

オペレーティングシステム・試験問題

2010 年度 E・O クラス (2011 年 2 月 8 日・試験時間 90 分)

書籍, 配布資料およびノート等は参照してはならない.

1. s をセマフォ, k を非負整数とする. $s.count$ の初期値を k , $s.waiting$ の初期値を ϕ (空集合) とする. s が作られてからの $P(s)$ と $V(s)$ の実行回数をそれぞれ $\#P(s)$ および $\#V(s)$ とする. これらの値は増加することはあっても減ることはない. ただし, あるスレッド (T とする) が $s.count = 0$ のときに $P(s)$ を実行して待ち (waiting) 状態になったときは, 他のスレッドが $V(s)$ を実行して T を起こす (notify する) までは T による $P(s)$ の実行は数えないものとする. また, $P(s)$ と $V(s)$ 以外は s に影響を与えないとする.

以下 (1)(2) は不変式 (invariant) である.

$$s.count \geq 0 \quad (1)$$

$$s.count = k + \#V(s) - \#P(s) \quad (2)$$

(2) が不変式であることを帰納法で証明しよう. $P(s)$ と $V(s)$ 以外は s に影響を与えないので, これらによる状態変化のみを考えればよい.

- 初期状態 (s が作られた状態) では $s.count = k$ であり, $P(s)$ も $V(s)$ もまだ実行されていないので $\#P(s) = \#V(s) = 0$. よって (2) は成立する.
- ある状態において (2) が成立しているとする (帰納法の仮定). このときあるスレッド T が $P(s)$ または $V(s)$ を行ったときを考える.

- $P(s)$ を実行したとする. 以下の (P1)(P2) より実行後も (2) は成立する.

(P1) $s.count = 0$ であった場合: $s.count$ は不変である. また T は待ち状態に入るので定義より $\#P(s)$ も不変.

(P2) $s.count > 0$ であった場合: $s.count$ の値は 1 減る. T は待ち状態に入らずに $P(s)$ を終了するため $\#P(s)$ は 1 増加する.

- $V(s)$ を実行したとする. 以下の (V1)(V2) より実行後も (2) は成立する.

(V1) $s.waiting = \phi$ であった場合: $s.count$ と $\#V(s)$ はそれぞれ 1 増加する.

(V2) $s.waiting \neq \phi$ であった場合: $s.count$ は不変であり, $\#V(s)$ は 1 増加する. ここで $s.waiting$ に入っているスレッドが 1 つ選ばれて notify されるので, $\#P(s)$ は 1 増加する.

以上より (2) は不変式であることが示せた. 同様に (1) も不変式であることを示すことができる.

(a) s をセマフォとし, $s.count$ と $s.waiting$ の初期値をそれぞれ k, ϕ とする ($k > 0$). N 個のスレッドがあり, 全て以下のプログラムを実行しているものとする.

```
while (true) {  
    P1: NC  
    P2: P(s);  
    P3: CS  
    P4: V(s);  
}
```

ここで CS を実行しているスレッドの数を $\#CS$ とする. 以下の (3) が不変式であることを帰納法によって示せ.

$$\#CS = \#P(s) - \#V(s) \quad (3)$$

(b) (1)(2)(3) が不変式であることを用いて以下の (4) が不変式であることを示せ.

$$\#CS \leq k \quad (4)$$

(c) 弱いセマフォ (weak semaphore) とはどのようなものか. また, 強いセマフォ (strong semaphore) と呼ばれるものの一例を挙げ, その性質を述べよ.

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
ページフォルト	F	F	F	F	F	F	F	F	F				F	F	
物理ページ割当 (ページフレーム数=4)	0	0	0	0	1	2	3	4	0	0	0	0	1	2	2
		1	1	1	2	3	4	0	1	1	1	1	2	5	5
			2	2	3	4	0	1	2	2	2	2	5	3	3
				3	4	0	1	2	5	5	5	5	3	4	4

図 1: ページフレーム数 4 のときの FIFO アルゴリズムによるページ置換

2. 図 1 はページフレーム数 4 のときに FIFO アルゴリズムによってページ置換を行った様子を表している。参照列の欄の各要素は要求されたページ番号であり、左から右に向かって順に要求が発生したものとする。ページフォルトの欄における F は、当該要求によってページフォルトが発生したことを示している。物理ページ割当の欄はページフレームの様子を表している。

3. Unix のファイルシステムにおいて、i-node が置かれるブロックについては単純な write-back キャッシュを用いない方がよい理由を述べよ。

(a) ページフレーム数=4 で図 1 に示したページ要求列が発生したとき、LRU アルゴリズムによるページ置換を行ったときのページフォルト数を答えよ。

(b) ページフレーム数=5 で図 1 に示したページ要求列が発生したとき、LRU アルゴリズムによるページ置換を行ったときのページフォルト数を答えよ。

(c) ページフレーム数=4 で図 1 に示したページ要求列が発生したとき、最適 (optimal) アルゴリズムによるページ置換を行ったときのページフォルト数を答えよ。

(d) ページフレーム数を増やせばページフォルト数は減る。FIFO アルゴリズムによるページ置換を行った場合、この主張は正しいか。正しいければそのことを示し、そうでなければ反例を示せ。