
2010年12月3日

組込み総合技術展 (Embedded Technology 2010)

マルチプロセッサ対応

リアルタイムOSのテストスイート開発

株式会社デジタルクラフト
組込みシステム開発センター
金 スノヨブ

アジェンダ

1. 背景
2. コンソーシアム型研究組織
3. これまでの研究成果
4. 今後の展望

マルチプロセッサRTOSをテーマとした背景

大きく二つの理由により、組み込みシステムにおけるマルチプロセッサの利用が進んでいる

性能向上と消費電力削減の両立

- 低性能CPU複数個の消費電力 < 高性能CPU1個の性能
- 消極的なマルチプロセッサの利用
 - ソフトウェアエンジニアとしては高性能で低消費電力なプロセッサが使いたい

要件の異なるサブシステムの組み合わせ

- リアルタイム性と高機能性の両立
 - 携帯電話, カーナビ, NC工作機
- サブシステムの設計の容易化
- 積極的なマルチプロセッサの利用

産学連携によるオープンソースのマルチプロセッサ向けRTOSの開発が進んでいる

対応RTOS: TOPPERS新世代カーネル

μITRON仕様の現状

- μITRON仕様発表から10年以上が経過
- 組込みシステムの発展により, 新たな仕様とそれに伴う実装が要求されている

組込みシステムにおける要求の変化

- システム/ソフトウェアの大規模化・複雑化
- これまで以上に高い信頼性・安全性
- ソフトウェアポータビリティの向上
- 小さい消費エネルギーで高い性能

TOPPERSプロジェクトでは, これらの要求に対応する新しいカーネル仕様を策定



対応RTOS:TOPPERS新世代カーネル

- μ ITRON仕様をベースとして信頼性, 安全性, ソフトウェアポータビリティを向上させるための改良・拡張
- 改良した仕様は“TOPPERS 新世代カーネル統合仕様書”にまとめられている
- シングルプロセッサ対応RTOS: [ASPカーネル](#)  ASP Kernel
Advanced Standard Profile Kernel
- マルチプロセッサ対応RTOS : [FMPカーネル](#)  FMP Kernel
Flexible MultiProcessor Kernel
- いずれもTOPPERSプロジェクトからオープンソースとして公開

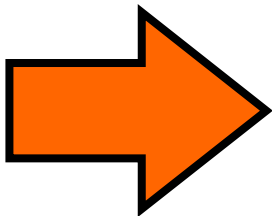
TOPPERS新世代カーネルのテストに関する問題点

包括的なテストは未実施

- RTOSのテストは規模が膨大で工数が多い
- 仕様検討/性能評価などの研究を中心としていた
- RTOSを実製品へ組み込む際、利用者側でテストを行っている

マルチプロセッサ対応RTOSのテストに課題がある

- マルチプロセッサ対応RTOSのテストに対する知見が蓄積されていない
- テスト開発/実施の工数がシングルプロセッサより肥大化する
- プロセッサ間の実行タイミングに依存したテストを考慮する必要がある



名古屋大学 組込みシステム研究センター(NCES)は
2009年度よりコンソーシアム型の研究組織を立ち上げ、
上記の問題を解決するためのテストスイートの研究開発を遂行

アジェンダ

1. 背景

2. コンソーシアム型研究組織

3. これまでの研究成果

4. 今後の展望

コンソーシアム型共同研究とは？

コンソーシアム型共同研究

- 大学がテーマを設定して、複数の企業に参加してもらい、大学の主導で研究を進める
- 2010年度は5社1団体が参加
- 成果物の知財は、部分毎に担当した企業が保有し、他の参加企業はそれを自由に使用できるものとする。さらに、一定期間後にオープンとする

マルチプロセッサRTOSに関する研究を実施中

- マルチプロセッサ向けRTOSの仕様策定・実装・評価環境・評価及び開発環境，検証手法に関する研究開発
- 対象RTOSとしてTOPPERS/FMPカーネルを使用

コンソーシアム型研究組織の参加企業と体制

TOPPERS新世代カーネルの利用や, マルチプロセッサ対応RTOSに関する知見を得たい複数の企業・団体が参加

参加企業(五十音順)

株式会社デジタルクラフト

株式会社東芝 セミコンダクター社

日本電気通信システム株式会社

富士ソフト株式会社

宮城県産業技術総合センター

ルネサス エレクトロニクス株式会社

体制

- ・ 名古屋大学へ常駐:5名
 - ・ 自社内での作業:1名
 - ・ オブザーバ:3社
- ⇒ 月1回のミーティングに参加

アジェンダ

1. 背景

2. コンソーシアム型研究組織

3. これまでの研究成果

- APIテスト概要
- テスト実施結果, 検出した不具合
- FMPにおけるAPIテスト
- テストプログラム生成ツールの開発
- 研究成果の公開予定

4. 今後の展望

研究内容

RTOSに対するテストの分類

- API(静的API含む)に着目したテスト
- 処理単位に着目したテスト
- SILに注目したテスト
- テストターゲット依存部単独でのテスト
- 共通部ロック関数テスト
- デッドロック回避テスト
- コンフィギュレータテスト
- ...

→ 全APIが仕様通りに動作することをテストする

検討及び実施中

マルチプロセッサ対応FMPカーネルの前段階として、
シングルプロセッサ対応ASPカーネルに対してテストを行った

これまでの主な研究成果

組込みマルチプロセッサ向けRTOSの検証手法確立と実践

- テストの全体像設計
- テストプロセス(APIテスト)の確立
- テストスイートパッケージ(TTSP)
 - TOPPERS/ASPカーネル APIテストスイート
 - TOPPERS/FMPカーネル APIテストスイート
 - テストプログラム生成ツール
 - コンフィグレーションツール



2009年度の成果については
2010年度末に一般公開予定

• 対外発表

- “組込みリアルタイムOSのAPIテストの実施”, ソフトウェアテストシンポジウム(JaSST)‘10東京, 2010.
- “マルチプロセッサ対応RTOSのテスト 開発”, “RTOSのテスト自動生成システムに関する一考察”, 組込み技術とネットワークに関するワークショップ(ETNET2010), 2010.
- “マルチプロセッサ対応のRTOSに関する共同研究”, 組込みシステム開発技術展(ESEC2010), 2010.
- “コンソーシアム型研究によるTOPPERS新世代カーネル向けテストスイート開発”, TOPPERSカンファレンス2010, 2010.
- “マルチプロセッサ対応RTOSを対象としたテストシナリオ記述法とテストプログラム生成ツール”, 第18回組込みシステム研究発表会, 2010.
- “コンソーシアム型研究によるRTOSのテストスイート開発”, “組込みリアルタイムOSを対象としたテストプログラム生成ツールの開発”, 組込みシステム技術に関するサマーワークショップ12, 2010

APIテスト概要

TOPPERS新世代カーネル統合仕様書に基づいて
API発行前後のシステム状態の変化を確認する

wup_tskの仕様抜粋

wup_tsk タスクの起床

:

【機能】

tskidで指定したタスク(対象タスク)を起床する。
対象タスクが起床待ち状態である場合には、
対象タスクが待ち解除される。



仕様の振舞いを確認するテストを実施する

- ・ 自タスク以外のタスクを指定して呼び出す。
- ・ 起床待ち状態のタスクを指定する。
- ・ 対象タスクの優先度が、実行状態のタスクより高い場合、対象タスクが実行状態になること。

前状態



(実行中)
優先度:中



(起床待ち)
優先度:高

処理



wup_tsk



後状態

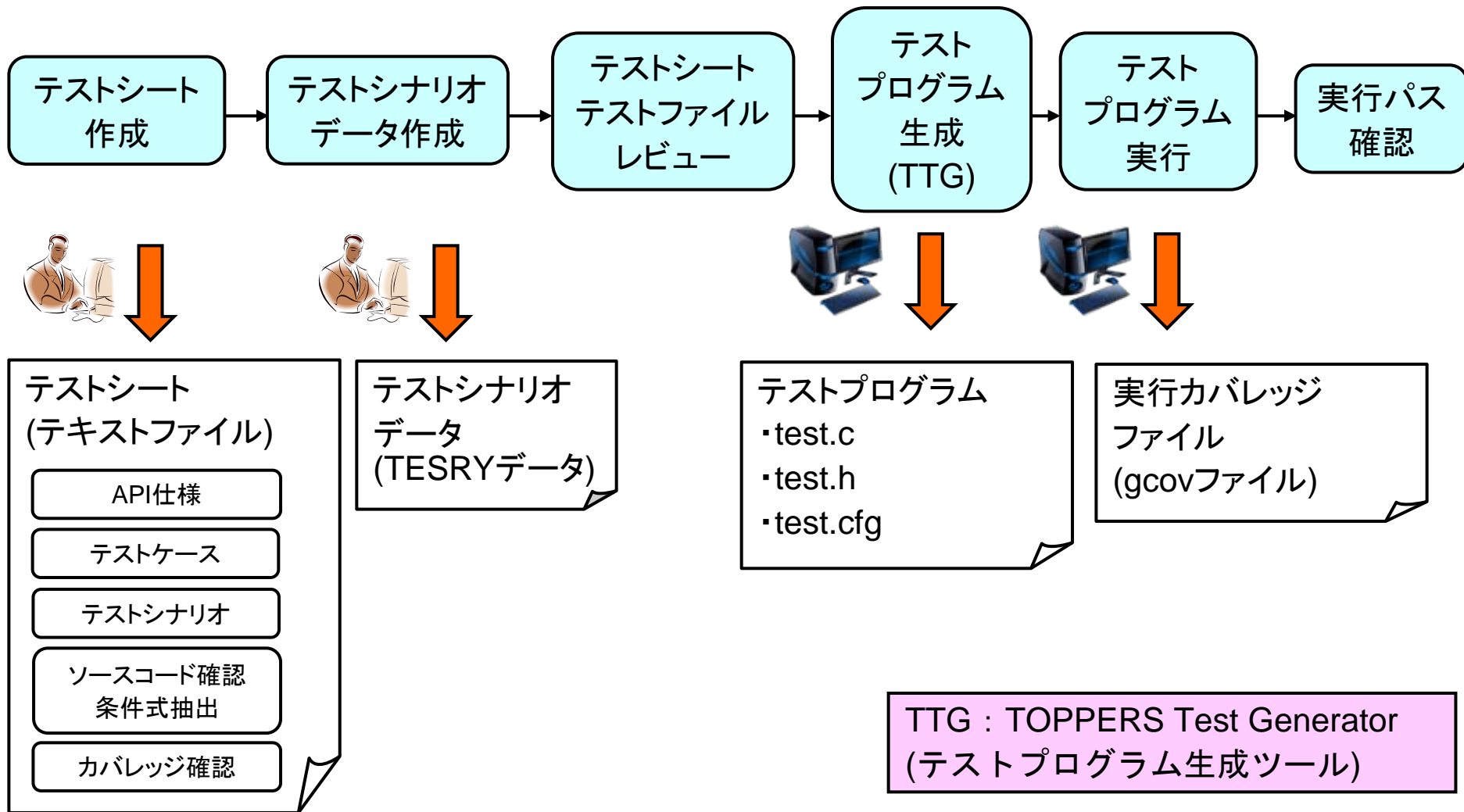


(実行可能)



(実行中)

APIテスト開発フロー



※ カバレッジ：条件分岐網羅

APIテスト実施結果(ASPカーネル)

機能毎のテストプログラム数一覧

機能名	テストケース数
タスク管理機能(task_manage)	181
タスク付属同期機能(task_sync)	205
タスク例外処理機能(task_except)	81
データキュー機能(dataqueue)	280
イベントフラグ機能(eventflag)	145
メールボックス機能(mailbox)	98
固定長メモリプール機能(mempfix)	97
優先度データキュー機能(pridataq)	263
セマフォ機能(semaphore)	106
周期ハンドラ機能(cyclic)	20
アラームハンドラ機能(alarm)	39
システム状態管理機能(sys_manage)	76
割り込み管理機能(interrupt)	22
CPU例外管理機能(exception)	9
タスクの状態参照機能(task_refer)	39
システム時刻管理機能(time_manage)	7
静的API	115
合計	1,783

全テストプログラム実行後のカバレッジ

LCOV - code coverage report

Current view: [directory - kernel](#) Found: 2005 Hit: 2005 Coverage: 100.0 %
 Test: [asp_total.info](#) Lines: 2005 Hit: 2005 Coverage: 100.0 %
 Date: 2009-12-07 Functions: 172 Hit: 172 Coverage: 100.0 %

Filename	Line Coverage	Functions
alarm.c	100.0 % 79 / 79	100.0 % 7 / 7
cyclic.c	100.0 % 59 / 59	100.0 % 6 / 6
dataqueue.c	100.0 % 240 / 240	100.0 % 18 / 18
eventflag.c	100.0 % 154 / 154	100.0 % 10 / 10
exception.c	100.0 % 6 / 6	100.0 % 2 / 2
interrupt.c	100.0 % 61 / 61	100.0 % 5 / 5
mailbox.c	100.0 % 114 / 114	100.0 % 8 / 8
mempfix.c	100.0 % 117 / 117	100.0 % 8 / 8
pridataq.c	100.0 % 225 / 225	100.0 % 14 / 14
semaphore.c	100.0 % 109 / 109	100.0 % 8 / 8
startup.c	100.0 % 14 / 14	100.0 % 3 / 3
sys_manage.c	100.0 % 101 / 101	100.0 % 15 / 15
task.c	100.0 % 121 / 121	100.0 % 14 / 14
task_except.c	100.0 % 68 / 68	100.0 % 6 / 6
task_manage.c	100.0 % 116 / 116	100.0 % 8 / 8
task_refer.c	100.0 % 57 / 57	100.0 % 1 / 1
task_sync.c	100.0 % 162 / 162	100.0 % 10 / 10
time_event.c	100.0 % 69 / 69	100.0 % 8 / 8
time_event.h	100.0 % 14 / 14	100.0 % 3 / 3
time_manage.c	100.0 % 21 / 21	100.0 % 2 / 2
wait.c	100.0 % 68 / 68	100.0 % 10 / 10
wait.h	100.0 % 30 / 30	100.0 % 6 / 6

ラインカバレッジ 100% !

APIテスト実施結果(FMPカーネル)

FMPで拡張したテストプログラム数一覧

機能名	テストケース数
タスク管理機能(task_manage)	604
タスク付属同期機能(task_sync)	279
タスク例外処理機能(task_except)	123
データキュー機能(dataqueue)	370
イベントフラグ機能(eventflag)	116
メールボックス機能(mailbox)	66
固定長メモリプール機能(mempfix)	66
優先度データキュー機能(pridataq)	321
セマフォ機能(semaphore)	91
周期ハンドラ機能(cyclic)	56
アラームハンドラ機能(alarm)	142
システム状態管理機能(sys_manage)	135
割り込み管理機能(interrupt)	13
CPU例外管理機能(exception)	10
タスクの状態参照機能(task_refer)	88
システム時刻管理機能(time_manage)	3
静的API	47
スピンロック(spin_lock)	48
合計	2,578

ASPのテストケースの拡張

- ASPのテストケースをベースに、複数のプロセッサに跨いだテストケースを拡張
- マルチプロセッサの中の1つのプロセッサに対しては、ASPカーネルと同等のテストケースを行うため、合計4,361件のテストを実施可能

カバレッジの測定は実施中

- FMPカーネルは、APIテストだけでは網羅できないパスが多く存在する(タイミングやロック状態に依存)
- 合計4,361件のテストケースについてカバレッジを確認するので時間が掛かっている

APIテスト開発で検出した不具合

	2009年度		2010年度	
	誤記, 要望等	不具合	誤記, 要望等	不具合
統合仕様書	7	9	15	1
コンフィギュレータ	0	2	0	1
ASPカーネル	3	9	2	2
FMPカーネル	0	11	1	26
合計	10	31	18	30

仕様上の不具合の例

- 曖昧な用語定義/エラー条件の整備
- ソース上考慮されている振舞いが、仕様に記載されていない

仕様, ソースの両面から包括的にAPIの振舞いをテストした

ソース上の不具合の例

- 統合仕様の記載と異なる実装が存在する
- 不要な条件分岐が存在する
- ターゲット依存部の実装不具合(ポーティングガイドに非準拠)

TOPPERS/FMPにおけるAPIテスト

APIテスト時に考慮すべきASPとFMPの仕様の違い

① APIのマルチプロセッサ拡張

- ASPのAPIの多くはFMPでは異なるプロセッサに割り付けられたカーネルオブジェクトの操作が可能のように拡張されている

② タスクの状態の追加

- ASPには無い「強制待ち状態[実行継続中]」という過渡的な状態を考慮する

③ 新規APIの追加

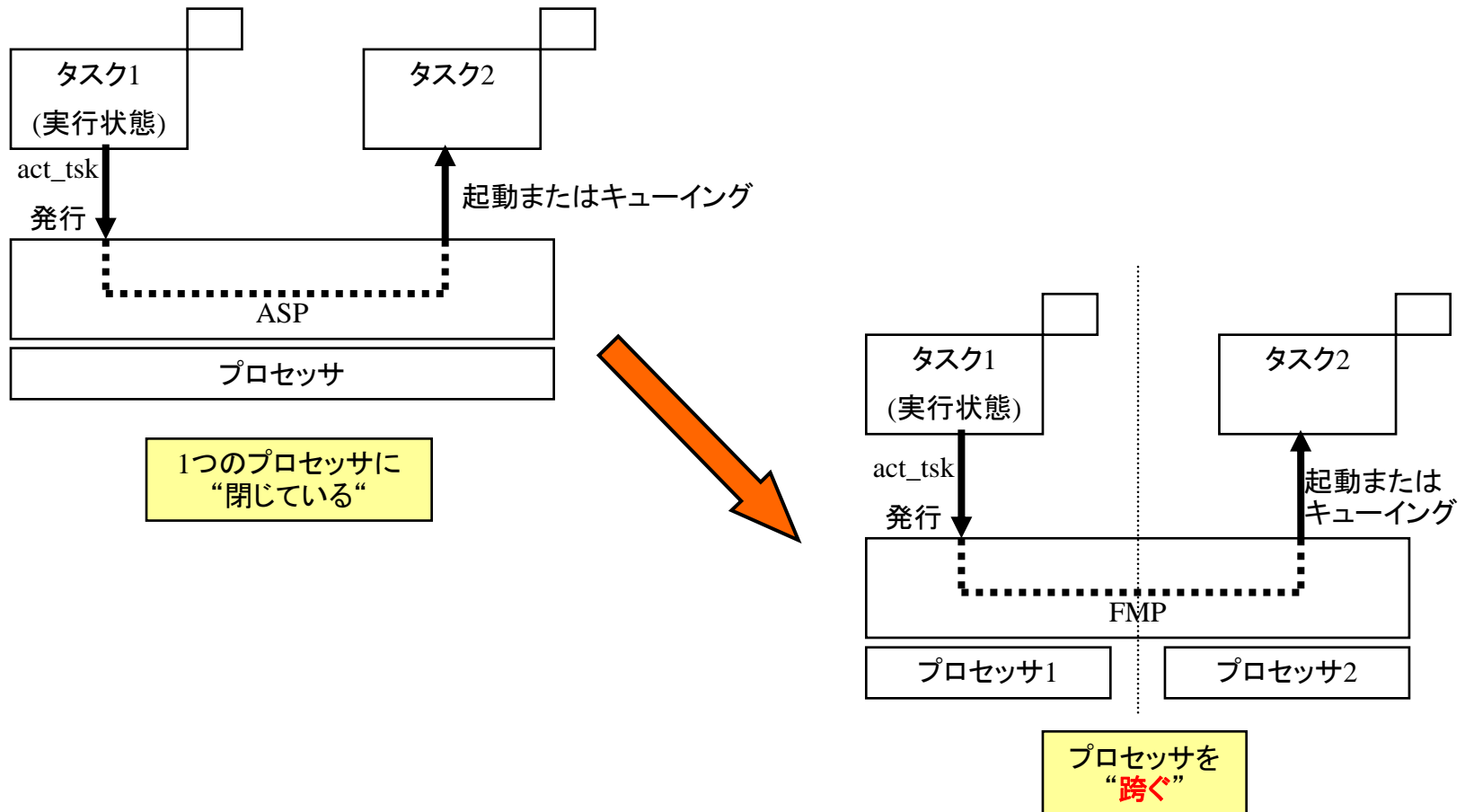
- 5つの機能に対して17個のAPIが新規に追加されている

④ クラス

- 複数のプロセッサを管理するカーネルオブジェクトの集合の定義

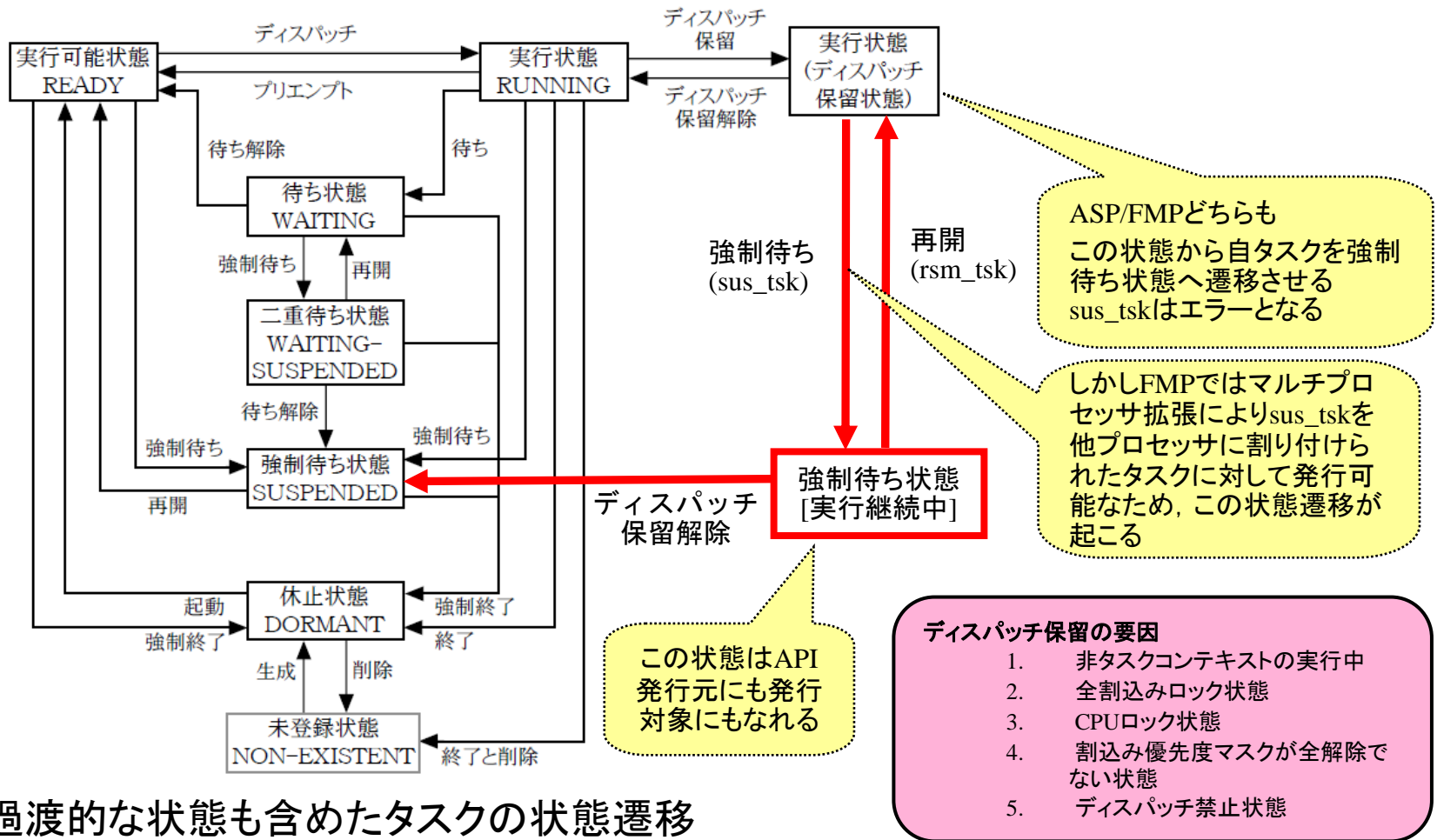
① APIのマルチプロセッサ拡張

ASPのAPIの多くはFMPでは異なるプロセッサに割り付けられたオブジェクトの操作が可能ないように拡張されている



② タスクの状態の追加

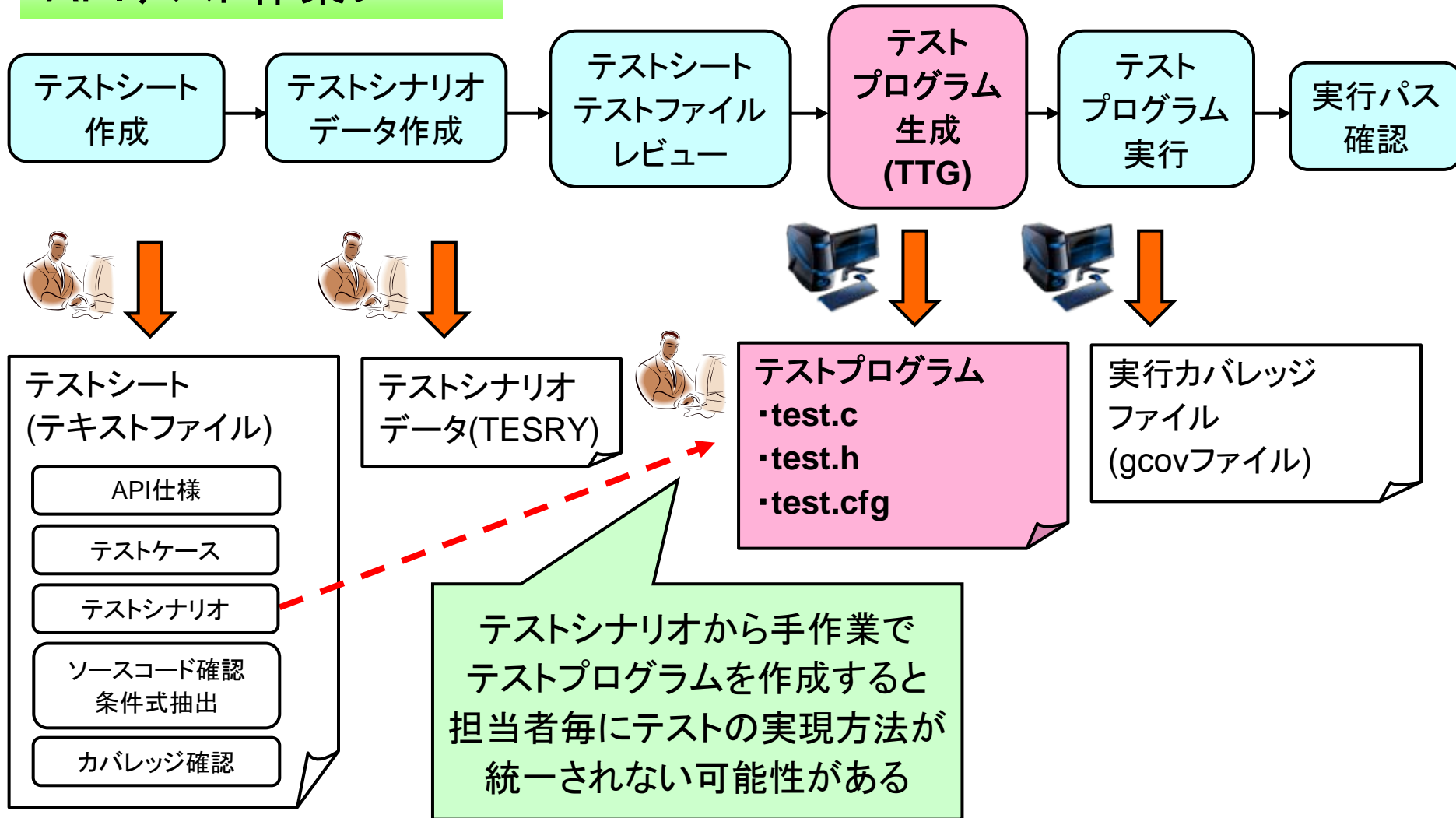
ASPには無い「強制待ち状態[実行継続中]」という過渡的な状態を考慮する



過渡的な状態も含めたタスクの状態遷移

テストプログラム生成ツールの開発

APIテスト作業フロー



テストプログラム生成ツールの必要性

担当者A作成

```
優先度 : TEST_TASK < TASK1 < TASK2  
TEST_TASK ()  
{  
    act_tsk (TASK2);  
    ref_tsk (TASK2);  
    act_tsk (TASK1);  
}  
TASK2 ()  
{  
    slp_tsk ();  
}  
TASK1 ()  
{  
    ref_tsk (TASK1); 前状態実現/確認完了  
    wup_tsk (TASK2);  
}
```

担当者B作成

```
優先度 : TASK1 < TASK2 < TEST_TASK  
TEST_TASK ()  
{  
    act_tsk (TASK2);  
    act_tsk (TASK1);  
    slp_tsk ();  
}  
TASK2 ()  
{  
    slp_tsk ();  
}  
TASK1 ()  
{  
    ref_tsk (TASK1); 前状態実現/確認完了  
    ref_tsk (TASK2); 前状態実現/確認完了  
    wup_tsk (TASK2);  
}
```

<実現する前状態>

[TASK1]

優先度: 中

実行状態

[TASK2]

優先度: 高

起床待ち状態

act_tsk: タスク起動

ref_tsk: タスク状態参照

slp_tsk: タスク起床待ち

前状態の実現/確認
方法が異なる

どちらも正しいテストプログラムではあるが、
正しいことを確認するレビュー時の可読性や
修正時の保守性が著しく低下する

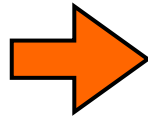
テストシナリオデータ(TESTRYデータ)の構成

階層型データ形式言語であるYAML形式を用いて
全カーネルオブジェクトの属性/状態の記述方法を定めた
(TESTRY記法で記述したデータファイルを**TESTRYデータ**と呼ぶ)

(例) wup_tskのテストケース

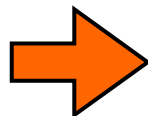
前状態

優先度中のTASK1が実行状態
優先度高のTASK2が起床待ち状態



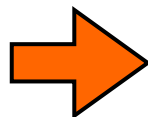
処理

TASK1がwup_tsk(TASK2)を発行する



後状態

TASK1が実行可能状態となる
TASK2が実行状態となる



```
pre_condition:
  TASK1:
    type      : TASK
    tskpri    : MID
    tskstat   : running
  TASK2:
    type      : TASK
    tskpri    : HIGH
    tskstat   : waiting
    wobjid    : SLEEP

do:
  - id       : TASK1
    syscall  : wup_tsk(TASK2)

post_condition:
  TASK1:
    tskstat  : ready
  TASK2:
    tskstat  : running
```

※前状態から変化のないパラメータは省略可能

テストプログラム生成ツール(TTG)の主な機能

- ① 形式化したテストシナリオ記法からのテストプログラム生成
- ② マルチプロセッサへの同期処理対応
- ③ マルチプロセッサへのバリエーション対応
- ④ 複数のテストシナリオ統合
- ⑤ 選択したRTOS に対するテストプログラム生成
- ⑥ テストシナリオの正当性チェック

①形式化したテストシナリオ記法からのテストプログラム生成

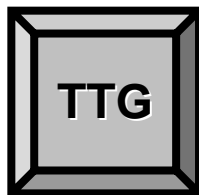
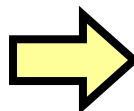
形式化したテストシナリオ

前状態
＜API発行前のシステム状態＞

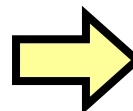
処理
＜API発行処理＞

後状態
＜API発行後のシステム状態＞

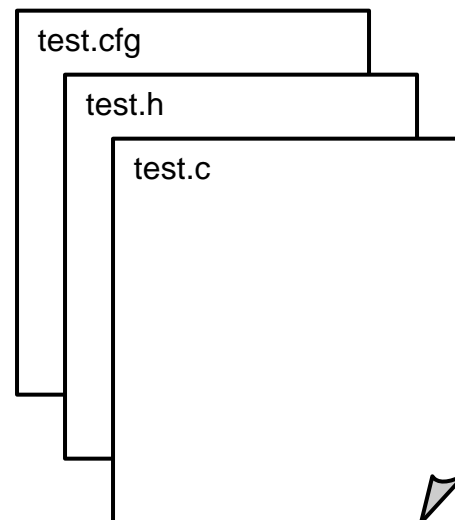
入力



出力



テストシナリオを
実現するテストプログラム

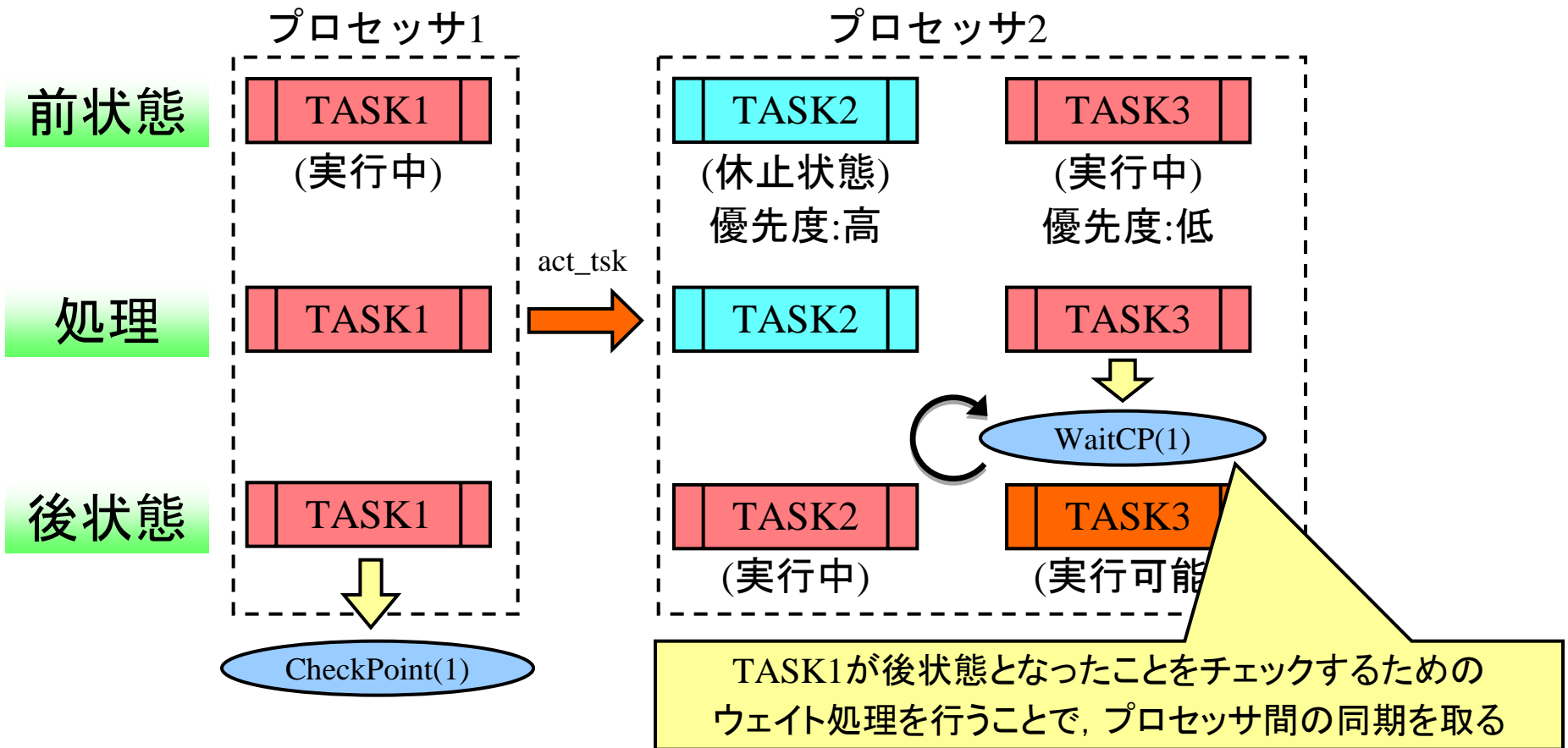


TESRY記法

(TEst Scenario for Rtos by Yaml)

- ・形式化した記法を用いたテストシナリオからテストプログラムを生成することで、担当者毎のばらつきが無くなる
- ・開発、管理対象が形式化したテストシナリオになることで保守性、可読性が向上する

②マルチプロセッサへの同期処理対応



マルチプロセッサ環境で必要な同期処理を整理し、
必要な同期用関数を開発した
→ 適切に同期を行うテストプログラムを生成する

③マルチプロセッサへのバリエーション対応

事前に定義したターゲットのバリエーションに合わせて、TTGが自動的に取捨選択を行う

FMP_alarm_msta_alm_F_h_1_1:

variation:
timer_arch: local

pre_condition:
TASK1:
type : TASK
tskstat: running
prcid : 1

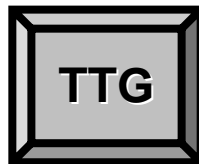
ALM1:
type : ALARM
almstat: TALM_STP
hdlstat: STP
prcid : 1

do:
id : TASK1
syscall: msta_alm(ALM1, 3, 1)
ercd : E_OK

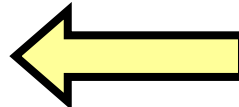
post_condition:

…略

テストケースを
構成する情報
(TESRYデータ)



テスト対象の
ターゲット情報



実行可能な
テストプログラム

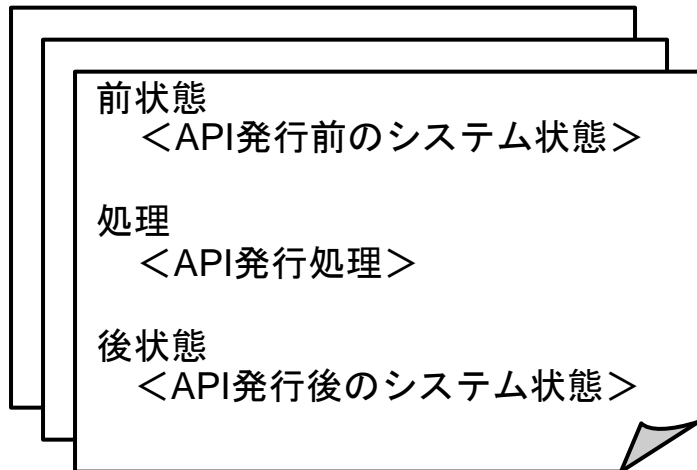
設定ファイル

- ・プロセッサ数: 3
- ・ロック方式: グローバル
- ・スピンロック数: 2
- ・タイマー方式: ローカル
- ・ターゲット依存関数:
 - ・時間操作: あり
 - ・割り込み発生: なし
 - ・CPU例外発生: なし

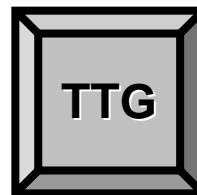
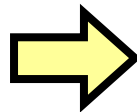
対象のターゲットでテスト実行可能な
テストケースのみをテストプログラムとして出力する

④複数のテストシナリオ統合

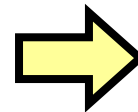
複数のテストシナリオ



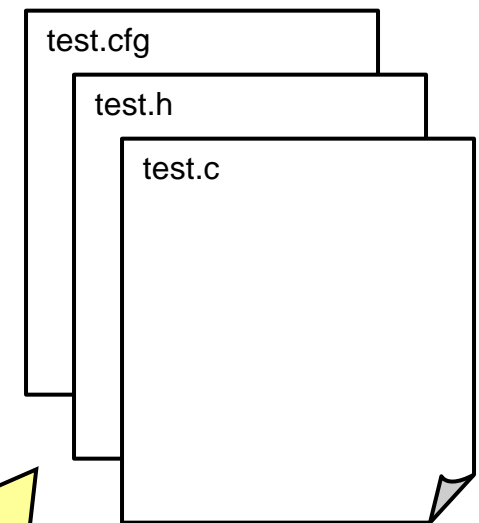
入力



出力



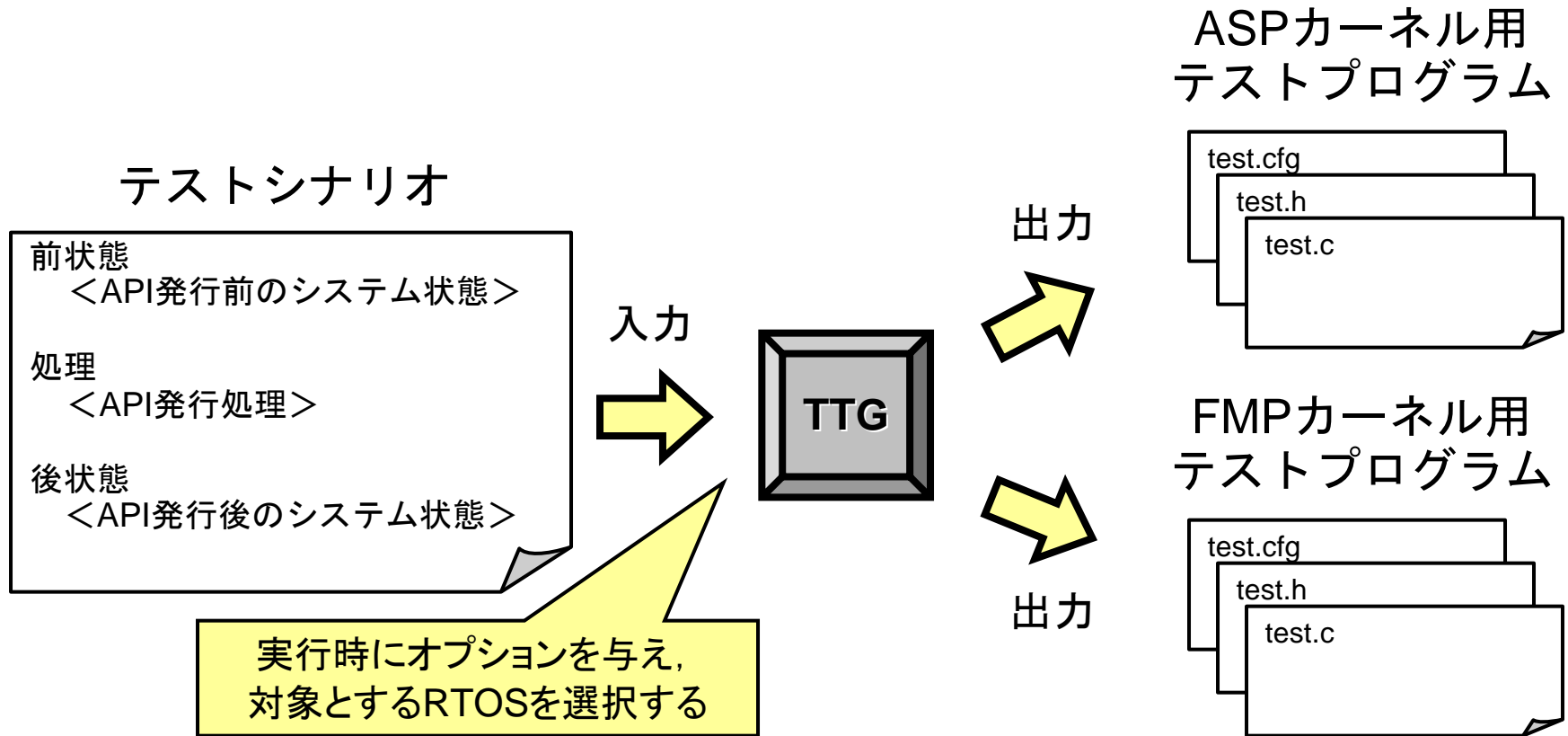
1つのテストプログラム



複数のテストシナリオ間で、タスクが使用するスタック領域を共有させメモリ使用量を抑える

テストケース毎に別々に開発を行い、
テスト実行時に1つのテストプログラムに
統合することを可能とした

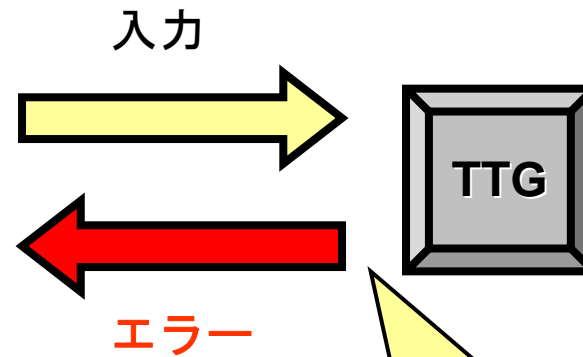
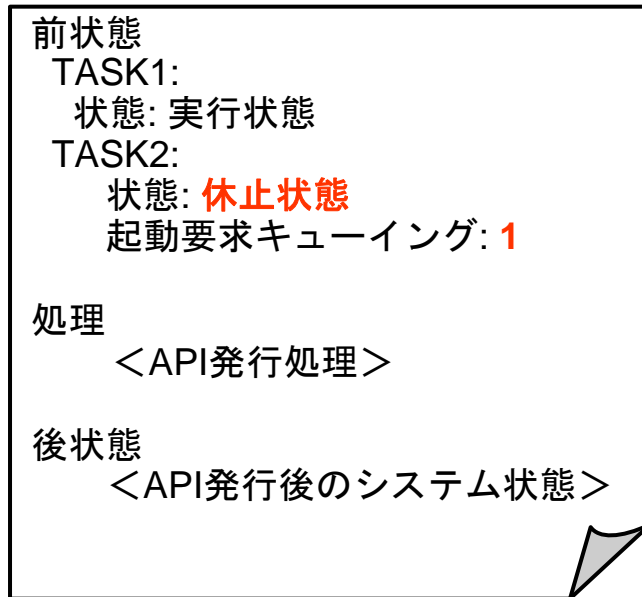
⑤選択したRTOS に対するテストプログラム生成



TTG実行時に指定したRTOSに対応した
テストプログラムを生成することで、
テストシナリオの流用を可能とした

⑥テストシナリオの正当性チェック

不正なテストシナリオ



テストシナリオ読み込み時に、
正当性をチェックして問題があれば
エラー原因を表示して終了する

テストシナリオに誤記や矛盾が存在しないかをチェックし、
問題のあるテストケースを検出可能とした

研究成果の公開予定

TTSP(TOPPERS Test Suite Package)

- ASP/FMPカーネル APIテストスイート
 - テストシート, TESRYデータ等
- テストプログラム生成ツール
- コンフィギュレーションツール
 - 環境設定ファイル
 - TTGを用いた実行モジュール生成ツール

2009年度の成果については2010年度末に
TOPPERSプロジェクトから一般公開予定

アジェンダ

1. 背景
2. コンソーシアム型研究組織
3. これまでの研究成果
4. 今後の展望

今後の計画と展望

検討及び実施予定のテスト項目

[A] APIテスト

[B] 処理単位テスト

[C] SILテスト

[G] ターゲット依存部テスト

[H] 共通部ロック関数テスト

[T] デッドロック回避テスト

[J] コンフィギュレータテスト

...



観点及び検証手法についても
改めて検討し、マルチプロセッサ
RTOSに対するテストプロセスの
確立を目指す

2010年度の実施中のテスト開発

処理単位に着目したテスト

- カーネルに定義された各処理単位において、規定した条件を満たすと、規定した振舞いをすることをテストする

SILに注目したテスト

- システムインタフェースレイヤが提供する、全割込みロック状態の制御、スピンロック、微小時間待ち、エンディアンの取得、メモリ空間アクセス(I/O空間アクセス)、プロセッサIDの参照の各機能をテストする

タイミング依存テスト対応

- タイミングに依存して容易にはテストプログラムで実現できないテストを実現する
 - 命令セットシミュレータ(ISS)改造
 - 割込み禁止区間の長さ測定、多重割込みの正常動作確認などのテストが可能になる

まとめ

- 名古屋大学を中心にマルチプロセッサ対応RTOSに関するコンソーシアム型研究組織を立ち上げた
- TOPPERS/ASP, TOPPERS/FMPに対して包括的なテストケース(4,361件)を開発し, 61件の不具合を検出した
- これからはAPIテスト以外のテストを実施する予定である
 - 2010年度は処理単位に着目したテストとSILに着目したテストを実施

NCESより

今後, マルチプロセッサ向けRTOSの共同研究を継続するにあたり, 参加して頂ける企業/団体を募集しています

mprtos-conso@nces.is.nagoya-u.ac.jp

<http://www.nces.is.nagoya-u.ac.jp/>

