

日曜研究室

技術的な観点から日常を綴ります

[xv6 #41] Chapter 3 – Locking – Real world

テキストの49ページ

本文

ロックを使ったプログラミングは未だ難しくやりがいがあるので、同期プリミティブと並列プログラミングは活発に研究が行われている。

同期化されたキューのような高レベルのものの基礎としてロックを使うのが（xv6はそれを行っていないとはいえ）一番良い。

もしあなたがロックを使ってプログラミングする場合、ロックを要求するようなインバリアントの保護は失敗しやすいので、競合状態を調査するためのツールを使うのは賢明である。

ユーザレベルのプログラムもロックを必要とする。

しかしxv6のアプリケーションはひとつの実行スレッドを持ち、複数のプロセスはメモリを共有しない。

よってxv6のアプリケーションでは、ロックを用いる必要はない。

アトミックな命令は、ロックが満たされたときでも自由ではない。

（追記：ここは「ロックを獲得した後もアトミックな命令が必要な場合がある。」のほうが近いだろうか）

あるプロセッサがそのローカルキャッシュにロックを持ち、そして他のプロセッサはロックを獲得しなければならない場合、そのプロセッサのキャッシュから他のプロセッサのキャッシュへそのラインを移動し、そしておそらくそのキャッシュラインの他のコピーを無効にしなければならない。

他のプロセッサのキャッシュからキャッシュラインを取り出す事は、ローカルキャッシュからラインを取り出すよりコストが高くなる。

ロックに関わるコストを避けるために、多くのOSはロックフリーなデータ構造とアルゴリズムを利用し、そのアルゴリズムのなかでアトミックな操作をできるだけ避けようとしている。

例えば、この章のはじめで出てきたような、“リストの検索にロックが必要なく、リストにアイテムを追加するとき一つだけアトミックな命令を使うリンクリスト”を実装することは可能である。

感想

「アトミックな命令は、ロックが満たされたときでも自由ではない。」のところはよく分かりませんが、例えばxv6ではロック関連の処理でxchgというアトミックな命令を使うけど、それ以外にも使うよ。ってことかなと思います。

もしくは、xchgの実行には、ロックが獲得できないとき（スピンして待つ）に比べてロックが獲得できる状態では時間はかからないけど、ロックのキャッシュが他のCPUコアのキャッシュに載ってるときはそれを移動しなきゃならないので、全くコストがかからないわけじゃないよ。ってことかもしれません。

キャッシュラインという言葉が出てきましたが、概要については [【コラム】コンピュータアーキテクチャの話 \(6\) キャッシュの構造\(基礎編\) – どういう単位でキャッシュに入れるのか? | エンタープライズ | マイナビニュース](#) を、ラインの移動の問題については [【コラム】コンピュータアーキテクチャの話 \(194\) キャッシュラインのフォールスシェアリング | エンタープライズ | マイナビニュース](#) を見ると分かりやすいかと思います。

カテゴリー: 技術 | タグ: xv6 | 投稿日: 2012/3/14 水曜日 [<http://peta.okechan.net/blog/archives/1529>] |
