に示す。

(1)システムコール、関数のオンラインAPからの使用可否

本件に関する標準化の状況としては、UNIX直下から使用されるシステムコール、関数についてはX/Openで広い範囲で規定がなされているものの、オンラインAPからの使用の点に関しては明確な規定がない。このような状況にも関わらず、UNIXベンダ各社の実装状況は以下の通りであった。

- ①システムコールの使用は、特に問題なく可。
- ②関数の使用についてはsleep関数を除き、今回必要となった関数が全てオンラインAPからの使用が可。

今回抽出したシステムコール、関数はNTTの代表的なオンライン業務を実現するために必要となった機能である。このことは、オンラインAPからの使用が必要となるシステムコール及び関数は、準拠すべき標準が規定されれば、各UNIXベンダともに比較的容易に実装可能であることを示していると思われる。

(2)トランザクション制御用APIからの使用可否

本件に関する標準化の状況としてはX/Open規定のTxインタフェースがある。しかし、Txインタフェースの規定には、今回オンラインAPからの使用が必要となったAPI機能のうち、以下の機能についての規定がなかった。

- ・トランザクション識別子取得機能
- ・トランザクションウェイト/ウェイト解除機能
- ・トランザクション出口機能

上記各機能についてのベンダの実装状況は、機能はあるが、標準が無いため、APIはベンダ固有という状況であった。

このことは準拠すべき標準が規定されれば実装可能な機能であることを示していると思われる。

5. 現状の対処

(1)システムコール、関数

下記により、オンラインAPの移植性の確保を図ることとした。

- ①システムコールについては、表2に示したシステムコールを必須機能としてベンダ各社にサポートしてもらう。
- ②関数については、表2に示した関数のうちベンダ間で共通使用可であるもののみを使用してAPを開発する。

(2)トランザクション制御用API

標準がなく、ベンダ固有となるAPI3種類(4.(2)参照)について、NTT独自のインタフェースを規定し、本インタフェース規定に従った機能をベンダ各社にサポートしてもらうことで、オンラインAPの移植性の確保を図ることとした。

6. おわりに

UNIXを前提としたオンラインホスト環境の実現に必要なシステムコール、関数、トランザクション制御用APIについて、標準化の状況、ベンダの実装状況、現状の対処について述べた。X/OpenでTxインタフェースが規定されたように、今後はOS上のプログラムレベルでなく、TPモニタ配下のオンラインAPレベルでの移植性保証についての標準化が重要となってくる。今後の標準化の進展が望まれる。

参考文献

- (1)Multivendor Integration Architecture, Version 1.2 第1編概説 テクニカル リクワイアメント Tr550001-1, NTT, Jan., 1993
- (2)Multivendor Integration Architecture, Version 1.2, 第2編アプリケーション・プログラムインタフェース仕様 Tr550001-1, NTT, Jan., 1993
- (3)X/Open Preliminary Specification(1993) Distributed Transaction Processing: The TxRPC Specification