

**Subject:** Lecture on Operating System Lecture Exercise 6.2

**From:** IKENOYA Katsutoshi <j05002@ie.u-ryukyu.ac.jp>

**Date:** Wed, 17 Jan 2007 17:58:22 +0900

**To:** Shinji KONO <kono@ie.u-ryukyu.ac.jp>

学籍番号 : 055702B

## 問題6.2

flock, pthread\_mutex, symbolic link のそれぞれの同期について、用途、利点と欠点などについて400字で記述せよ。

### (1)flock

オープンされたファイルにロックの適用、解除を行う。  
ロックモードには共有ロック、排他ロックなどがある。  
flockはシステムが提供するロック機構であるため、  
確実なロックが期待でき、使い方も簡単である。  
しかし、flock()を呼び出すと、ロックが解除されるまで  
制御が戻らないので処理の効率が悪くなってしまう場合がある。

### (2)pthread\_mutex

mutex は、排他制御の仕組みであり、共有データの同時更新 からの保護、  
クリティカルセクション (critical section) や モニタの実装などに使われる。  
mutexはアンロック状態とロック状態の二つの状態を取りうる。  
二つの異なるスレッドが同時に一つのmutexを保有するということはない。  
他のスレッドによってロックされたmutexをロックしようとするスレッドは、  
保有 側のスレッドが先にそのmutexをアンロックするまで実行を停止させられる。  
また、mutexには、デッドロックが検出されないという欠点がある。

### (3)symbolic link

ファイルやディレクトリに別の名前を付け、ユーザーやアプリケーション  
が、その名前を本体同様に扱えるようにするもの。  
symbolic linkは実行時に解釈されリンクの内容でパスを置き換えて  
そのパスを辿ることで、ファイルやディレクトリに到達する。  
if symlink(\$datafile,\$lockfile)でロックの検査と設定とがシステムコール  
レベルで行われるので、これらに間に隙間が生じない。  
しかし、プログラムが何らかの原因で異常終了してしまうと、プロセスが作成  
したシンボリックリンクが残ったままで常にロックがかかってしまう状態に陥る  
可能性があるという欠点がある。