

# オペレーティングシステム・期末試験の解答と解説

2013年度E・Oクラス (2014年2月13日・試験時間90分)

1. (a) 正しく行われない

**理由** lk->lockedの読み出しと書き込みがアトミックに行われず、while文と続く代入文の間に他CPUが書き込みを行う可能性があるため。

**解説** 講義で説明したアルゴリズム0に相当する。

(b) 正しく行われる

**理由** 割込みが禁止されていることで、最初にロックを獲得したスレッドが全てのロックを解放するまで他のスレッド（および割り込みハンドラ）は一切動作できないため。

**解説** 講義スライド04の「割り込み禁止による相互排除」(p.60)に相当する。タイマー割り込みを含む全ての割り込みが禁止されるため、割り込みハンドラは起動せず、CPUの横取りも生じない。よってこれらによるlk->lockedの参照・変更は起らない。既に獲得したロックを解放せずに再び獲得しようとするのでデッドロックになるが、これはプログラムが正しくかけていけば生じないはずなので無視してよい（実際のXv6のコードではpanicにしている）。

(c) 正しく行われない

**理由** あるスレッドがロックを獲得し、それを解放する前に同じCPU上で割り込みが発生したとする。その割り込みのハンドラが同じロックを獲得しようとした場合、そのハンドラはロックが解放されるまで待ち、割り込まれたスレッドはハンドラの終了を待つことになり、結果としてデッドロックになるため。

**解説** Xv6では割り込みハンドラ内でロックを獲得しようとすることもある(idelock)。

(d) 正しく行われない

**理由** 問題(c)と同じ理由。

**解説** 問題(c)のケースは同一CPU上のスレッドと割り込みハンドラ間でのデッドロックなので、単一CPUの場合も避けることはできない。

(e) 正しく行われない

**理由** stiによってすべてのロックが解放される前に割り込み禁止が解除されてしまうと、問題(c)(d)の場合と同じように割り込みハンドラによるデッドロックが生じることがあるため。

**解説** 元のコードでは、pushcliを実行した回数だけpopcliを実行して、初めて割り込み禁止が解除されるようになっている。これによって全てのロックが解放されるまで割り込み禁止状態が保たれることになる。

(f) 以下は活性の例である。

- どのプロセスもクリティカルセクションに入れないということはない（デッドロックの回避）。
- クリティカルセクションに入ろうとしたプロセスは、いつかは必ず入ることができる（飢餓の回避）。

(g) (2) スリープロック

**解説** 恒例の四択問題である。(1)のビジーロック(busy lock)であるが、あまりこうは言わず普通はビジーウェイト(busy wait)と言う。これはスピンロックと同義である。

(4)のグローバルインタープリタロック(global interpreter lock, GIL)は、プログラミング言語のインタプリタの実装における、スレッド間で共有される（単

一) 資源を保護するために導入されるロックのことをいう。マルチプロセッサマシン上で被実装言語のスレッドを並列動作させたい場合、GIL が存在すると実行速度に悪影響を及ぼす。

(3) のマスロック (math rock) は音楽ジャンルの一つで同期機構とは何の関係もない<sup>1</sup>。“Math”と称しているものの数学とは無関係な単なる虚仮威しで、例えば厳密なアルゴリズムに基づく作曲法などを期待して聴くのがっかりする。

(h) スピンロックが利用される例：

- カーネル内部での同期,
- (スリープロックが使えない場合の) ユーザスレッドの同期,
- 待ち時間がごく短いことが期待される場合の同期.

2. (a) 2M バイト (2097152 バイト)

**解説** 512 バイトのブロックが 4096 個あるので、単純に  $512 \times 4096 = 2097152$  となる。

(b) 63

**解説** 大きさ 64 バイトの inode (正確には dinode 構造体) が 500 個なので、inode 領域は  $500 \times 64 = 32000$  バイト必要。したがって必要なブロック数は  $\lceil 32000 \div 512 \rceil = 63$  個。

(c) 1 (2 でも可)

**解説** 総ブロック数が 4096 なので、ビットマップのビット数も 4096 となる。従ってビットマップのバイト数は  $4096 \div 8 = 512$  で、ブロック数は 1 となる。

(d) 4010

**解説**  $4096 - (1 + 1 + 63 + 1 + 20)$ .

(e) もし nlink が実際の参照数より多い場合、ディレクトリからの参照を全て unlink で消しても inode は消去されず、そこから参照されているデータブロックも (実際には使われていないのに) 使用中のままになる。逆にもし nlink が実際の参照数より小さい場合は、まだ参照しているディレクトリがあるのにファイルが消去されることがある。

---

<sup>1</sup>そもそも rock であって lock ではない