C言語でスレッド (Pthread)

2007年1月11日

海谷 治彦

目次

- 簡単な例
- Mutex
- 同期

コンパイルの仕方

- Mac10の場合，ただコンパイルすればよい.
- Linux, BSD他，一般的なLinuxでは，

  gcc なんか.c -lthread

  と追加ライブラリの指定が必要.
- 無論，pthread用のヘッダーがいる
  - 以降のサンプルで

sample1.c

- 最も基本的なコードの形
- pthread_create でスレッドを作成および開始
  - この時点で実行の中身となる関数を指定する
- pthread_join で指定スレッドが終わるのを待つ
  - コレがないとthread実行が始まる前にmainスレッドが終わってしまう.
  - ためしにコメントアウトしてみよう.

- なんか実行してもあまり並列処理っぽくない(涙)
- pthread_setconcurrency があると並行っぽい
  - 利用したい CPU 数をシステムに要求する関数.

sample2.c

- 実際に実行される関数への引数を渡す方法.
- 基本的にアドレスを1個しかわたせない.

sample3.c

- usleepを使って，並列っぽく見せた.

sample4.c

- バッファーにある文字列をログファイルに0.1秒おきに書き込む.
- キーボード入力があったら，バッファーを更新する.
- 結果として，入力された文字の時系列情報を記録できる.

- 並列動作っぽい.

sample5.c

- クイックソート

- あんまり実行結果も面白くないけど，効果的なスレッド化の例.
- もし，複数のCPUを使ってスレッドを実行できれば，ホントに早い (はず).

mutex

- 変数等の相互排斥を行うための仕組み.
- Javaのsynchronized とほぼ役目は同じ.

sample6.c sample7.c

- 二つのスレッドが共有変数をそれぞれ加算する.
- たした数が合わない.
- 一時的に関数のローカル変数にいれようが, いれまいが, ダメ.

sample8.c

- mutex = Mutual Execlusion Lock = 相互排除ロック
- pthread_mutex_t をlock/unlockして排他制御.
- 要は同じmutexを同時にlockできるのは1つのスレッドのみ.
- Javaのsynchronized にほぼ同じ.

同期

- Javaのwait, notify, notifyAllに相当するロック中のスレッドを起こす方法.
- Javaの時同様, いわゆる「生産者・消費者問題」で説明.

sample9.c

- 生産者，消費者問題 pthread版
- はっきりいってJavaをみながら移植.
- 生産者，消費者はそれぞれ一人.

Javaとの対比

Java	Pthread
new Thread() start()	pthread_create()
join()	pthread_join()
synchronized	pthread_mutex_lock() pthread_mutex_unlock()
wait	pthread_cond_wait()
wait(long)	pthread_cond_timedwait()
notify()	pthread_cond_signal()
notifyAll()	pthread_cond_broadcast()