科学分野向けの基本ライブラリ "SLLIB-1.4.0" の開発・公開

○山内 千里 (NAOJ, ISAS)
chisato.yamauchi@nao.ac.jp
松崎 恵一 (ISAS)
山本 幸生 (ISAS)

http://www.ir.isas.jaxa.jp/~cyamauch/sli/

SLLIB は,「ストリーム」「文字列」「多次元配列」をスクリプト言語のように扱う事が可能 (拡張正規表現や配列演算が利用可能) な,『**言語環境**』を提供する C プログラマ向けのライブラリである．科学者にとって重要な「正しさ」「透明さ」を重視しており，天文データ処理をはじめ様々な分野で利用できる．前回の年会で発表した SFITSIO(FITS I/O ライブラリ) のコア・ライブラリでもある．

近年の天文データ処理ではその複雑さのため，Python などのスクリプト言語が多用される．その一方で，スクリプト言語の速すぎる進化等の問題から，C++を主な開発言語とする場合もみられるが，成功しない事も多い．その理由として，プログラマが C++ の代表的ライブラリの設計思想 (抽象化や汎用化) にひきづられ，スクリプト言語的な実用性を重視してこなかった事が挙げられる．

SLLIB は，**実用性**を重視した API 群によってスクリプト言語に近い手軽さを提供し，プログラミング言語を切り替える事なく最高のパフォーマンスを狙える開発環境を実現する．2013 年 2 月にリリースされた Version 1.4.0 では，SIMD 命令による高速化に加え，IDL 風の記法 (例: `"0:99,*"`) で配列を編集・演算可能になり，よりスクリプト言語に近い感覚でコーディングが可能になった．

概要

SLLIB
Google 検索

- **SLLIB とは?**
  「ストリーム」「文字列」「多次元配列」の『3点セット』について，スクリプト言語のような**実用性重視**の API を C/C++環境で使えるようにし，科学研究に適した『**言語環境**』を提供する基本ライブラリ．(Python では,「Python 本体 + numpy の基本機能」に相当)
- **開発の動機**
  - 「あかり」「旧 L1TSD(ASTRO-H 向け)」のデータ処理でみえてきた C 言語の限界 (開発者への過度の負担，プロジェクトの難航)
  - 速度やメモリ管理の点で C++が良いのはわかっているが，日常的コーディングに必要なものが揃っておらず，C++は不便である
- **設計思想**
  - 天文業界で平均的スキルを持つ研究者が主なターゲット．プログラマはクラスを使って**手続き型のプログラミング**を行なう事を想定
  - C++特有の作法を使わず，C 言語の良さを継承した API
    ✕ `cout << "foo" << endl;`　○ `printf("foo");`
  - 工場メソッドは一切なし．プログラマが new や malloc() しない限り，メモリリークは起こらない構造
  - 大規模データの扱いを前提とした最適化．スレッドセーフな設計
- **正しさ，透明性へのこだわり**
  - C1(分岐網羅)，多数のコードによる「make test」等の試験
  - 科学に直結する部分 (統計用関数) はコンパイルせず，ヘッダファイルにコードを記述．インストール後も確認・流用が容易
- **サポート・ライセンス**
  C-SODA/ISAS の公式サポート．MIT ライセンス or GPL

最新安定版 Version 1.4.0 の現状

- **ストリーム**
  - ネットワーク経由 (http, ftp) の圧縮ファイル (gzip, bzip2) に対応
  - シーク可能/不可能それぞれでの最適化を支援する API を提供
- **文字列**
  文字列配列, 連想配列, 拡張正規表現, シェル風のマッチ，デリミタで分割，数式の括弧の解析, LIBCと同等の関数 (printf(), atof(), etc.) 等
- **多次元配列**
  演算子等による配列とスカラー・配列との演算，次元数・要素数の動的な変更，**IDL/IRAF 風の表現 (例: `"0:99,*"`) による配列の編集・演算**，**2D・3D データのポインタ配列の自動生成**，y方向・z方向への高速スキャンのためのtranspose機能，配列に対する数学関数，統計用関数 (moment, median, etc.)，C99準拠の複素関数 等
- **速度性能**
  - SIMD 命令 (SSE2) を使った，メモリ初期化，メモリコピー，上下左右反転，バイトオーダ変換などの高速処理用コードを搭載
  - inline・マクロ等によるループ中の if 文や関数ポインタの除去，本物 median の算出・transpose の高速化等の地道な最適化

利用・検討しているプロジェクト

- JAXA 宇宙科学研究所 (C-SODA，赤外，月惑星 等)
  L1TSD TLM2FITS (前回の年会 W58b),「あかつき」姿勢データ作成 (現在運用中)，AKARI-DAS 開発プロジェクト，他
- すばる HSC チームで試用中，東広島天文台 HONIR 他で検討中.

プログラム例: ストリーム

- **URL をいきなりオープン**

```
digeststreamio f_in;
f_in.open("r", "http://foo.ac.jp/file.txt.gz");
```

「digeststreamio」は様々な圧縮・非圧縮ストリームに対応する最強のクラス

- **printf() 形式の引数が，様々なところで使える**

```
for ( i=0 ; i < N ; i++ ) {
    f_in.openf("r", "ftp://foo.ac.jp/file_%d.txt.gz", i);
      :
    f_in.close();
}
```

プログラム例: 文字列

- **POSIX 拡張正規表現 (「sed -e …」と同様に後方参照も OK)**

```
stdstreamio sio;
tstring my_url = "http://darts.isas.jaxa.jp/foo/";
my_url.regreplace("([a-z]+://)([^/]+)(.*)", "\\2", false);
sio.printf("hostname = %s\n", my_url.cstr());
```

実行結果: `hostname = darts.isas.jaxa.jp`

- **高度な文字列分割 (括弧やクォーテーションを認識)**

```
const char *line = "winnt( )  'program files'";
tarray_tstring my_arr;
my_arr.split(line, " ", false, "'()", '\\', false);
my_arr.dprint();
```

実行結果: `tarray_tstring[] = {"winnt( )", "'program files'"}`

プログラム例: 多次元配列

- **基本**

```
mdarray_float arr0(false);          /* オブジェクト作成 */
arr0.resize_2d(8,4);                /* 配列の大きさの設定 */
arr0 = 500.0;                       /* 全要素への値の代入 */
arr0 *= 2;                          /* スカラーとの演算 */
arr0.dprint();                      /* 内容表示 */
```

- **高速な要素の代入 (内部で自動生成されるポインタ配列を利用)**

```
float *const *arr0_ptr = arr0.array_ptr_2d(true);
size_t i, j;
for ( i=0 ; i < arr0.row_length() ; i++ ) {        /* Y */
    for ( j=0 ; j < arr0.col_length() ; j++ ) {    /* X */
        arr0_ptr[i][j] = 100 + 10*i + j;
    }
}
```

- **切り出してコピー (型は double へ変換)，指定範囲へペースト**

```
mdarray_double arr1(false);
arr1 = arr0.sectionf("1:4, *");          /* 0-indexed */
arr0.pastef(arr1, "*, 2:3");             /* x,y の順で指定 */
```

※ 型が同一の場合の「=」では，shallow copy となる.

- **統計値を得る**

```
/* get mean, variance, skewness, kurtosis */
mdarray_double moment = md_moment(arr0,false, NULL,NULL);
```

- **y 方向で「本物 median」を得る**

```
/* y 方向で median が計算され，1 次元データが返される */
arr1 = md_median_y(arr0);
```

各統計用関数で x,y,z 方向それぞれのものを提供．詳細は Web ページにて