

日曜研究室

技術的な観点から日常を綴ります

[xv6 #40] Chapter 3 – Locking – Memory ordering

テキストの48～49ページ

本文

この章では、プロセッサが、プログラムの中で現れる順番で一連の命令を開始し完了すると仮定する。

しかしながら多くのプロセッサは、より高いパフォーマンスを達成するために命令を順番通りには実行しない。（アウトオブオーダー実行）

ある命令が、その完了までに多くのサイクルを要する場合、他の命令とオーバーラップして実行しプロセッサのストールを避けるために、プロセッサはその命令を早めに発行したいはずである。

例えば、プロセッサは直列に並んでいる命令Aと命令Bがお互いに依存してない事に気づき、命令Aを完了するときに命令Bも完了させるために、命令Aの前に命令Bを開始する。

しかしながら同時実行は、命令の並び替えによってソフトウェアに影響を与え、間違った振る舞いを発生させるかもしれない。

例えば、`release`関数が`xchg`命令を使わずに`lk->locked`に0を割り当てる場合、何が起きるのか心配しなければならないだろう。

x86プロセッサは、世代によって保証するメモリの順序性に違いがあるため、この問題に対する答えは不明瞭である。

`lk->locked=0`が`popcli`関数の後に並び替え可能だった場合、ロックが解放される前に他のスレッドで割り込みが有効になる可能性があるため、`acquire`は壊れるかもしれない。

（ここ特に訳に自信なし。原文: If `lk->locked=0`, were allowed to be re-ordered say after `popcli`, then `acquire` might break, because to another thread interrupts would be enabled before a lock is released.）

メモリの順序性に関する不明瞭なプロセッサの特性に頼るのを避けるため、xv6は、プロセッサが並び替えしない事を保証しなければならない`xchg`命令を使って、リスクを犯さないようにしている。

`spinlock.c`の`release`関数

```
01 // Release the lock.
02 void
03 release(struct spinlock *lk)
04 {
05     if(!holding(lk))
06         panic("release");
07
08     lk->pcs[0] = 0;
09     lk->cpu = 0;
10
11     // The xchg serializes, so that reads before release are
12     // not reordered after it. The 1996 PentiumPro manual (Volume 3,
13     // 7.2) says reads can be carried out speculatively and in
14     // any order, which implies we need to serialize here.
15     // But the 2007 Intel 64 Architecture Memory Ordering White
16     // Paper says that Intel 64 and IA-32 will not move a load
17     // after a store. So lock->locked = 0 would work here.
18     // The xchg being asm volatile ensures gcc emits it after
19     // the above assignments (and after the critical section).
20     xchg(&lk->locked, 0);
21
22     popcli();
23 }
```

感想

ロック機構は小さな穴さえ無いよう慎重に作られていますが、CPUのアウトオブオーダー実行の機能の差異によってロック機構の作り方次第では穴が出来ちゃうよ、という話ですかね。

release関数内の長いコメントにもより具体的に似たような事が書いてありますね。

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/1441>] |
