
最終更新日 : 2009年7月15日

FMPカーネルテストプログラムマニュアル

テストプログラムの概要

本テストプログラムには、大きく 種類のテストが含まれている。
テスト項目については、今後拡充していく予定である

コア動作テスト

を1コアで動作させ、ASP付属のテストプログラムと同等のテストプログラムを実行する

テストの詳細は各テストプログラムのコメントを参照のこと

コア動作テスト

2コアで を動作させ、 独自 の基本動作を確認する
テスト
テスト

テストプログラムのファイル構成

1コア動作テスト

test_task1.c/h/cfg

test_cpuexc[1-13].c/h/cfg

test_sem[1-2].c/h/cfg

test_tex[1-2].c/h/cfg

test_sysstat1.c/h/cfg

テストプログラムのファイル構成

2コア動作テスト

mig_tsk()テスト

- ・ test_mig_tsk1.c/h/cfg
- ・ test_mig_tsk2.c/h/cfg

mact_tsk()テスト

- ・ test_mact_tsk1.c/h/cfg
- ・ test_mact_tsk2.c/h/cfg
- ・ test_mact_tsk3.c/h/cfg

テスト用ライブラリプログラム

test_lib.c/h

テストプログラムの実行方法

- プロジェクトディレクトリの作成

Makefileはsample1付属のものをベースにする

コア数はテスト毎に異なるため、個々のテストに従う

- コンパイル対象のファイルのコピー

テスト用ライブラリ(test_lib.c/test_lib.h)

テスト毎のファイル(h/c/cfg)

- Makeファイルの編集

APPNAMEをテスト毎のファイル名とする

APPL_COBJS に test_lib.o を追加

- ビルドと実行

チェックポイントを全て通過すればテストをパスしたとみなす

の拡張内容

- ASP付属のファイルをマルチコア用に拡張
 - チェックポイントをコアごとに管理するように拡張

```
task(){  
    chech_point(3)  
}
```

2コア動作テスト : mig_tskテスト(1)

- 内容

2コアで動作させ, mig_tsk のパターン1と2をテストする

- ファイル

test_mig_tsk1.cfg / test_mig_tsk1.h / test_mig_tsk1.c

- テストパス条件

2コア共にチェックポイントを通過すればOK

- コア1 : 10個
- コア2 : 11個

2コア動作テスト : mig_tskテスト(2)

- 内容

2コアで動作させ, mig_tsk のパターン3をテストする

- ファイル

test_mig_tsk2.cfg / test_mig_tsk2.h / test_mig_tsk2.c

- テストパス条件

2コア共にチェックポイントを通過すればOK

- コア1 : 9個
- コア2 : 12個

2コア動作テスト：mact_tskテスト(1)

- 内容

2コアで動作させ, mact_tsk のパターン1のaをテストする

- ファイル

test_mact_tsk1.cfg / test_mact_tsk1.h / test_mact_tsk1.c

- テストパス条件

2コア共にチェックポイントを通過すればOK

- コア1 : 5個
- コア2 : 8個

2コア動作テスト : mact_tskテスト(2)

- 内容

2コアで動作させ, mact_tsk のパターン1のbをテストする

- ファイル

test_mact_tsk2.cfg / test_mact_tsk2.h / test_mact_tsk2.c

- テストパス条件

2コア共にチェックポイントを通過すればOK

- コア1 : 8個
- コア2 : なし

2コア動作テスト：mact_tskテスト(3)

- 内容

2コアで動作させ, mact_tsk のパターン1のbをテストする

- ファイル

test_mact_tsk3.cfg / test_mact_tsk3.h / test_mact_tsk3.c

- テストパス条件

2コア共にチェックポイントを通過すればOK

- コア1 : 13個
- コア2 : 5個

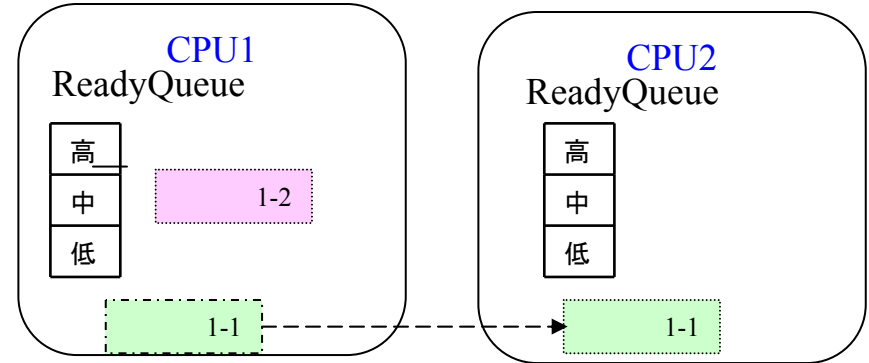
テスト内容詳細

mig_tskテスト(1)(2) テスト概要

マイグレートする対象タスクの状態により, パターン1,2,3に分類

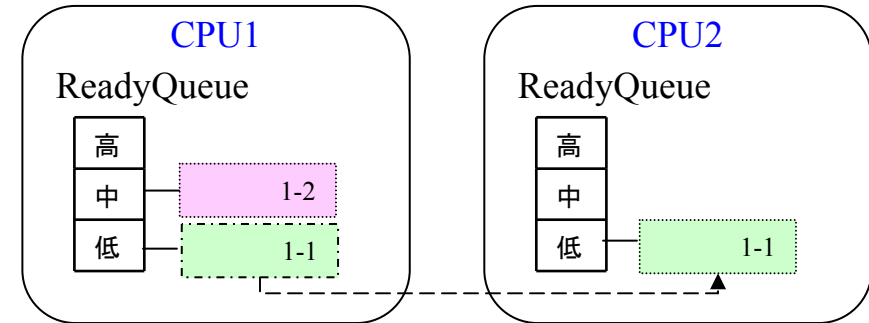
- パターン1

- 対象タスクが休止状態
 - ReadyQueueにつながない



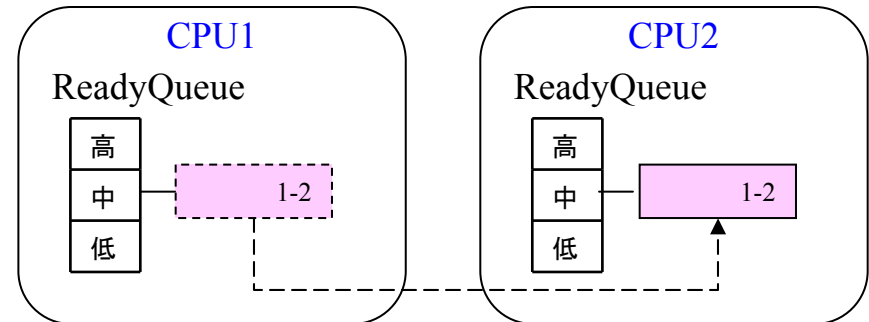
- パターン2

- 対象タスクが実行可能状態
 - ReadyQueueにつながない他をタスクを移動



- パターン3

- 対象タスクが実行状態
 - 自分自身が移動する



mig_tskテスト(1)(2) テストパターン

最高優先度になる	実行状態のタスクと 同優先度になる	実行状態のタスクより 低優先度になる	レディキューが空の所へ 移動する
----------	----------------------	-----------------------	---------------------

パターン1

mig_tskテスト(1)：詳細シーケンス

- パターン1と2をテスト

コア1 レディキュー					RUNタスク	check_point	テストパターン	サービスコール	コア2レディキュー					RUNタスク	check_point	サービスコール	
1-5	1-1	1-3															
	1-4	1-6															
	1-7	1-8	1-9														
					1-1	① ②	1-A 1-E	mig_tsk(1-5,2) mig_tsk(1-3,2) フラグ8待ち	1-5		1-3			1-3	①	フラグ8セット	同期1 フラグ8
					1-1	③ ④ ⑤	1-D 1-D	mig_tsk(1-4,2) mig_tsk(1-7,2) フラグ1セット			1-3 1-4 1-7			1-3		フラグ1待ち	同期2 フラグ1
					1-1			フラグ2待ち			1-3 1-4 1-7	1-5	1-3	② ③	act_tsk(1-5) slp_tsk		
													1-4 1-5 1-7	④ ⑤ ⑥	slp_tsk slp_tsk フラグ2セット		
																	同期3 フラグ2
					1-1	⑥	1-B	mig_tsk(1-6,2)					1-7			フラグ4待ち	
													1-6	⑦ ⑧	フラグ4セット slp_tsk		
					1-8	⑦ ⑧ ⑨ ⑩	1-C	slp_tsk フラグ10待ち mig_tsk(1-9,2) テスト終了					1-7 1-9	⑨ ⑩ ⑪	フラグ10セット slp_tsk テスト終了		

mig_tskテスト(2)：詳細シーケンス

・パターン3をテスト

2

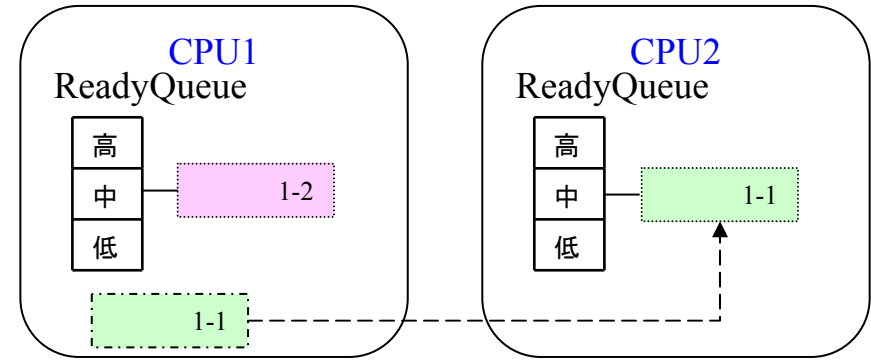
コア1 レディキュー			RUNタスク	check_point	テストパターン	サービスコール	コア2レディキュー		RUNタスク	check_point	サービスコール	
	1-1											
	1-4											
	1-7	1-9										
			1-1	①	1-I	mig_tsk(1-1,2)		1-1	1-1	①	act_tsk(2-2)	
										②	act_tsk(2-3)	
										③	slp_tsk	
										④	フラグ2待ち	
	1-4		1-4	②		フラグ2セット		2-2	2-2			
	1-7	1-9						2-3				
			1-4			フラグ10待ち		2-3	2-3	⑤	slp_tsk	同期1 フラグ2
										⑥	フラグ10セット	
			1-4	③	1-G	mig_tsk(1-4,2)		2-3	2-3		フラグ4待ち	同期2 フラグ10
			1-7	④		フラグ4セット		1-4				
	1-7	1-9										同期3 フラグ4
			1-7	⑤	1-H	mig_tsk(1-7,2)		2-3	2-3	⑦	slp_tsk	
			1-9	⑥		act_tsk(1-6)		1-4	1-4	⑧	slp_tsk()	
								1-7	1-7	⑨	フラグ1セット	
	1-6		1-6	⑦		フラグ1待ち						同期4 フラグ1
	1-9											
			1-6	⑧	1-F	mig_tsk(1-6,2)		1-6	1-6	⑩	フラグ8セット	
						フラグ8セット		1-7	1-7	⑪	slp_tsk	
			1-9	⑨		テスト終了				⑫	フラグ8待ち	
											テスト終了	

mact_tskテスト(1)(2)(3) : テスト概要

- 対象タスクの状態により分類

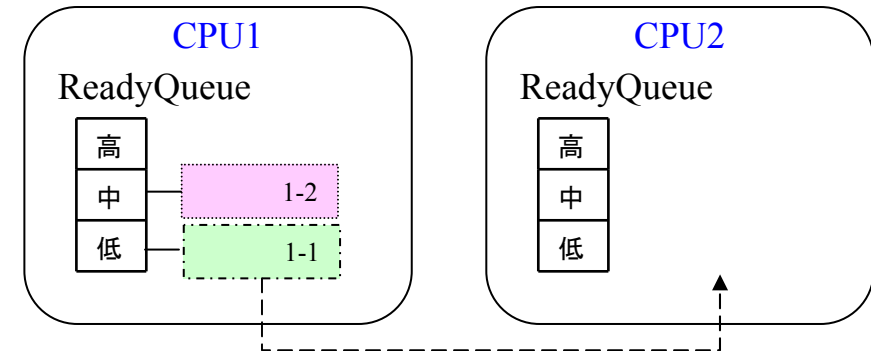
パターン1

- 対象タスクが休止状態
– ReadyQueueにつなぐ



パターン2

- 対象タスクが休止状態以外
– キューイング数の操作のみ



- 対象タスクの所属により分類

パターンa

- 自プロセッサ所属のタスクを他プロセッサへ

パターンb

- 他プロセッサ所属のタスクを自プロセッサへ

パターンc

- 他プロセッサ所属のタスクを他プロセッサへ

mact_tskテスト(1)(2)(3) : テストパターン

- パターン1
対象タスクが起動するReadyQueueの状態で、さらに場合分け
- パターン1, 2とパターンa,b,cを組み合わせ、テストパターンを抽出

対象タスクの状態		最高優先度	実行状態のタスクと 同優先度	実行状態のタスクより 低い優先度	レディキューが空
パターン1 休止状態 (レディキューへ つなぐ作業発生)	パターンa 自→他	2-A	2-B	2-C	2-D
	パターンb 他→自	2-E	2-F	2-G	
	パターンc 他→他	実施せず	実施せず	実施せず	実施せず
パターン2 休止状態以外 (レディキューへ つなぐ作業なし)	パターンa 自→他	2-H			
	パターンb 他→自	2-I			
	パターンc 他→他	実施せず	実施せず	実施せず	実施せず

mact_tskテスト(1)：詳細シーケンス

- 対象タスクが，休止状態，自プロセッサのタスクを他プロセッサへ

2

コア1 レディキュー		RUNタスク	check_point	テストパターン	サービスコール	コア2レディキュー		RUNタスク	check_point	サービスコール	
	1-4										
		1-4	①	2-D	mact_tsk(1-1,2) フラグ1待ち		1-1 2-2	1-1 2-2	① ② ③ ④	act_tsk(2-2) slp_tsk フラグ1セット	同期1 フラグ1
			② ③ ④ ⑤	2-A 2-B 2-C	mact_tsk(1-2,2) mact_tsk(1-5,2) mact_tsk(1-7,2) テスト終了		1-2 2-2 1-7	2-2 1-2 2-2 1-5 1-7	⑤ ⑥ ⑦ ⑧	フラグ2待ち フラグ2セット slp_tsk slp_tsk slp_tsk テスト終了	

mact_tskテスト(2)：詳細シーケンス

- 対象タスクが、休止状態、他プロセッサのタスクを自プロセッサへ

2

コア1 レディキュー		RUNタスク	check_point	テストパターン	サービスコール	コア2レディキュー		RUNタスク	check_point	サービスコール	
	1-4										
		1-4	①	2-E	mact_tsk(2-1,1)						
	2-1	2-1	②		フラグ1セット						
	1-4		③		slp_tsk						
		1-4			フラグ1待ち						
			④	2-F	mact_tsk(2-4)						
	1-4	2-4	⑤	2-G	mact_tsk(2-7)						
	2-7		⑥		slp_tsk						
		2-4	⑦		slp_tsk						
		2-7	⑧		テスト終了						

mact_tskテスト(3)：詳細シーケンス

- 対象タスクが, 起床待ち

2

コア1 レディキュー		RUNタスク	check_point	テストパターン	サービスコール	コア2レディキュー		RUNタスク	check_point	サービスコール	
	1-4										
		1-4	①		mact_tsk(1-5,2)			1-5	①	フラグ1セット slp_tsk	
		1-4	②	2-H	mact_tsk(1-5,2) フラグ1待ち		1-5				
											同期1 フラグ1
			③		wup_tsk(1-5) フラグ2待ち				② ③	1回目の実行終了 2回目起動 フラグ2セット slp_tsk	
											同期2 フラグ2
	2-1	1-4	④		wup_tsk(1-5)				④		
	1-4	2-1	⑤		mact_tsk(2-1,1)				⑤	テスト終了	
		1-4	⑥	2-I	slp_tsk						
		1-4	⑦		mact_tsk(2-1,1)						
		2-1	⑧		wup_tsk(2-1)						
		2-1	⑨		1回目終了						
		2-1	⑩		2回目起動						
		1-4	⑪		slp_tsk						
		2-1	⑫		wup_tsk(2-1)						
		2-1	⑬		slp_tsk						
					テスト終了						