

実習の目的

- 基本的な考え方を、肌身を持って実感すること（頭でなくて、体で実感して欲しい）
- ①異なるプログラミングスタイルがあること(どう異なるか)、②並列の制約(⇒並列化への制約)を実感して欲しい

実習課題[1] 単純なデータ並列を、OpenMP を使って実際に体験する

OpenMP の環境で、「区分求積法」で 1/4 円の定積分の計算をせよ。

環境は、venus.is.sci.toho-u.ac.jp を用いよ。

実験のため、分割数を非常に多く(10万、100万など)して、計算量を増やせ。念のため、分割数を増やすと結果がどれだけ円周率に近づくか、確認しておくこと。

このとき、計算は double 型(倍精度浮動小数)を用いる。Double 型は仮数部が 52 ビットなので、10進換算で15～6桁の有効数字が得られる。プリントするときに、printf(“%.18f\n”, S) などとして、十分な桁数を表示すること。

分割数を固定し、同時プロセス数を変えて、所要時間を測定してみよ。「アムダールの法則」がどう見えるか？

区分求積法: 別紙(HP <http://naop.jp/text/3/seki10.html>)参照

これでは長方形を並べて計算しているが、1つ1つを台形にする(台形法)や、2つ併せて放物線にする(シンプソン法)などの工夫がされている。ネット等で探してみよ。

時間の測定:

時間の測定は、C のライブラリの中にある gettimeofday() 関数を使うとできる。これはウォールクロック(壁掛け時計の時間)を測定する。

```
#include <time.h>
#include <sys/time.h>

int main () {
    int time;
    struct timeval tv_start, tv_end;

    gettimeofday(&tv_start, NULL); /* 開始時刻を記憶する */

    処理を実行

    gettimeofday(&tv_end, NULL); /* 終了時刻を記憶する */

    time =
        (double)(tv_end.tv_sec-tv_start.tv_sec)+ /* 小数点以上 */
        (double)(tv_end.tv_usec-tv_start.tv_usec)/1000000.0; /* μ秒部分 */
    printf(“%.6f\n”, time); /* 小数点以下6桁表示 */
}
```

実習課題[2] 単純な同期問題を、pthread を使って実際に体験する

Pthread の環境で、生産者・消費者問題(producer-consumer problem)をプログラムし、実行せよ。プロセス数を2とし、うまく動作するかを確認せよ。(正しく動作していることを見せられるように工夫せよ)

生産者・消費者問題の参考ページ (自分でも検索してみよ) :

ウィキペディア「セマフォ」: <https://ja.wikipedia.org/wiki/%E3%82%BB%E3%83%9E%E3%83%95%E3%82%A9>

慶応寺岡先生の授業のページ: <http://www.tera.ics.keio.ac.jp/person/tera/os/2004/OS-04-20041019.pdf>

なお、2013 年度のコンピュータ科学特論の授業ページも参考になるかもしれない。(今回とは少し違うが。)

<http://pepper.is.sci.toho-u.ac.jp> ⇒ 左の目次「授業のページ」⇒ 2013 年度授業「大学院コンピュータ科学特論」