プロセス間同期テスト結果

内容

1. テストプログラム 1
2. テスト PC 環境 2
3. テスト結果 2
4. 考察 3

1.　テストプログラム

テストプログラムの一覧を示す。

詳細については、No.0～No.7 のフローチャートを付録 A に示す。(No.8,No.9 は No.0 と同じ)

(1) プロセス間同期

プロセス間同期に関するテストプログラムの一覧を表 1.1 に示す。

同期に関して使用したプログラム要素も合わせて示す。重要項目を黄色で示す。

表 1.1 プロセス間同期テストプログラム

No	テストプログラム条件	起動確認	メインプロセス →サブプロセス	サブプロセス →メインプロセス	終了確認
0	元のテストプログラム	セマフォ	セマフォ	Sleep	Sleep
0+	No.0→計算処理回数100倍	セマフォ	セマフォ	Sleep	Sleep
1	No.0→同期なしとする	セマフォ	なし	なし	セマフォ
1a	No.1→Sleepを外し、 計算処理を入れる	セマフォ	なし	なし	セマフォ
1a+	No.1a→計算処理回数100倍	セマフォ	なし	なし	セマフォ
2	No.0→セマフォによる同期	セマフォ	セマフォ	セマフォ	セマフォ
2+	No.2→計算処理回数100倍	セマフォ	セマフォ	セマフォ	セマフォ
3	No.0→Sleepをタイマにする	セマフォ	セマフォ	待機可能タイマ	待機可能タイマ
4	No.0→イベントによる同期	イベント	イベント	イベント	イベント
4+	No.4→計算処理回数100倍	イベント	イベント	イベント	イベント

(2) マルチプロセスとマルチスレッド

CPU に負荷をかける計算処理をマルチプロセスおよびマルチスレッドで並列実行させるテストプログラムの一覧を表 1.2 に示す。

表 1.2 マルチプロセスおよびマルチスレッド条件

No	プロセス数	スレッド数(1プロセスあたり)
5	1	1
6	1	2
7	2	1

(3) CPU 割り当て

デュアルコア CPU の PC において、No.0 プログラム実行時に、メインプロセスとサブプロセスの CPU コア割り当てを替えた場合のテスト条件を表 1.3 に示す。

CPU 割り当てには、CPU 最適化ソフト Process Lasso を用いた。(URL : http://www.bitsum.com/)

表 1.3 CPU 割り当て条件

No		メインプロセス	サブプロセス
8	別CPUコア	CPU 1	CPU 0
9	同一CPUコア	CPU 1	CPU 1

2.　テスト PC 環境

表 2.1 に示す 3 つの PC 環境にてテストを行った。

表 2.1 テスト PC 環境

	CPUコア数	CPUクロック	メモリ	OS
IBMノート	1	1.6GHz	512MB	WindowsXP Professional SP3
kimuraPC	2	2GHz	2GB	WindowsXP Professional SP3
社長PC	4	2.67GHz	2GB	Windows7 Home Premium

3.　テスト結果

表 3.1 に示す結果が得られた。(No は「1. テストプログラム」の No に対応) 重要項目を黄色で示す。

表 3.1 テスト結果

No		CPUコア数			備考
		1(IBMノート)	2(kimuraPC)	4(社長PC)	
0	CPU負荷最大値(%)	49%	2%	5%	元となる同期処理プログラム
	実行時間(秒)	1秒	124秒	156秒	
0+	CPU負荷最大値(%)		9.8%		
	実行時間(秒)		156秒		
1	CPU負荷最大値(%)	51%	15%	5%	同期なしの場合もSleepありだと実行速度改善なし
	実行時間(秒)	11秒	156秒	156秒	
1a	CPU負荷最大値(%)		36%		
	実行時間(秒)		1秒		
1a+	CPU負荷最大値(%)		100%		同期させなければ、各CPU使用率100%が可能
	実行時間(秒)		37秒		
2	CPU負荷最大値(%)	65%	22%	6%	セマフォによる同期で実行速度改善
	実行時間(秒)	1秒	1秒	1秒未満	
2+	CPU負荷最大値(%)		53%		
	実行時間(秒)		37秒		
3	CPU負荷最大値(%)	47%	8%		
	実行時間(秒)	1秒	138秒		
4	CPU負荷最大値(%)	60%	20%		イベントによる同期で実行速度改善
	実行時間(秒)	1秒	1秒		
4+	CPU負荷最大値(%)		52.8%		
	実行時間(秒)		37秒		
5	CPU 負荷最大値(%)	100%	53%		1スレッドではCPU使用率は50%程度まで
	実行時間(秒)	33 秒	32 秒		
6	CPU 負荷最大値(%)	100%	100%		2スレッドでCPU使用率100%が可能
	実行時間(秒)	88 秒	37 秒		
7	CPU 負荷最大値(%)	100%	100%		2プロセスでCPU使用率100%が可能
	実行時間(秒)	66 秒	33 秒		
8	CPU 負荷最大値(%)		11%		別CPUのプロセス間同期で実行速度遅延
	実行時間(秒)		145 秒		
9	CPU 負荷最大値(%)		11%		同一CPUのプロセス間同期で実行速度改善
	実行時間(秒)		9 秒		

4. 考察

表 3.1 に示す結果を元に、以下に考察を記述する。

(1) プロセス間同期手段

メインプロセスとサブプロセスの間の同期手段について考察する。

CPU コア数=2 の PC(kimuraPC)において、No.0(セマフォ/Sleep による同期)の実行時間は 124 秒だったが、No.2(セマフォ/セマフォによる同期)および No.4(イベント/イベントによる同期)では共に 1 秒となり、実行時間を短縮することができた。CPU コア数=1 の場合は特に問題ないが、マルチコア(CPU コア数≧2)の場合は、Sleep による同期は実行速度の大幅な遅延の原因と考えられる。マルチコアの PC では、セマフォまたはイベントに置き換えることが、実行速度改善に有効である。

(2) CPU 使用率の限界

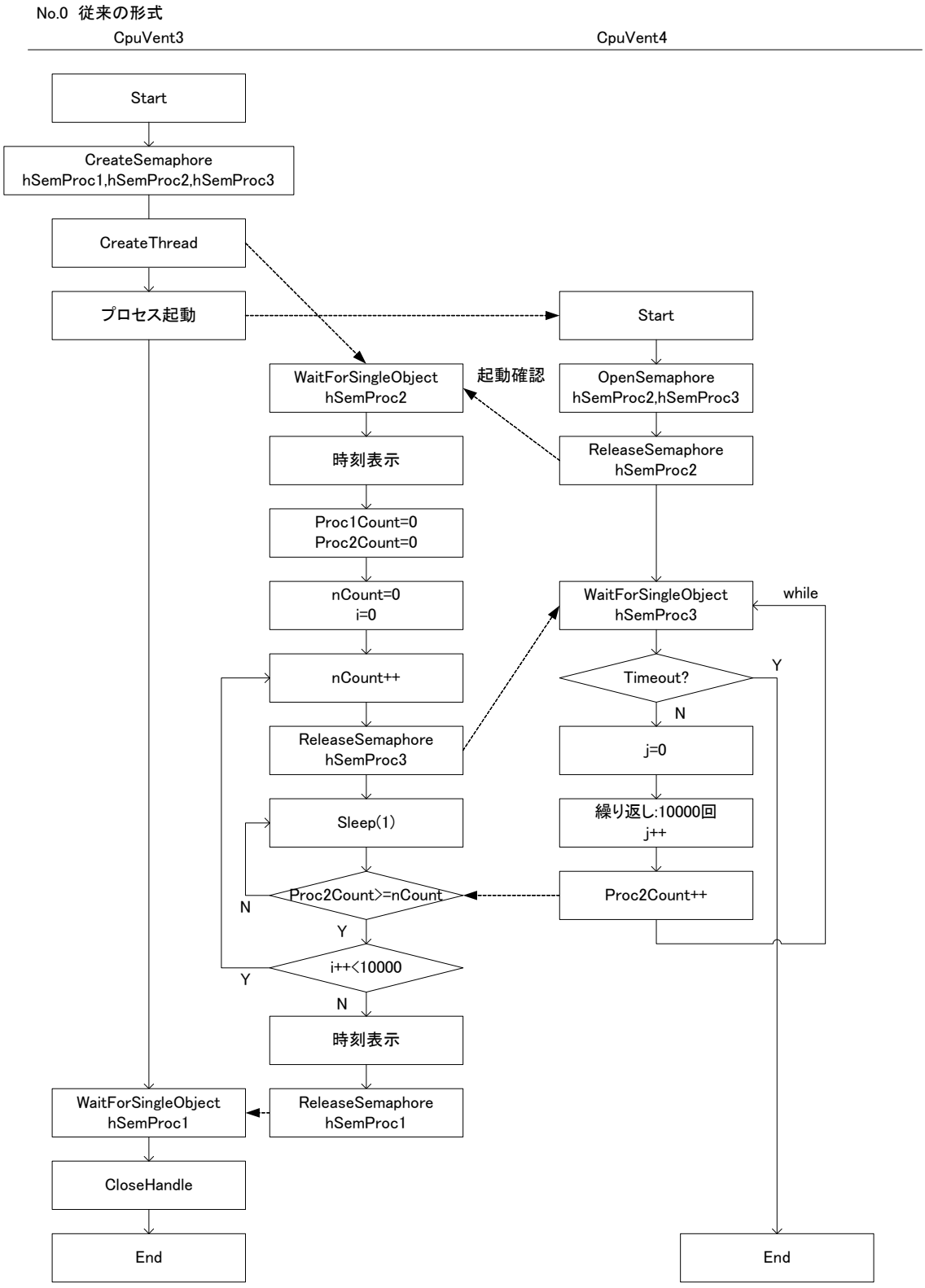
CPU コア数=2 の PC(kimuraPC)において、No.0(セマフォ/Sleep による同期)の CPU 使用率は 2%と低く、CPU の力を充分使い切っているとは言えない。そこで、CPU 使用率 100%とすることが可能かの確認を行った。No.0 から同期処理をなくし、メインプロセスとサブプロセスの両方に充分に負荷の大きい計算処理を入れて実行した結果、CPU 使用率が 100%となった(No.1a+)。よって、負荷が大きい処理ならば CPU 使用率を使い切ることができることが確認できた。

ただし、No.5(1 スレッドで高負荷の処理)の CPU 使用率が 53%であることから推察できるように、CPU コア数=2 の PC(kimuraPC)において、1 つのスレッドでは 1 つの CPU コアを使用するため、CPU 使用率は 50%までとなる（CPU_1 の使用率＝100%、CPU_2 の使用率＝0%）。よって、CPU 使用率 100%まで使い切るには、CPU コア数以上の数のスレッドに分割することが必要となる。

付録 A. テストプログラムフローチャート

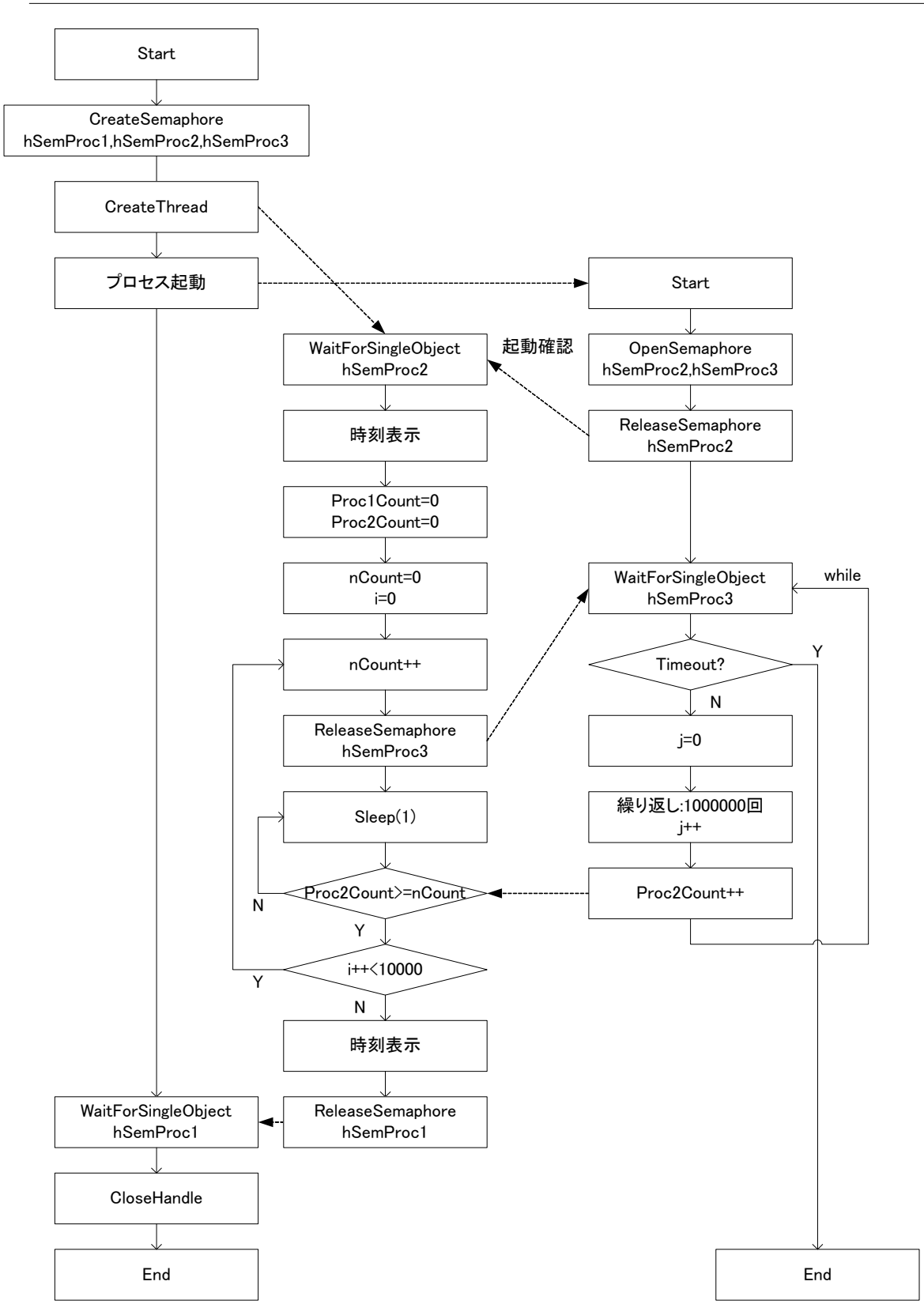
テストプログラムのフローチャートを以下の<No.0>～<No.7>に示す。

<No.0>



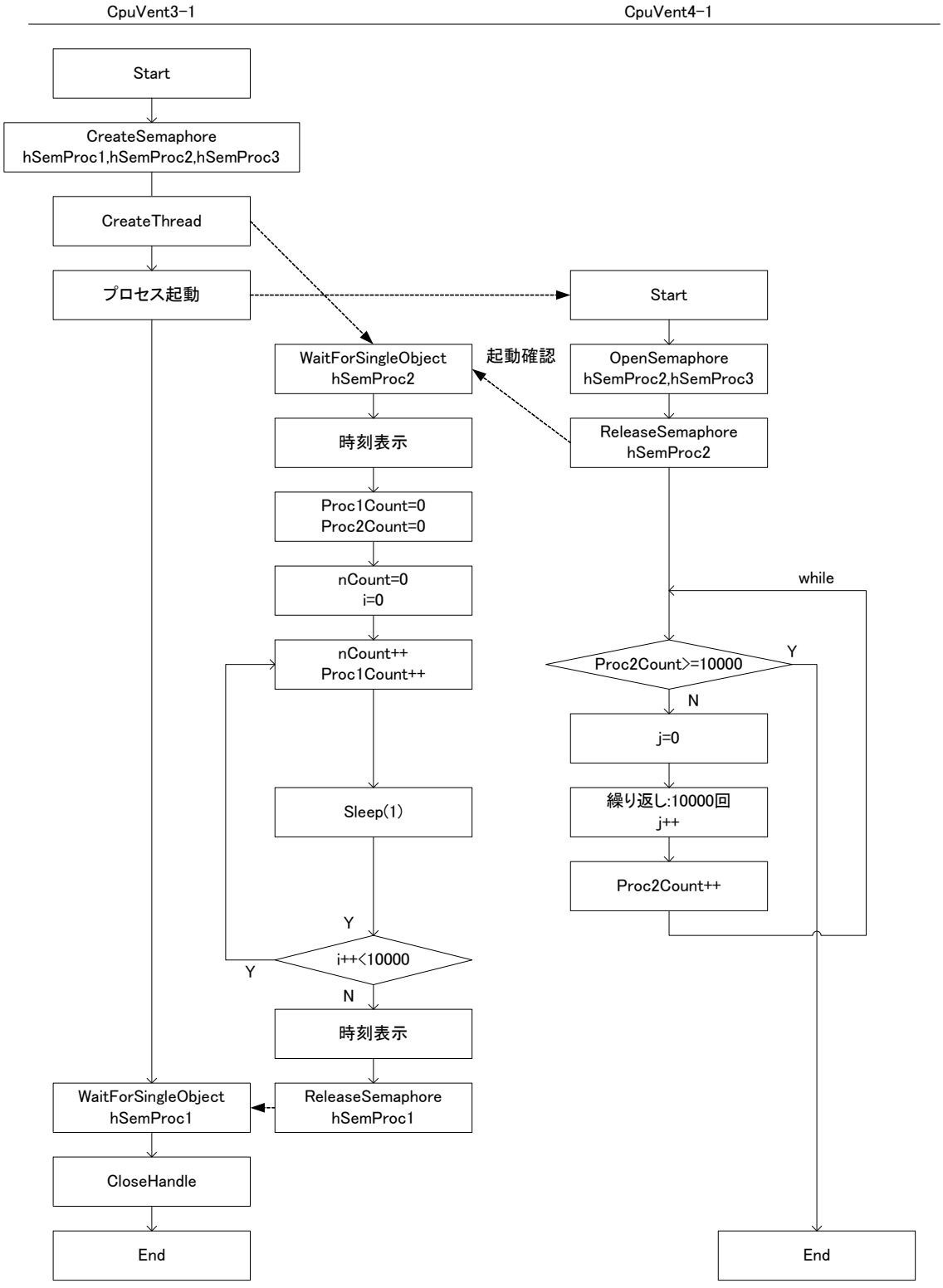
<No.0+>

No.0+ No.0→サブプロセスCpuVent4の計算処理の回数を100倍した



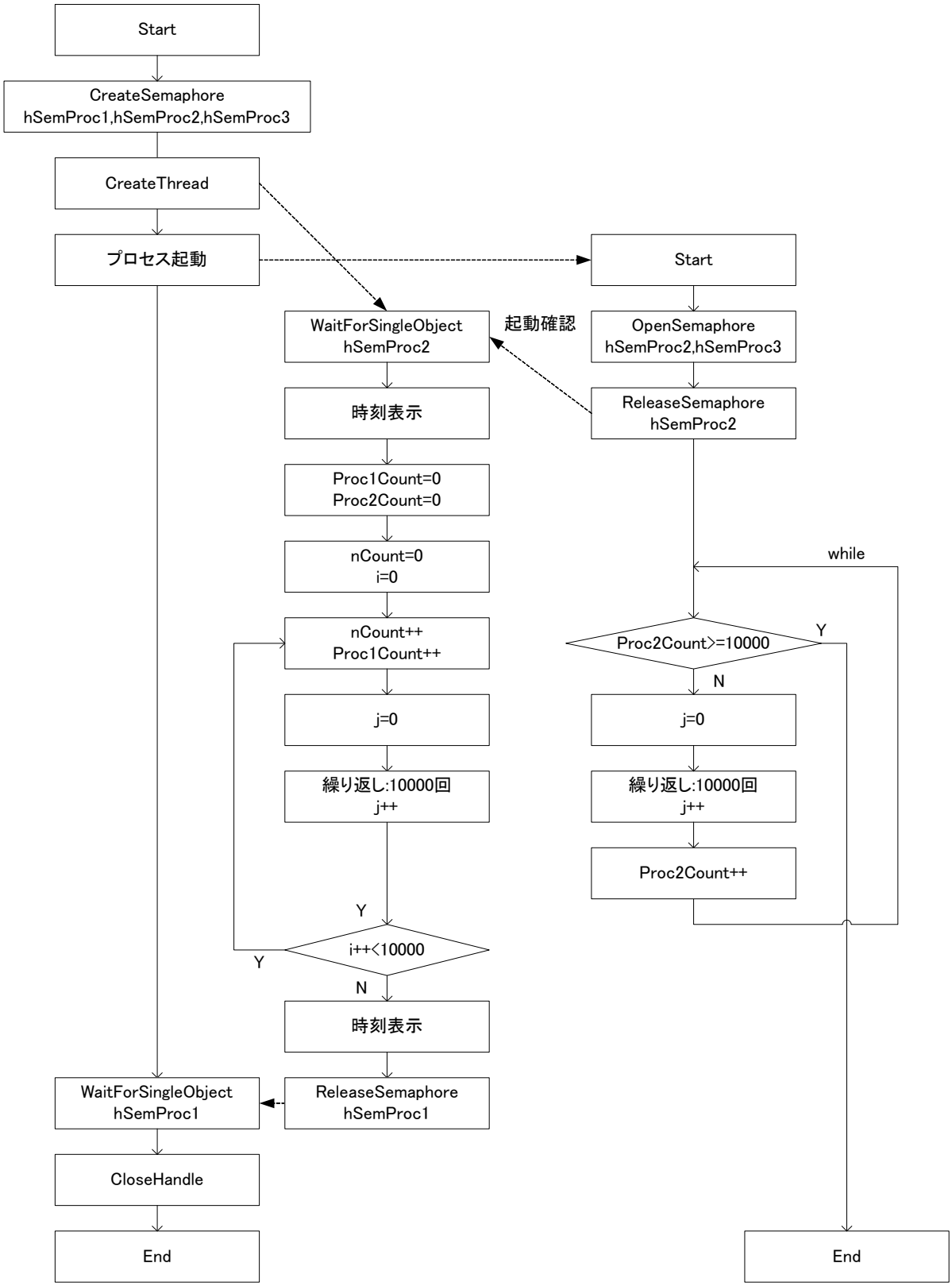
<No.1>

No.1　No.0→同期なしとする



<No.1a>

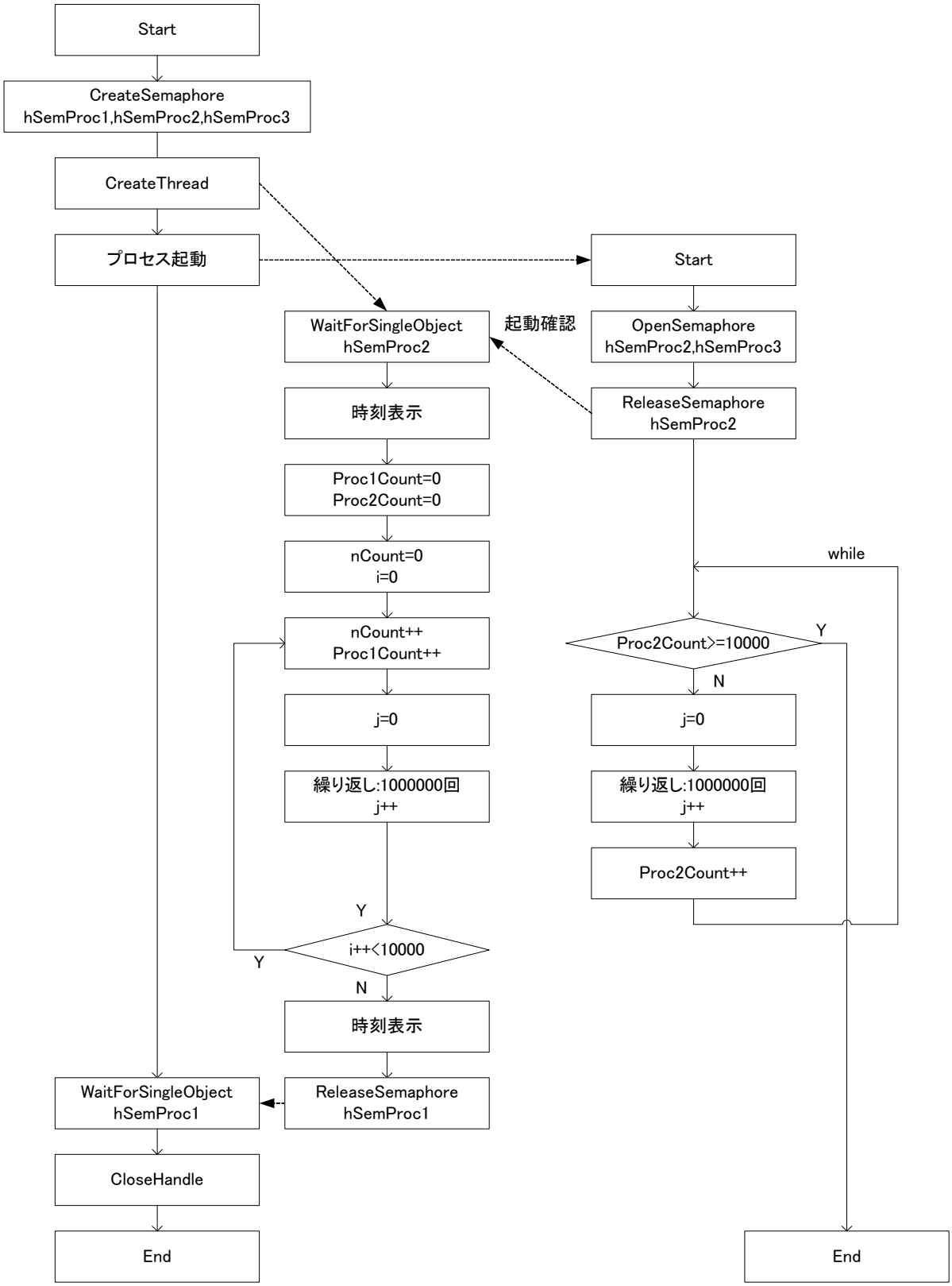
No.1a No.1→メインプロセスCpuVent3-1のSleepを計算処理に入れ替え



<No.1a+>

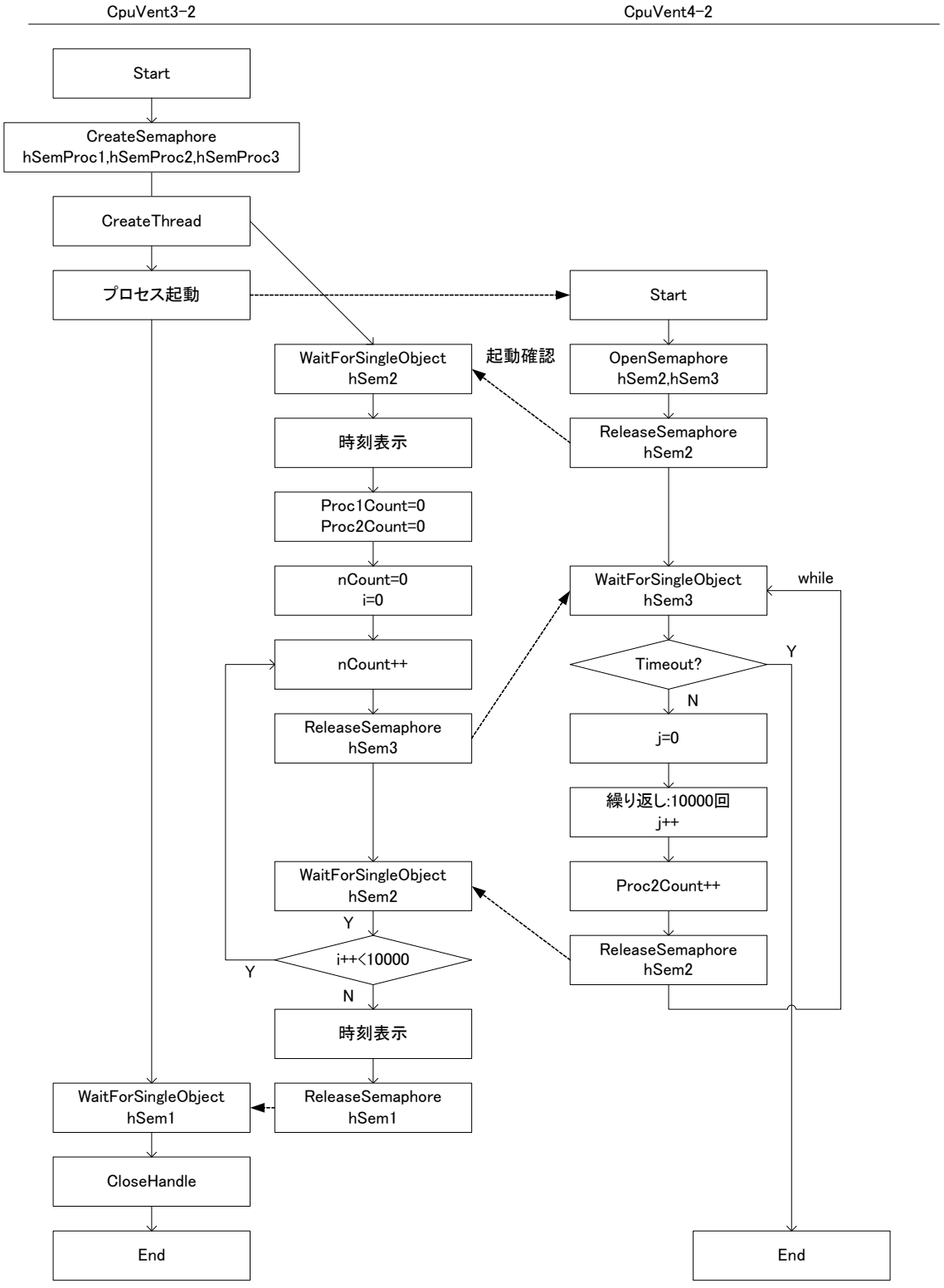
No.1a+　No.1a→計算処理の回数を100倍した

CpuVent3-1　　CpuVent4-1



<No.2>

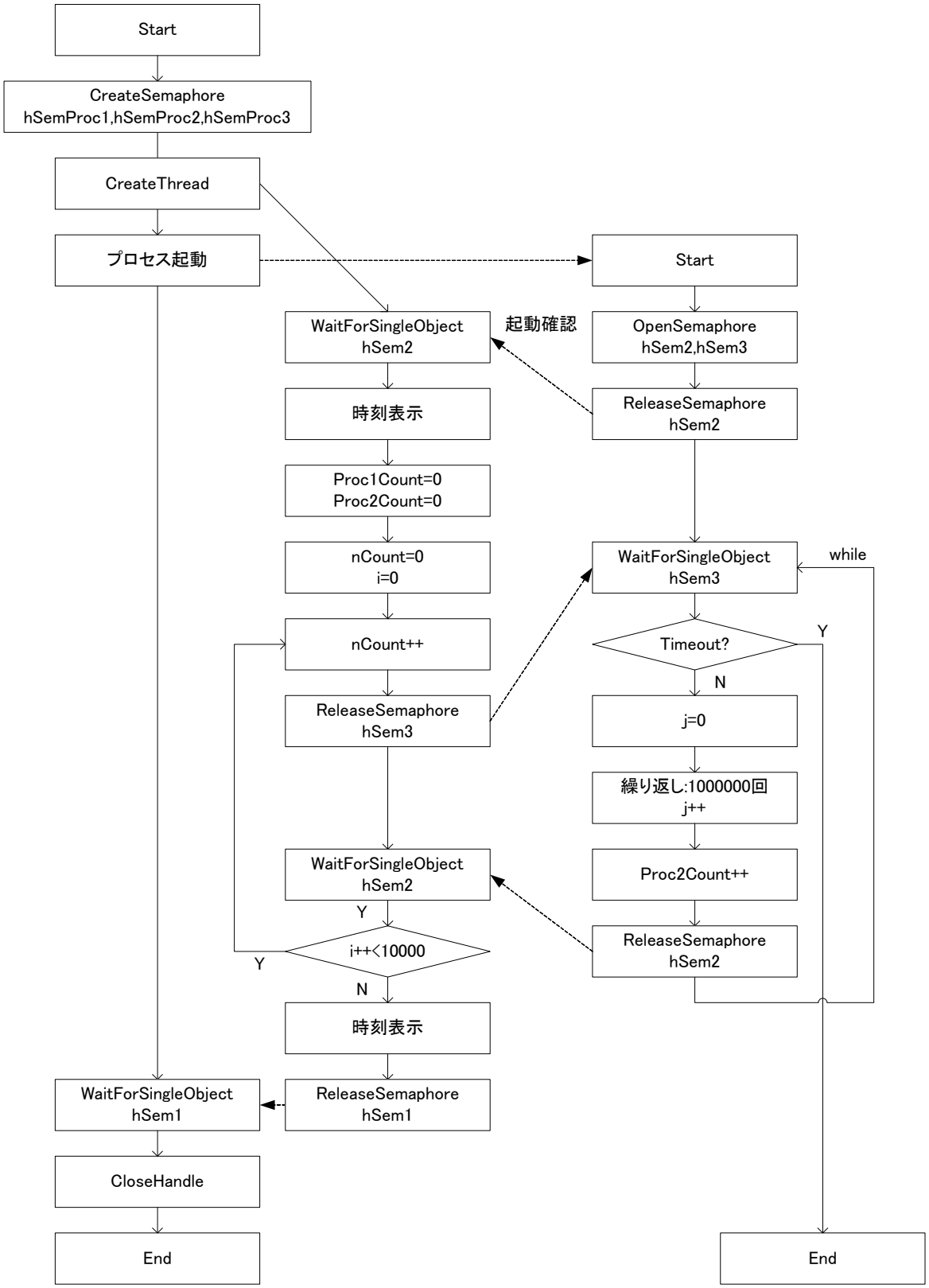
No.2 セマフォによる同期



<No.2+>

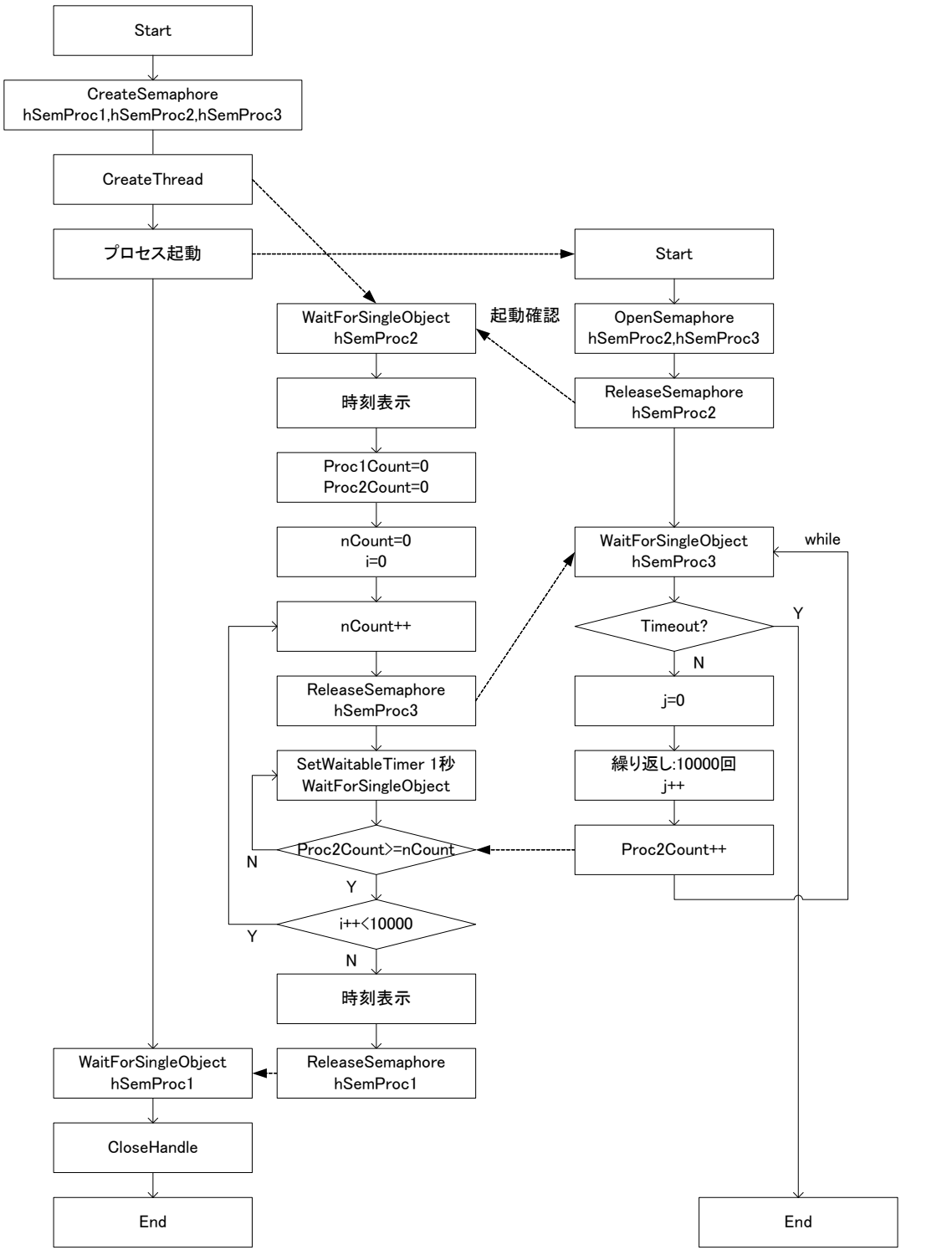
No.2+ No.2→サブプロセスCpuVent4-2の計算処理の回数を100倍した

CpuVent3-2　　CpuVent4-2



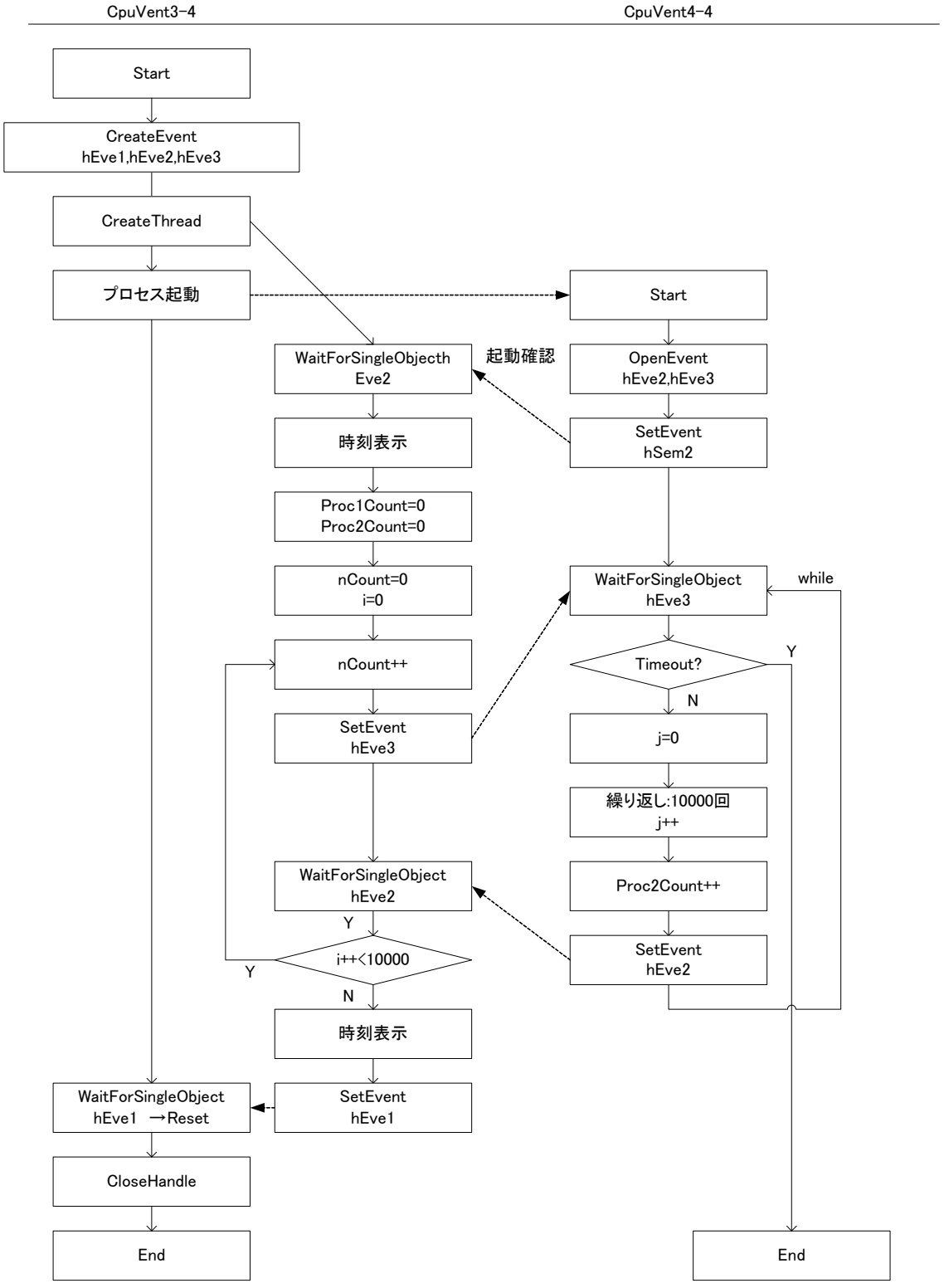
<No.3>

No.3　No.0→Sleepを待機可能タイマーとする



<No.4>

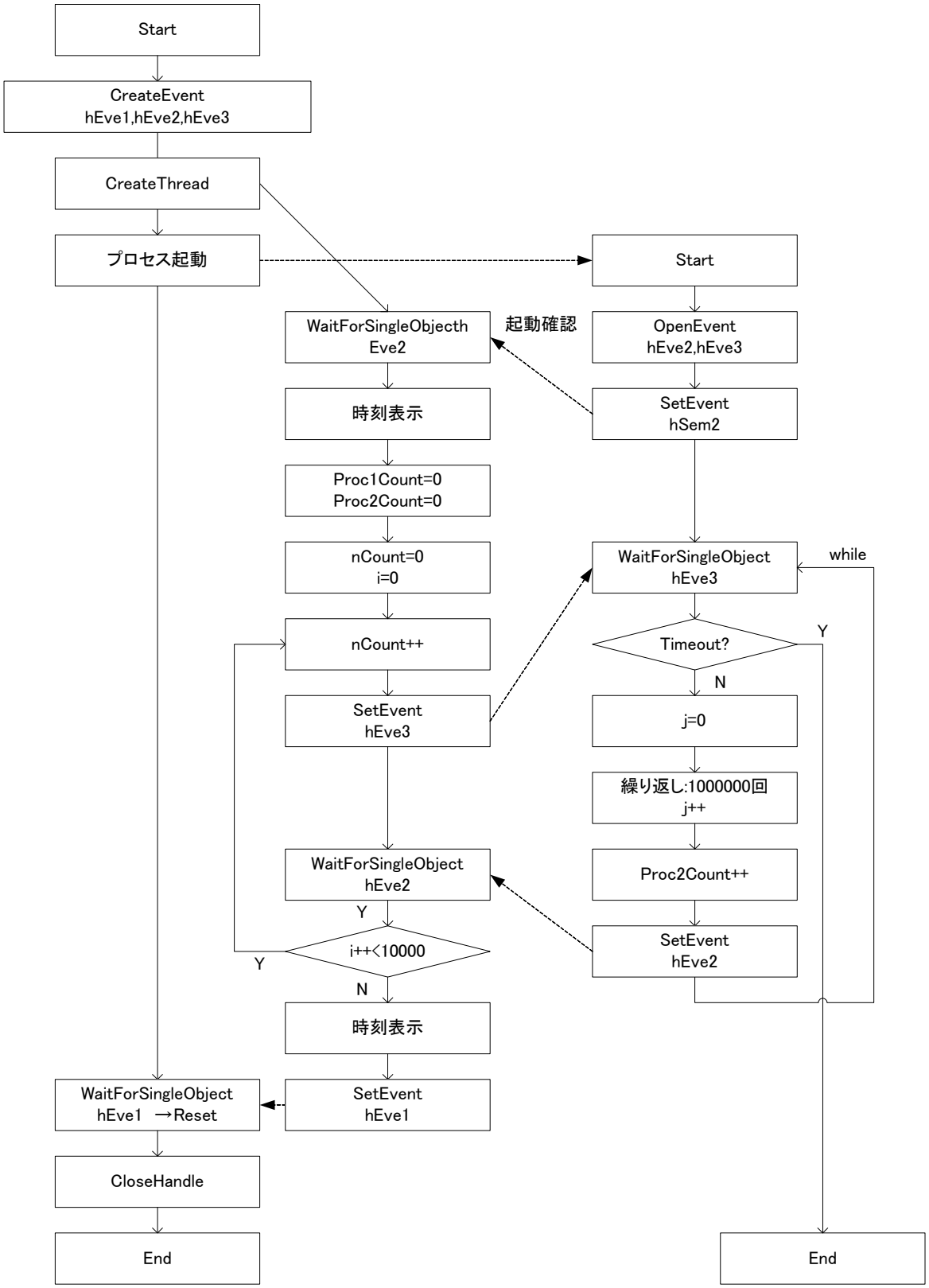
No.4 イベントによる同期



<No.4+>

No.4+ No.4→サブプロセスCpuVent4-4の計算処理の回数を100倍した

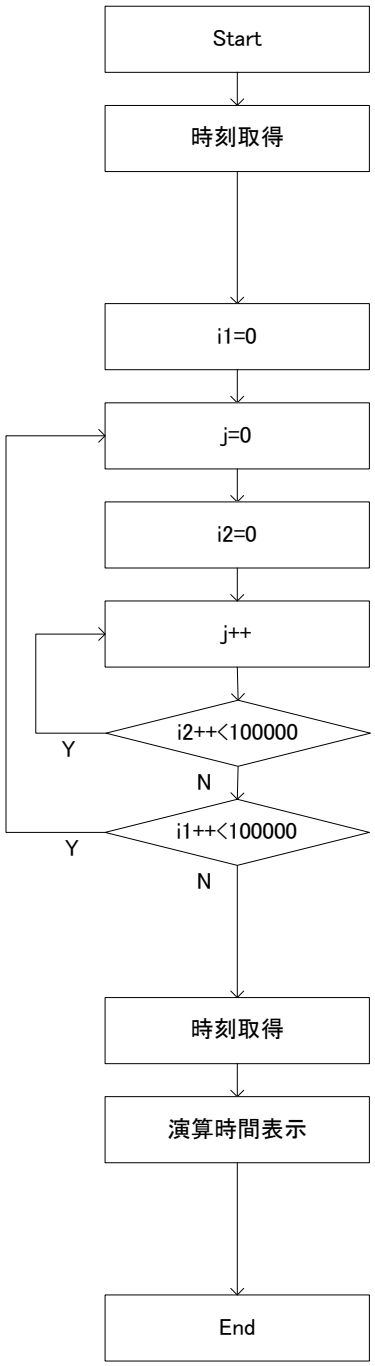
CpuVent3-4　　CpuVent4-4



<No.5>

No.5 CPUに負荷をかけるプログラム

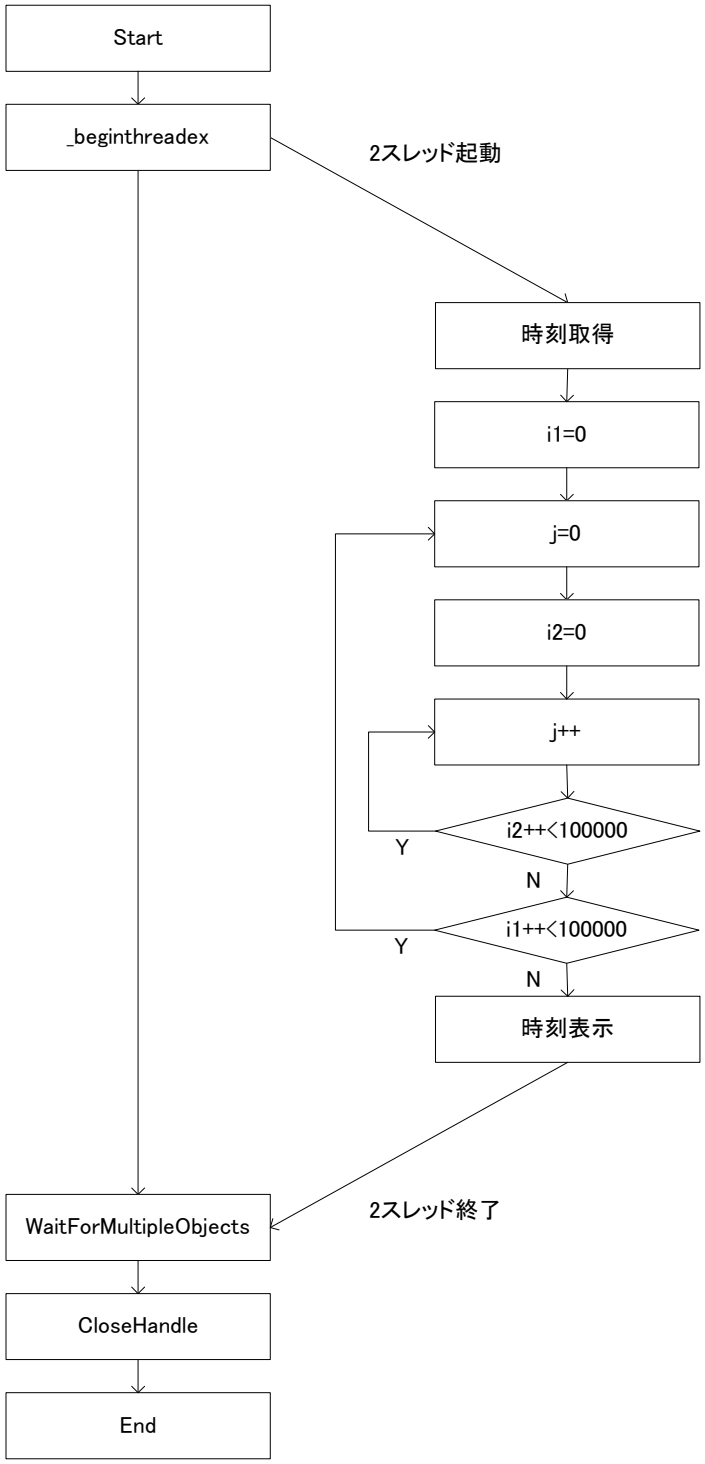
CpuVent2



<No.6>

No.6 CPUに負荷をかけるスレッドを2つ起動

CpuVent2



<No.7>

No.7 CPUに負荷をかけるプロセスを2つ起動

CpuVent2Parent　　CpuVent2Child