

TwinOSのバージョンアップの検討

岡山大学 工学部 情報工学科

牛尾 裕

研究背景

<OSのバージョンアップの重要性>

- (1) 安全性/安定性の向上
- (2) 近年のハードウェアやソフトウェアに対応

新機能の実現には独自コードの追加が必要
例: TwinOSはLinuxに独自コードを追加

<TwinOSのバージョンアップの問題点>

- (1) 古いLinuxの改造に基づいている
- (2) 機能毎にモジュール化されていない
- (3) ソースコードが整理されていない

バージョンアップは
困難

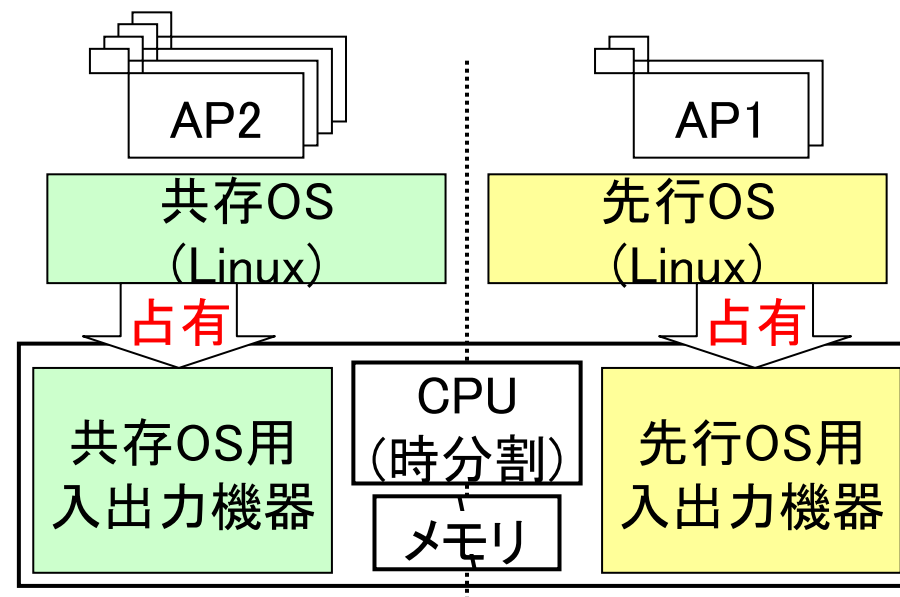
バージョンアップを容易にする独自コード追加手法の明確化
(TwinOSを事例に検討)

TwinOS

1台の計算機上で、2つのLinuxが同時走行する方式

<特徴>

- (1) 両OSともLinux 2.4を改造
- (2) 各OSで別々の変更



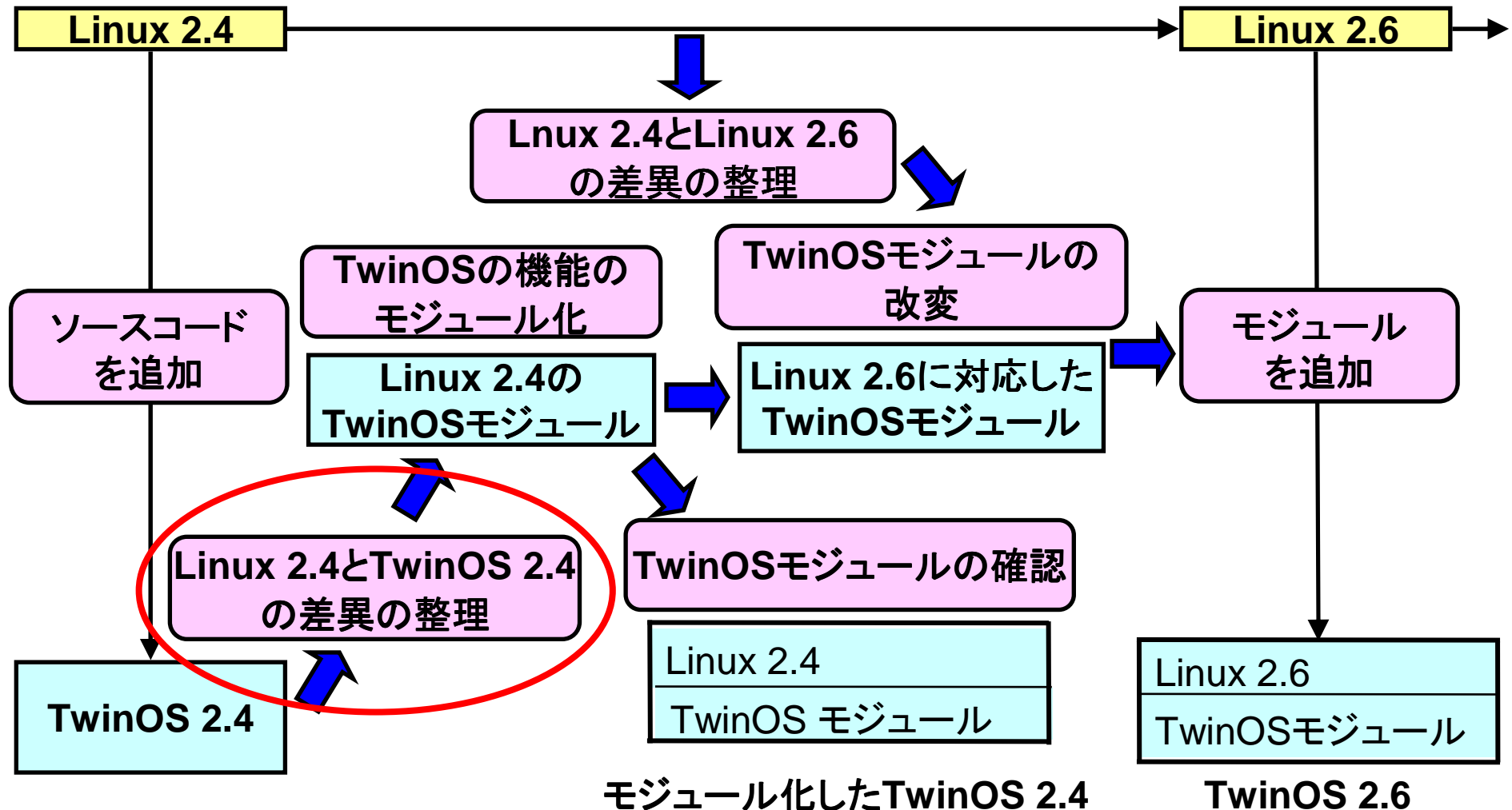
TwinOSをLinux 2.6に対応(TwinOSのバージョンアップ)

<要求>

- (1) LinuxとTwinOSの差異を機能毎に分割
- (2) Linuxのバージョン間差異を明確化
- (3) バージョンに依存しないTwinOSモジュールの作成

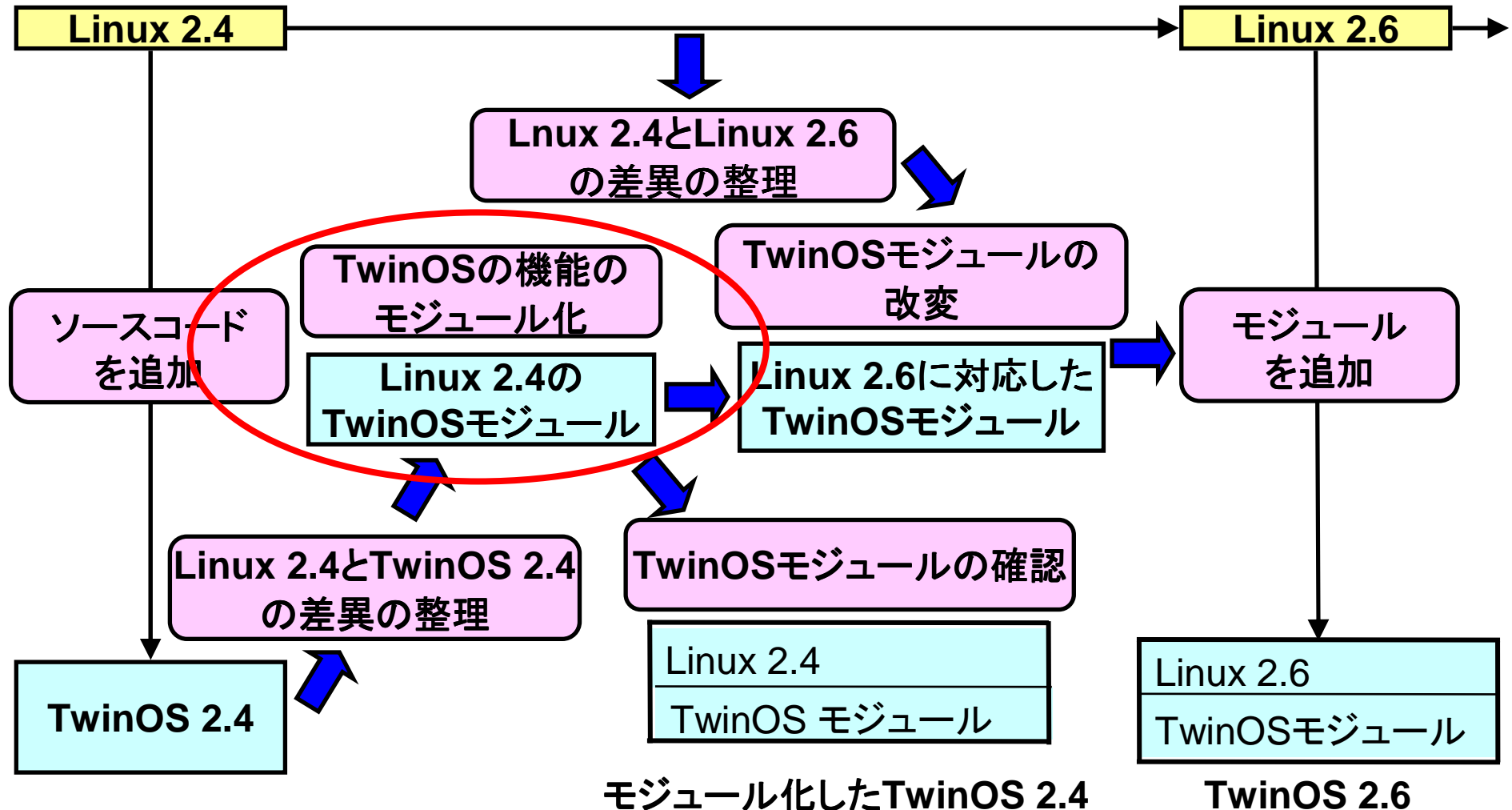
バージョンアップ方式

(1) Linux 2.4 とTwinOS 2.4 の差異の整理



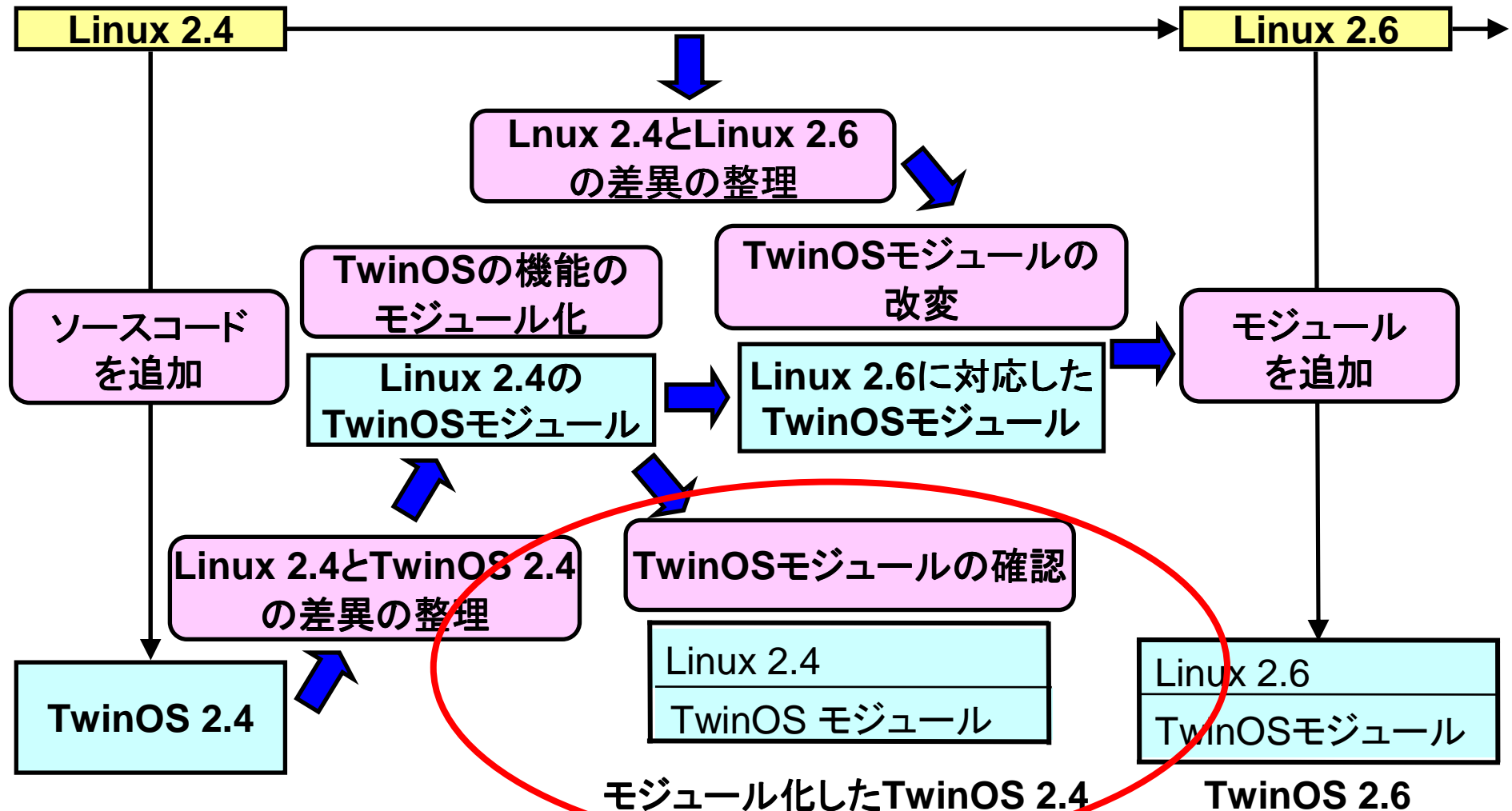
バージョンアップ方式

(2) TwinOSの機能のモジュール化(TwinOSモジュール作成)



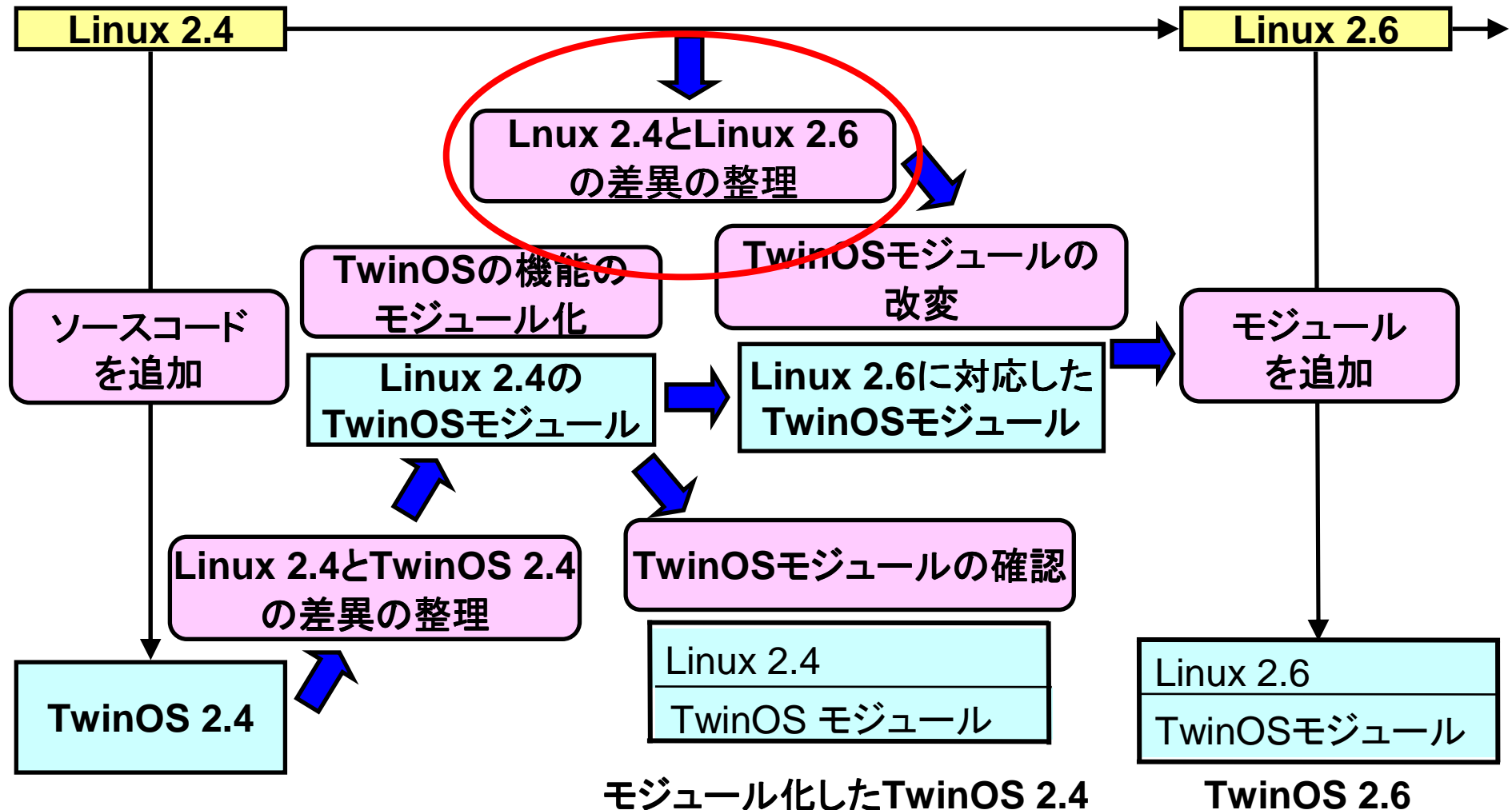
バージョンアップ方式

(3) TwinOSモジュールの確認



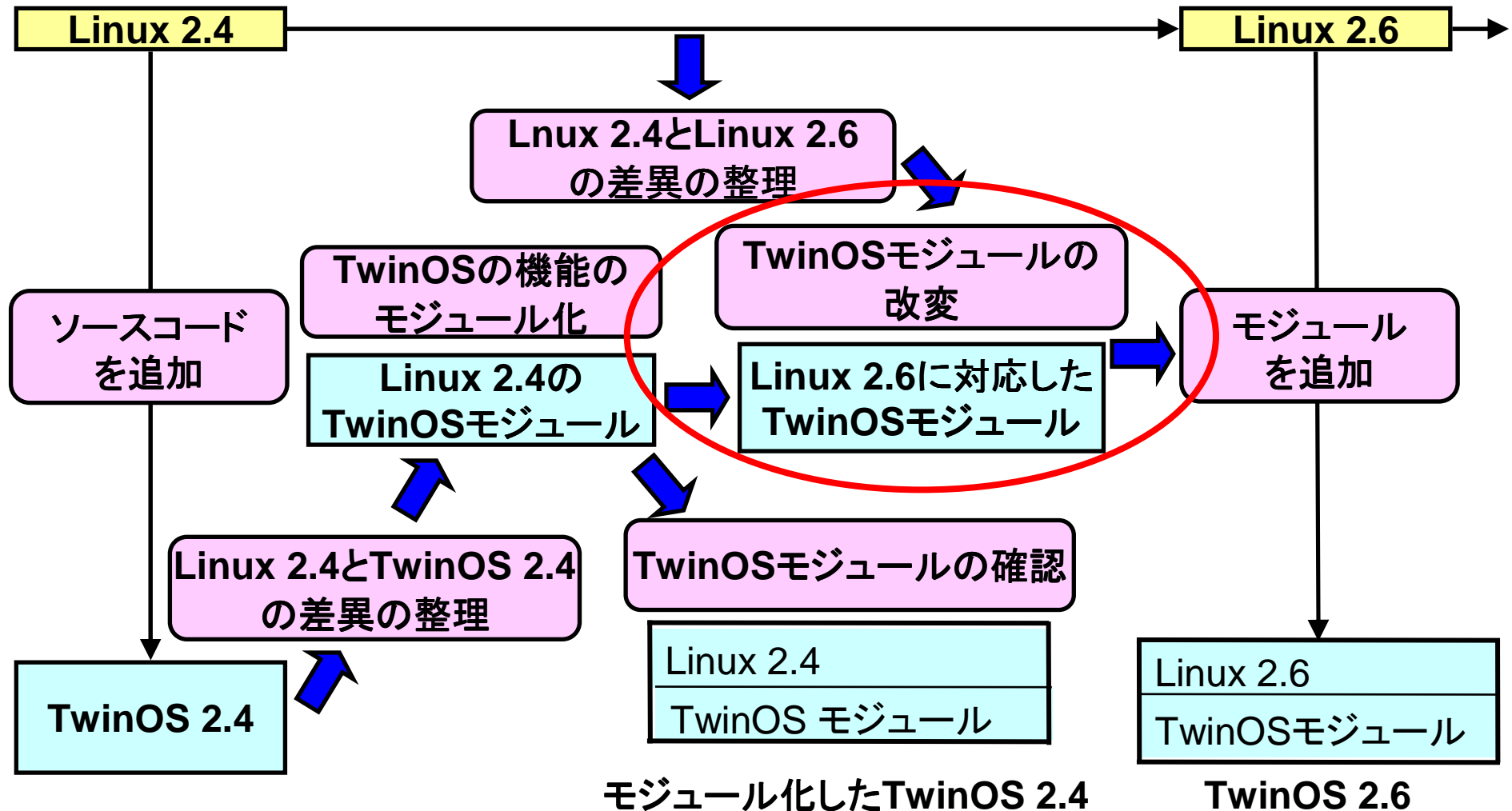
バージョンアップ方式

(4) Linux 2.4とLinux 2.6の差異の整理



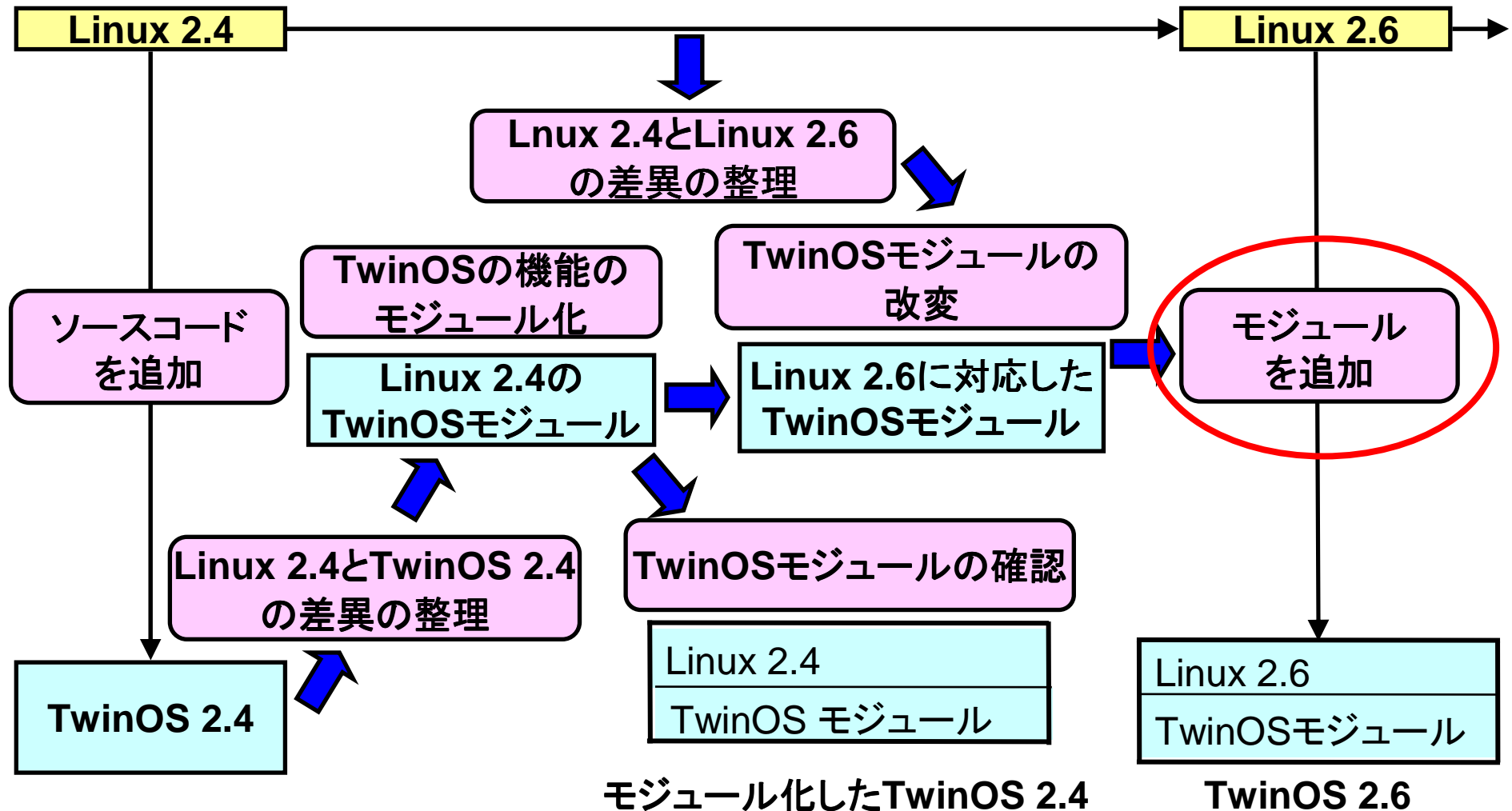
バージョンアップ方式

(5) TwinOSモジュールの改変



バージョンアップ方式

(6) TwinOS 2.6の作成



TwinOSのバージョンアップ方式

(1) Linux 2.4とTwinOS 2.4の差異の整理

(2) TwinOSの機能のモジュール化(TwinOSモジュール作成)

(3) TwinOSモジュールの確認

(4) Linux 2.4とLinux 2.6の差異の整理

(5) TwinOSモジュールの改変

(6) TwinOS 2.6の作成

TwinOSの機能のモジュール化

TwinOSを機能毎に5つのモジュールに分割

(1) メモリ分割機能

物理メモリを若番と老番で分割する機能

(2) 共存OSメモリ展開機能

共存OSを物理メモリの若番に展開する機能

(3) 共存OS起動機能

共存OSを起動し, 共存状態に移行する機能

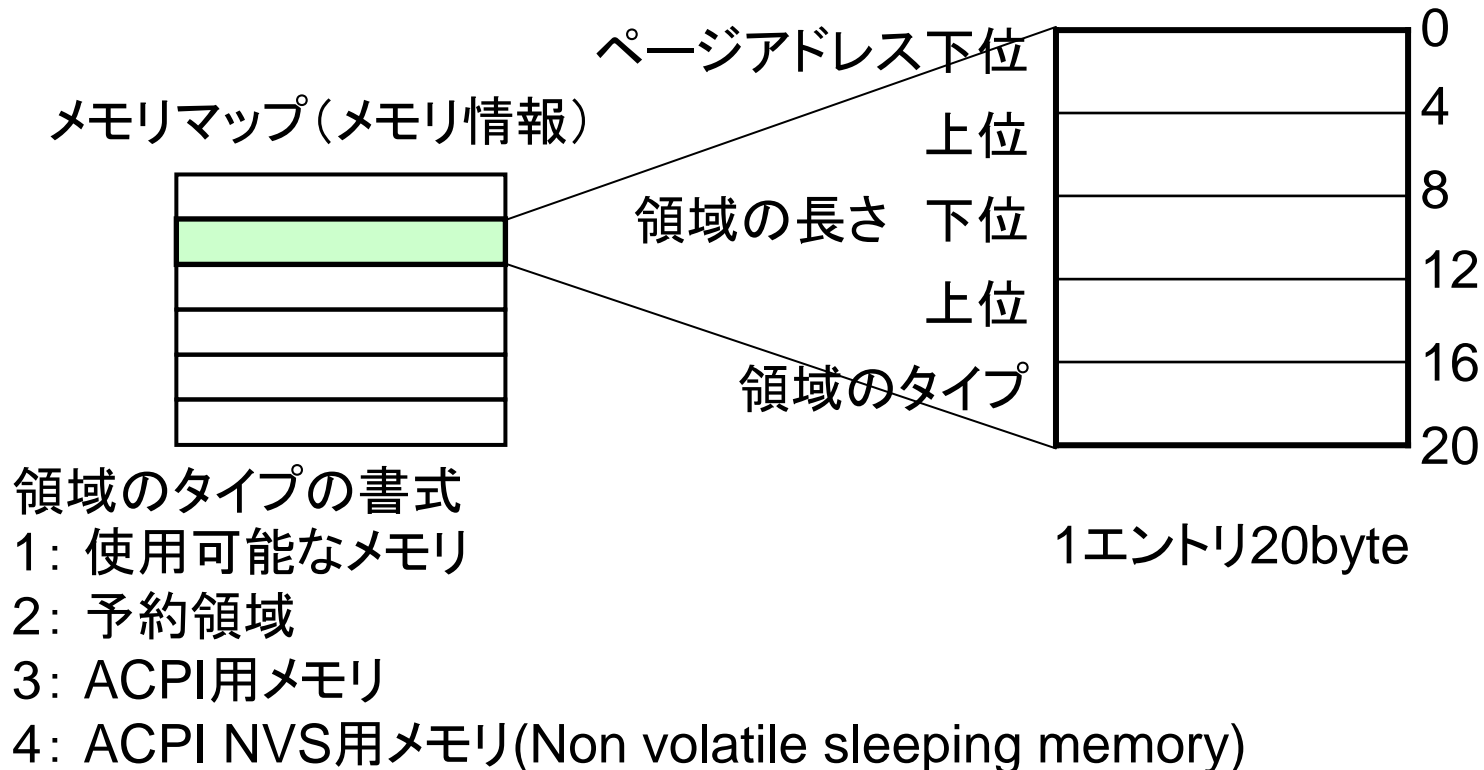
(4) 入出力機器占有機能

各OSに入出力機器を割り当てる機能

(5) CPU時分割機能

OS毎にCPUを時分割で割り当てる機能

メモリ分割機能のモジュール化



Linuxではメモリマップを取得することによりメモリ情報を得る

➡ TwinOSはメモリマップを書き換えることによりメモリを分割

TwinOSモジュールの改変

メモリ分割機能のTwinOSモジュールをLinux 2.6に対応

<TwinOS 2.4の問題点>

- (1) 分割するメモリアドレスをマジックナンバーで指定
- (2) メモリマップの分割方法が冗長

<Linuxバージョン間の違い>

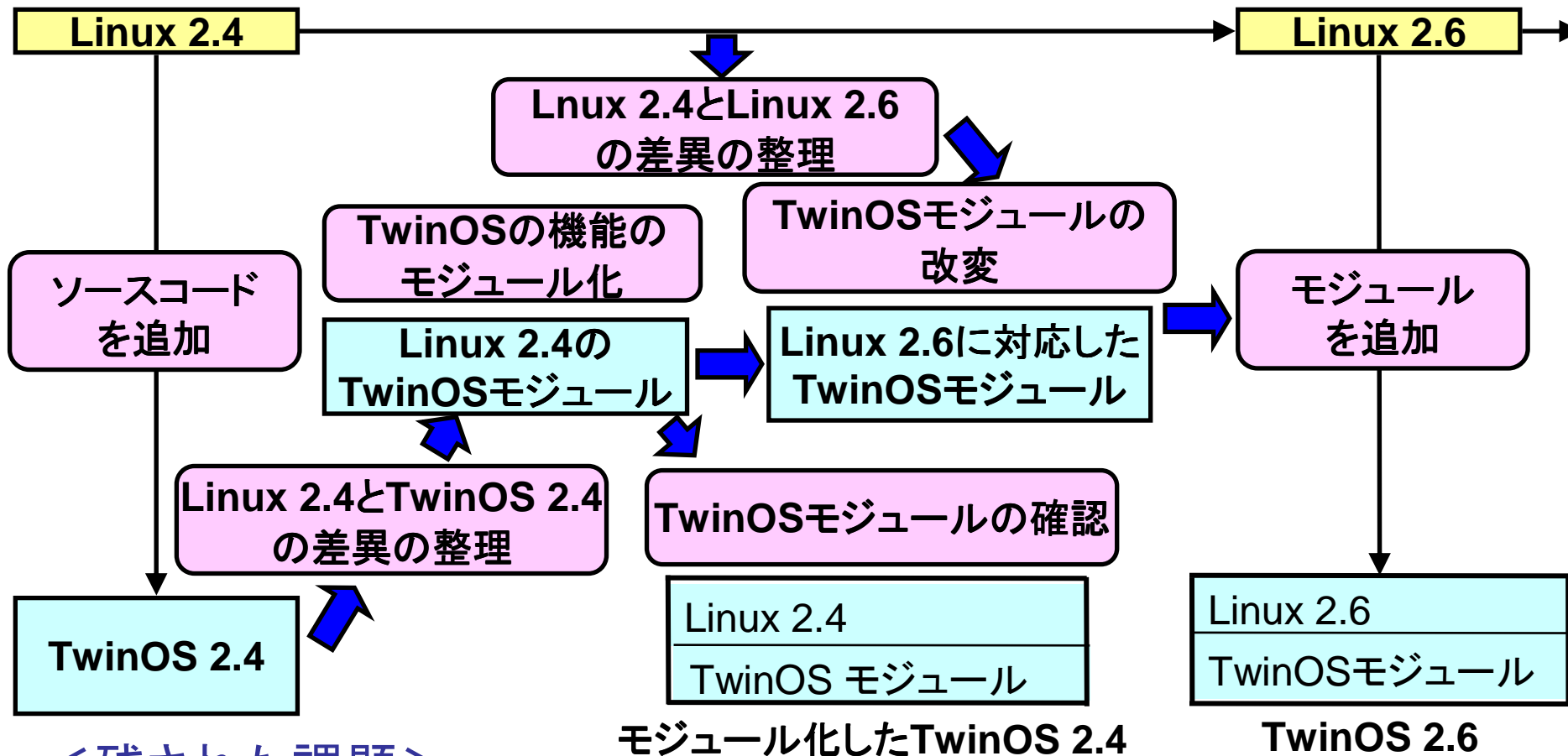
記述言語がアセンブリ言語からC言語に変更

<TwinOS 2.6での対処法>

- (1) マクロ使用によるマジックナンバーの削除
- (2) メモリマップ分割方法の簡易化
- (3) 記述言語をC言語に変更

本発表のまとめ

TwinOSのバージョンアップ方式を提案しメモリ分割機能を実装



＜残された課題＞

- (1) メモリ分割機能以外のモジュールのLinux 2.6への適用
(2) Linux 2.6以外へのTwinOSの適用可能性の調査