

OSI 7層ボードのアプリケーション・プログラミング・インタフェース

6V-1

加藤 聡彦 井戸上 彰 鈴木 健二

国際電信電話(株) 研究所

1. はじめに

筆者らは、応用層までのプロトコルをサポートするOSI 7層ボードの開発を進めている^[1]。これまでに、要求条件に応じて選択できるように、32ビットCPUを搭載し上位・下位部分をモジュール化したボードと、16ビットCPUを使用しROMの切替え機能を実装したボードの2種類のハードウェアを開発している^[1]。2種類のボード上では同様なソフトウェアが実行される。OSI 7層ボードを搭載するホスト計算機上において、業務に応じたアプリケーション・プログラムを開発するためには、ボードのOSI通信機能を容易に利用可能とするアプリケーション・プログラミング・インタフェース(API)を提供する必要がある。本稿では、筆者らが開発したOSI 7層ボードのAPIの概要について述べる。

2. 設計方針

OSI 7層ボードのAPIを設計するにあたり、次のような方針を立てた。

- ① 7層ボードが提供する層のサービス・プリミティブに対して、構造体、ポインタ、共用体などを用いて構造化されたデータ型を採用する。
- ② プリミティブの型を持つデータをボードに送受信する関数を準備し、その関数をホスト計算機の機種や7層ボードのハードウェア種別に依存しないようにする。
- ③ 複数のアプリケーション・プロセスが同時に7層ボードを使用可能とする。
- ④ APIとして基本的な機能を提供し、その上に、ユーザが独自の設計方針に基づいて固有のAPIを構築できるようにする。
- ⑤ 各層にインタフェース可能とする。

3. ホスト・ボード間インタフェースの実現方式

OSI 7層ボードにおけるホストとボード間のインタフェースは、図1に示すように実現される。

(1) アプリケーション・プログラムの構成

ホスト上には、OSI 7層ボードのためのデバイス・ドライバが用意され、複数のアプリケーション・プロセス(AP)が同時に起動される。各APには、ボードとのインタフェース機能を実現するAPIライブラリが組み込まれる。APIライブラリは、構造化されたプリミティブ・データを、デバイス・ド

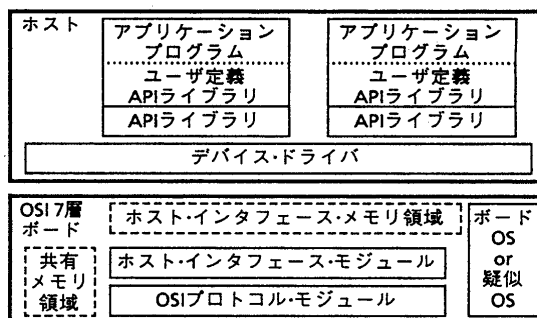


図1 ホスト・インタフェースのためのプログラム構成

ライバの機能を使用し、ボード上のホスト・インタフェース・メモリ領域に読み書きする。さらに本APIの上に、ユーザ定義のAPIライブラリを用意することもできる。

(2) デバイス・ドライバの機能

デバイス・ドライバは、ホスト計算機のオペレーティング・システム(OS)の機能を用いて、OSI 7層ボードへの読み書き(read/write)などの機能を提供する。複数のAPが存在する場合に、接続の着信および接続ごとに受信されたデータに対応するAPに割り当てる処理を実現するために、デバイス・ドライバは、接続番号とプロセスの対応を保持する。

(3) ホスト・インタフェース・モジュールの機能

OSI 7層ボード上には、ホスト・インタフェース・メモリ領域にアクセスするホスト・インタフェース・モジュール(HIF)が用意される。APIライブラリとHIFとの間でやり取りされるデータの形式は、ホスト・ボード間のデータ転送に最適化されたものを用いる。HIFは、ホスト・インタフェース・メモリ領域にかかれたデータと、ボード内部の処理のために共有メモリ領域上に展開された構造化されたプリミティブの間の変換を行う。その際に、ホストCPUとボードCPUのデータ表現が異なる場合のデータ表現変換も行う^[2]。このためHIFは、プリミティブ中のポインタの位置およびデータ表現変換を行うパラメータの種類と位置を示すテーブルを、各層のすべてのプリミティブに対して持つ。

APPLICATION PROGRAMMING INTERFACE OF OSI 7 LAYER BOARD

Toshihiko KATO, Akira IDOUE and Kenji SUZUKI
KDD R & D Laboratories

4. APIの概要

4.1 プリミティブのデータ型

図2にOSI7層ボードを使用するためのプリミティブの例として、P-CONNECT要求のデータ型の一部を示す。この図が示すように、プレゼン

```

struct {
    prim_head_t    PrimHead;
    common_param_t ComParam; /* 共通パラメータ */
    user_data_list_t UserDat;
} p_connect_req_t;

typedef struct {
    p_cntxt_list_t PCDefList; ...
} common_param_t;
typedef struct {
    long Num; p_cntxt_t PC[PC_MAX];
} p_cntxt_list_t;
typedef struct {
    long PCId; obj_id_t AbsSyn; } p_cntxt_t;
typedef struct {
    long Num, ObjIdComp[OID_MAX]; } obj_id_t;

typedef struct {
    long Len; pointer_t(user_data_t) UDList;
} user_data_list_t;
typedef struct {
    long PCId; /* Pコンテキスト識別子 */
    long Encodings; /* 3種類の選択肢 */
    long Len;
    pointer_t(char) Data; /* データへのポインタ */
} user_data_t;
#define pointer_t(TYPE) struct { \
    TYPE *Ptr; char Pad[4]; }

```

図2 P-CONNECT要求のデータ型(一部)

テーション・コンテキスト(PC)定義リストは、PCの配列の領域と配列の有効数を示す要素からなる構造体で表される。PCの型p_cntxt_tは整数型のPC識別子と、抽象構文を示すオブジェクト識別子から構成される。オブジェクト識別子は、要素となる整数の配列とその有効数からなる構造体である。また、ユーザデータリストやユーザデータの型のように、可変長の要素を持つ場合は、ポインタを示すpointer_tマクロにより要素の型の配列を指し示し、その前に配列の個数を示す変数を設ける形式を用いている。pointer_tマクロは、OSI7層ボードで6バイト長のポインタを用いる場合があるため、ポインタ変数自身に8バイトを割り当てる。

この例のように、プリミティブのデータ型は、APIライブラリとHIFの処理を容易にするために、要素が4バイトアライメントとなるように定義し、ビットフィールドや双方向リストは使用しない。

4.2 プリミティブを送受信するための関数

プリミティブの型を持つデータを、OSI7層ボードとの間で送受信するために、送受信用の関数(osi_snd/osi_rcv)を提供する。この関数は、デバイス記述子(ファイルディスクリプタ)と、プリミティブバッファへのポインタを引数に持つ。

サイズの大きなユーザデータを持つプリミティブを送受信する場合には、APIライブラリとHIFの

間で、プリミティブを複数回に分割してデータ転送する場合がある。アプリケーション・プログラムからのosi_snd/osi_rcvの発行回数と、OSI7層ボードへのread/writeの回数に対応しない場合に、read/writeの完了待ちでAPIライブラリがブロックしないように、以下の方法を用いる。

- ① osi_snd/osi_rcvの戻り値として、成功/失敗のほかに、不完全入出力を設ける。
- ② アプリケーション・プログラムからのosi_snd/osi_rcvの発行に対して、分割したデータ転送の場合も、APIライブラリは1回のread/writeのみを発行する。その結果、プリミティブ全体を読み書きできなかった場合は、不完全入出力の戻り値を返す。
- ③ アプリケーション・プログラムは、戻り値が不完全入出力である場合は、そのプリミティブに対して、成功または失敗が返されるまで、osi_snd/osi_rcvの発行を繰り返す。

この方法では、APIライブラリがデバイス・ドライバの読み書きの機能のみを使用するため、APIライブラリをホスト計算機のOSの機種に依存しない構成とすることが可能である。しかし、1つのプリミティブを送受信するためにアプリケーション・プログラムがosi_snd/osi_rcvを複数回発行する場合がある。これは、必要に応じて、ユーザ定義APIライブラリにより隠蔽することができる。

4.3 複数アプリケーション・プロセスの起動

複数のアプリケーション・プロセス(AP)からのread/writeの調停は、デバイス・ドライバが行うため、各APは他とは独立に起動される。APは起動されると、OSI7層ボードのデバイスをオープンし、デバイス記述子を得る。また、コネクションは、プリミティブのヘッダに含まれるアプリケーション識別子とコネクション識別子により区別する。アプリケーション識別子は、各APに固有の値が割り当てられる。コネクション識別子については、コネクション確立時にはAPがその中で一意に奇数の値を割り当て、コネクション着信時にはボードが偶数の値を付加する。

5. おわりに

本稿では、OSI7層ボードのAPIについて述べた。本APIでは、構造体やポインタなどにより構造化されたプリミティブのデータ型と、機種に依存しない送受信関数を使用可能とすることにより、アプリケーション・プログラムの開発を容易にしている。最後に日頃ご指導いただくKDD研究所小野所長、浦野次長に感謝する。

参考文献

- [1]:加藤, 井戸上, 鈴木, "パソコン用OSI7層ボード," 1992年信学秋季全大, B-412, Sept. 1992.
- [2]:加藤, 井戸上, 鈴木, "汎用OSI7層ボードにおけるプロトコル・プログラムの構成法," 第44回情処全大, 4L-11, Mar. 1992.