

遠隔手続き呼び出し

RPC(Remote Procedure Call)

環境設定

- Linux端末上にて

```
kit@pandrabox:~$ sudo apt-get install rpcbind
```

- そのあとソフトウェアがインストールされます。途中で

続行しますか? [Y/n]

- と聞かれますがYを入力してください。
- 必要なファイルをダウンロードし解凍する。(rpcgen.gz)

```
unzip rpcgen.gz
```

- いくつかのフォルダーが出来上がるので以下のコマンドで移動する。

```
kit@pandrabox:~/Documents$ ls
rpcgen.gz  rpcgen/
kit@pandrabox:~/Documents$ ls -l
合計 12K
-rw-r--r-- 1 kit kit 7.4K  5月 15 06:55 rpcgen.gz
drwxrwxr-x 5 kit kit 4.0K  5月 14 21:53 rpcgen/
kit@pandrabox:~/Documents$ cd rpcgen/
kit@pandrabox:~/Documents/rpcgen$ cd data2/
kit@pandrabox:~/Documents/rpcgen/data2$
```

- フォルダーの中身を確認する (rpcgen/data2)

```
kit@pandrabox:~/Documents/rpcgen/data2$ ls -l
合計 12K
-rw-rw-r-- 1 kit kit 282  5月 14 16:01 strcap.x
-rw-rw-r-- 1 kit kit 379  5月 14 21:21 sv_upchar.c
-rw-rw-r-- 1 kit kit 950  5月 14 21:00 client_string.c
```

- rpcgenコマンドを使って、定義ファイルを作成する

```
kit@pandrabox:~/Documents/rpcgen/data2$ rpcgen -C strcap.x
kit@pandrabox:~/Documents/rpcgen/data2$ ls -l
合計 24K
-rw-rw-r-- 1 kit kit 282 5月 14 16:01 strcap.x
-rw-rw-r-- 1 kit kit 692 5月 17 08:22 strcap.h
-rw-rw-r-- 1 kit kit 379 5月 14 21:21 sv_upchar.c
-rw-rw-r-- 1 kit kit 2.2K 5月 17 08:22 strcap_svc.c
-rw-rw-r-- 1 kit kit 556 5月 17 08:22 strcap_clnt.c
-rw-rw-r-- 1 kit kit 950 5月 14 21:00 client_string.c
```

- サーバのコンパイル

```
kit@pandrabox:~/Documents/rpcgen/data2$ gcc -o sv_upchar sv_upchar.c strcap_svc.c
kit@pandrabox:~/Documents/rpcgen/data2$
```

- クライアントのコンパイル

```
kit@pandrabox:~/Documents/rpcgen/data2$ gcc -o client_string client_string.c strcap_clnt.c
kit@pandrabox:~/Documents/rpcgen/data2$
```

- 実行画面

- サーバ実行前に "sudo rpcbind"を実行してください。

The screenshot shows two terminal windows. The background window shows the execution of the client program `./client_string` with various arguments, including `hostname`, `^C`, `127.0.0.1 distributed`, `127.0.0.1 AAAAA`, `127.0.0.1 sadfas asa`, and `127.0.0.1 sadfasdfdf`. The foreground window shows the execution of the server program `./sv_upchar`.

```
kit@pandrabox:~/Documents/rpcgen/data2$ ./client_string
usage: rdate hostname string
kit@pandrabox:~/Documents/rpcgen/data2$ ./client_string ^C
kit@pandrabox:~/Documents/rpcgen/data2$ ./client_string 127.0.0.1 distributed
arg = distributed
Run on host 127.0.0.1 = DISTRIBUTED
kit@pandrabox:~/Documents/rpcgen/data2$ ./client_string 127.0.0.1 AAAAA
arg = AAAAA
Run on host 127.0.0.1 = AAAAA
kit@pandrabox:~/Documents/rpcgen/data2$ ./client_string 127.0.0.1 sadfas asa
usage: rdate hostname string
kit@pandrabox:~/Documents/rpcgen/data2$ ./client_string 127.0.0.1 sadfasdfdf
arg = sadfasdfdf
Run on host 127.0.0.1 = SADFASDFDFFF
kit@pandrabox:~/Documents/rpcgen/data2$
```

コードの説明

- 定義コード(strcap.x)

```
/*
 * 遠隔手続きの宣言
```

```

* strcap_1() - 小文字列から大文字列を返す
*/
program DATE_PROG {
    version DATE_VERS {
        string STRDATE(string) = 2; /* procedure number = 2 */
    } = 1; /* version number = 1 */
} = 0x41320021; /* program number = 0x41320021 */

```

- クライアントコード (client_string.c)

```

#include <stdio.h>
#include <rpc/rpc.h>
#include "strcap.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 != 3) {
        fprintf(stderr, "usage: rdate hostname string\n");
        exit(1);
    }
    server = argv[1];
    *result = argv[2];
    printf("arg = %s\n", *result);
    /* 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(result, client);
    if (result == NULL) {
        clnt_perror(client, server); /* 呼び出し失敗 */
        exit(2);
    }
    printf("Run on host %s = %s\n", server, *result);
    clnt_destroy(client); /* クライアント情報を破棄 */
    exit(0);
}

```

- サーバコード(sv_upchar.c)

```

#include <string.h>
#include <ctype.h>
#include <rpc/rpc.h>

```

```
#include "strcap.h"

char ** strdate_1_svc(char **arg, struct svc_req *req) {
    static char *result;
    int numrcv;
    int i;

    numrcv = strlen(arg[0]);

    for (i=0; i< numrcv; i++){ // bufの中の小文字を大文字に変換
        if(isalpha(arg[0][i])) arg[0][i] = toupper(arg[0][i]);
    }
    return(&arg[0]);
}
```