

# ネットワークプログラミング

Jacques Garrigue, 2005 年 2 月 1 日

## 6 遠隔手続き呼び出し

TCP のようなプロトコルは通信の完全性を保証しているが、分散計算などで行う通信とは概念が違う。分散計算ではメッセージ通信が基本だが、TCP はストリーム通信を提供している。ストリーム通信の上でメッセージを送るのは当然可能だが、効率が悪く、書きにくかったりする。UDP の方が自然かも知れないが、UDP はメッセージの大きさに制限があり、伝達も保証していない。

Remote Procedure Call (RPC) はこういう用途のために作られたプロトコルである。通信が関数呼び出しのように書けるので、プログラムの中で使いやすい。基本的な概念はメッセージ一個ではなく、要求と応答の対である。サーバに要求を送り、応答が来たら処理を続けるという、自然な流れになる。効率も毎回 TCP の接続を始めるよりいい。

RPC に必要な機能は以下のとおりである。

1. 呼びたいプログラムと手続きを調べる機能 (mapper)
2. 引数や結果をネットワークで送れる形に変換する機能 (serialization)
3. 要求と応答を対応させる機能
4. 通信の完全性を保証する機能

有名な RPC の実装として以下の三つがある

**SunRPC** Sun Microsystems が開発した UNIX と C のための実装。NFS などで使われている。ほとんどの UNIX や Linux で提供されている。

**CORBA** 分散アプリケーション向けの RPC 機構。C や C++ を使う。

**Java RMI** プログラミング言語 Java のための RPC (オブジェクト指向言語なので Remote Method Invocation という)

三つとも、使いやすさのために、スタブコンパイラという機能が含まれる。呼び出したい手続きのプロトタイプ (型情報) を与えると、引数や結果の変換を行うコードや、通信するためのコードが自動的に生成される。これで (2) と (3) が実装される。

(1) と (4) は実装によるが、SunRPC の場合では、

- プログラムを調べる機能は portmapper という RPC サービスで提供される。各プログラムの ID を送ると、portmapper がそのポート番号を教えてくれる。portmapper 自身は決まったポート (111 番) を割り当てられる。
- プログラムの中の手続きも別の ID で区別される。自分のポートにメッセージが来ると、その ID を見て手続きを選ぶ。
- 通信プロトコルとして UDP を利用するが、長過ぎるメッセージや紛失されるメッセージは RPC レベルで処理される。

## 実習

以下に簡単な RPC を利用するプログラムを見せる。このプログラムはサーバに日付を聞きに行く。

前回の積分を分散で計算するプログラムも RPC で実装すれば書きやすくなるはずである。

date.x

```
/*
 * 遠隔手続きの宣言
 *   strdate_1() - 日時を書いた文字列を返す
 */

program DATE_PROG {
    version DATE_VERS {
        string STRDATE(void) = 1; /* procedure number = 1      */
    } = 1; /* version number = 1 */
} = 0x41320020; /* program number = 0x41320020 */
```

“rpcgen -C date.x” で date.h, date\_svc.c, date\_clnt.c という 3 つのファイルが生成される。

### svdate.c

```
#include <sys/time.h>
#include <rpc/rpc.h>
#include "date.h"

char ** strdate_1_svc(void *arg, struct svc_req *req) {
    static char *result;          /* Must be static. */
    struct timeval theTime;       /* To hold binary time. */

    gettimeofday(&theTime, NULL);
    result = ctime((const time_t *)&theTime.tv_sec);
    return(&result);
}
```

### rdate.c

```
#include <stdio.h>
#include <rpc/rpc.h>
#include "date.h"

int main(int argc, char **argv)
{
    char *server;                /* 遠隔サーバ */
    CLIENT *client;              /* クライアント情報 */
    char **result;               /* 遠隔呼び出しの結果 */

    /* サーバ名をコマンドラインから取る */
    if (argc != 2) {
        fprintf(stderr, "usage: rdate hostname\n");
        exit(1);
    }
    server = argv[1];

    /* UDP 型のクライアントを作る */
    client = clnt_create(server, DATE_PROG, DATE_VERS, "udp");
    if (client == NULL) {
        clnt_pcreateerror(server); /* 接続失敗 */
        exit(2);
    }
}
```

```
    }

    /* strdate を呼ぶ */
    result = strdate_1(NULL, client);
    if (result == NULL) {
        clnt_perror(client, server); /* 呼び出し失敗 */
        exit(2);
    }
    printf("Time on host %s = %s\n", server, *result);

    clnt_destroy(client);          /* クライアント情報を破棄 */
    exit(0);
}
```

コンパイル方法 以下のコマンドでコンパイルする .

```
rpcgen -C date.x
gcc -DDEBUG svdate.c date_svc.c -o svdate
gcc rdate.c date_clnt.c -o rdate
```

Solaris では、さらに `-lrpcsvc -lnsl` を付けなければならない .

`svdate` を立ち上げておいて、`rdate` サーバ名 で日時を調べる .

ちなみに、`portmapper` が動いていなければ、サーバもクライアントも動かない . `ps -aux` で `portmap` が `rpcbind` が動いていることを確かめればいい .