

## はじめに

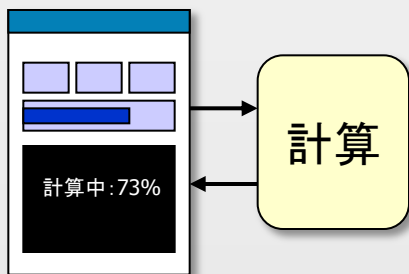
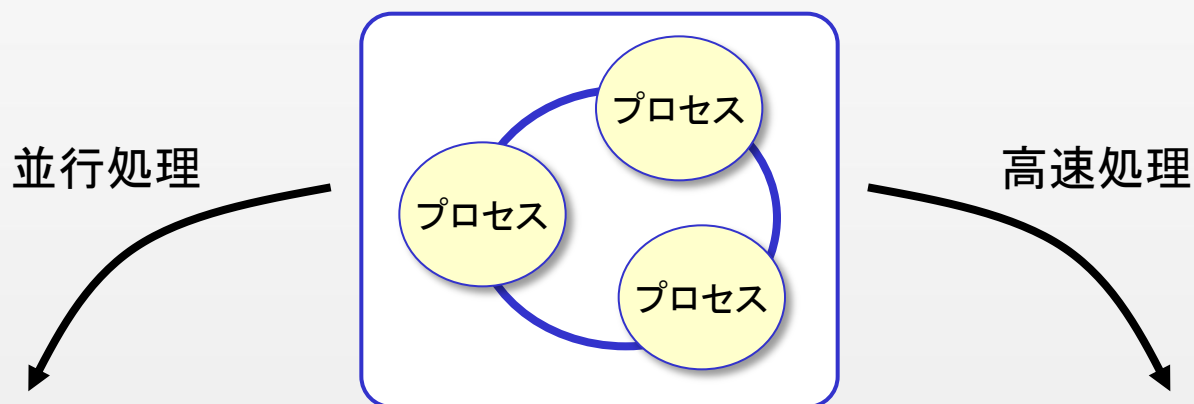
平成21年9月4日

トップエスイープロジェクト

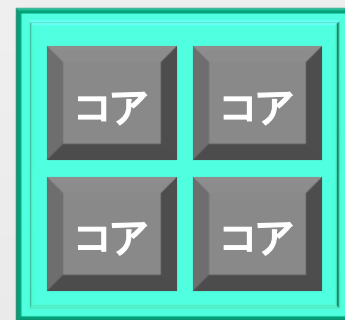
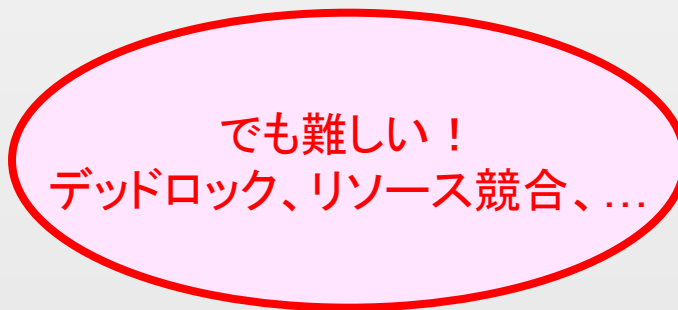
磯部祥尚（産業技術総合研究所 & 国立情報学研究所）

## 本講座の目的

- 高信頼な並行システムの開発方法を習得する。



ユーザインターフェースなど



マルチコアCPUなど

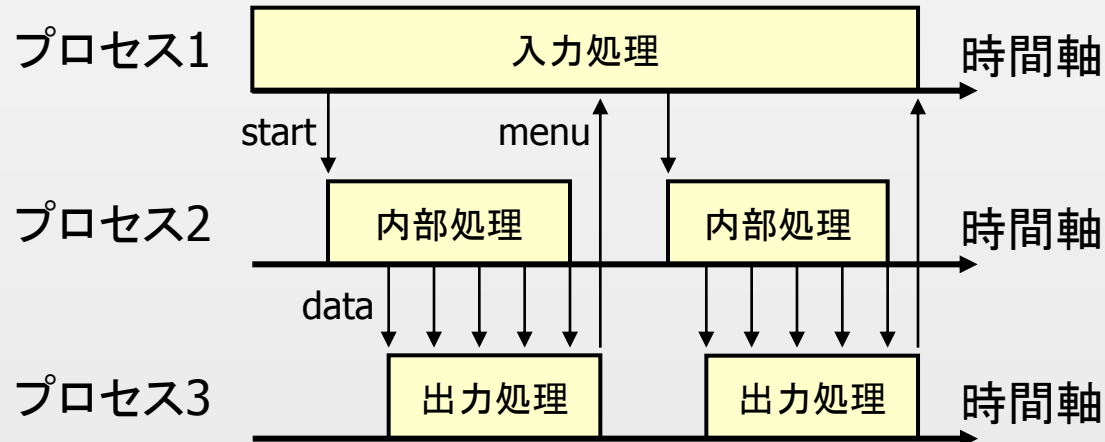
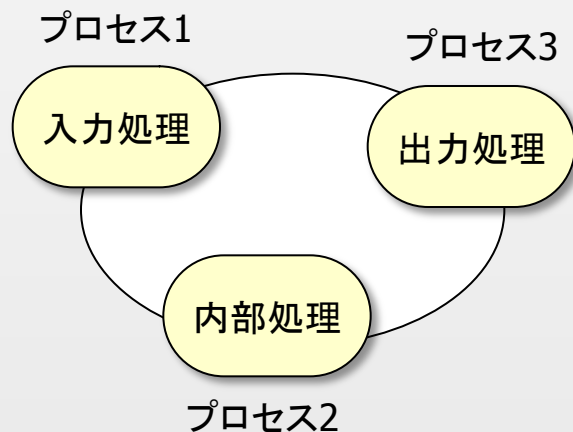


## はじめに

- 並行システムの概論
- JCSP, FDR, CSPとは
- 本講座の特長

## 並行処理の例

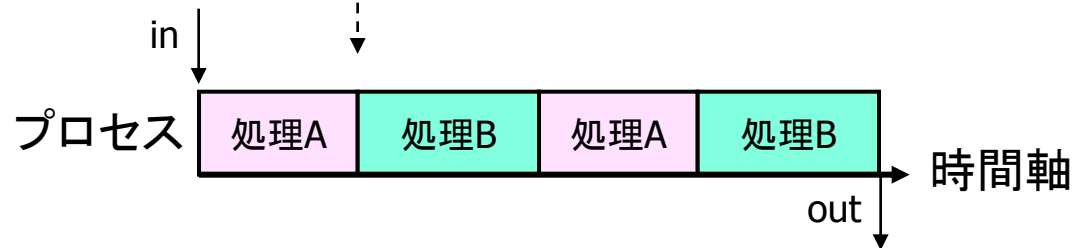
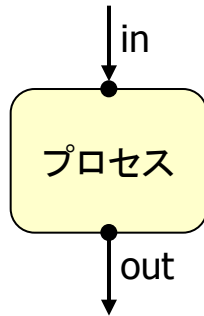
- 入力処理プロセスで常に入力を受け付けつつ、
- 内部処理プロセスで入力操作に応じて内部処理を実行し、
- 出力処理プロセスで内部処理中にその進捗状況を表示する。



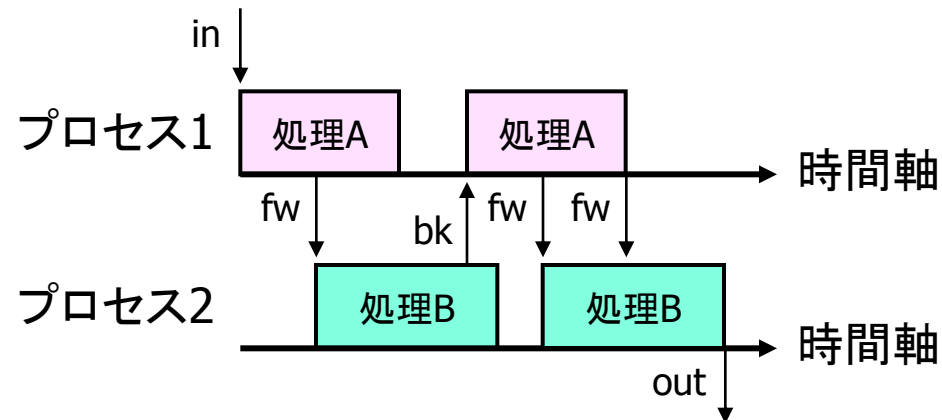
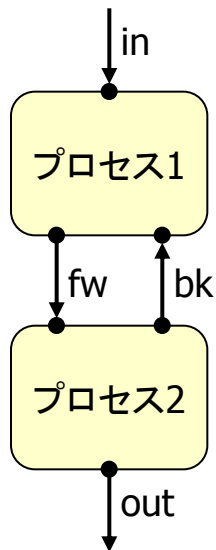
## 高速処理の例

A, B: 互いに依存関係があり、同時に実行する必要のない2つの処理

### 逐次処理

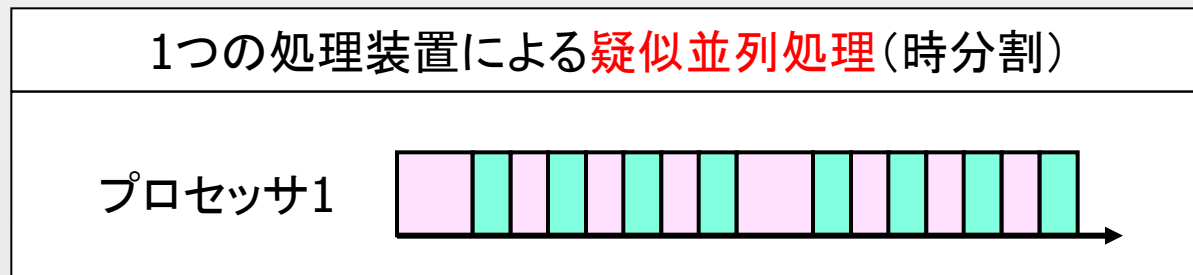
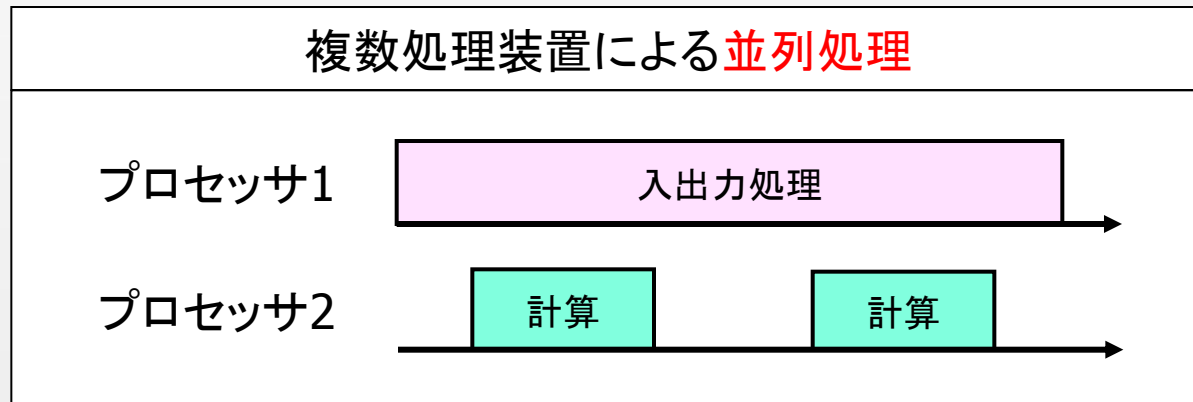
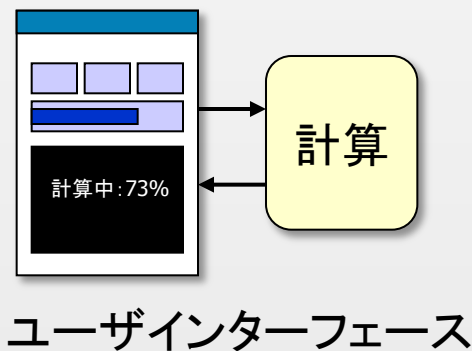


### 並列処理



## “並列処理”と“並行処理”の違い

- **並列処理**: 複数の処理を同時に実行すること。
- **疑似並列処理**: 複数の処理を時分割で実行すること。
- **並行処理**: 複数の処理を時間的重なりをもって実行すること。

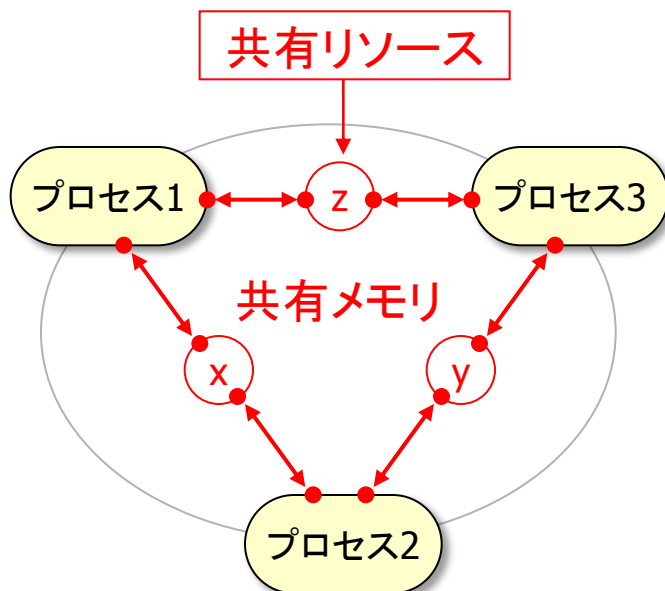


**並行処理の意味は並列処理と疑似並列処理の両方を含む**

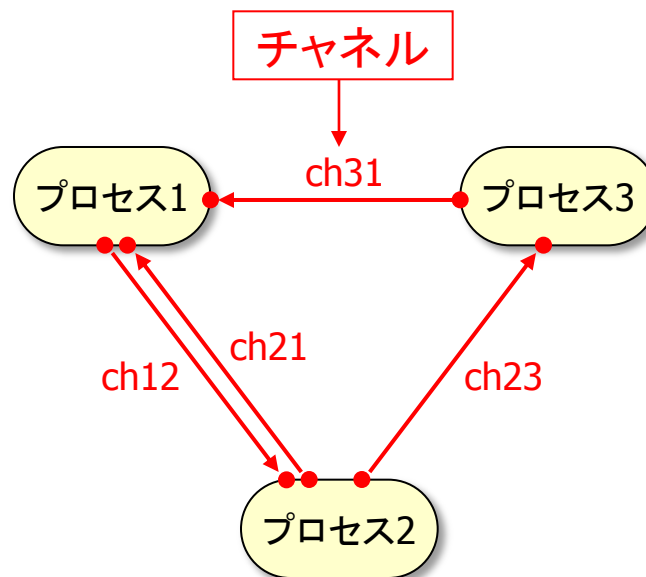
## 2つの通信方式

- **共有メモリ通信方式**: 共有メモリ上でメッセージを交換  
⇒ 処理が軽い(早い)
- **メッセージパッシング通信方式**: チャンネルを通してメッセージを送受信  
⇒ 動作が分かりやすい

共有メモリ通信方式

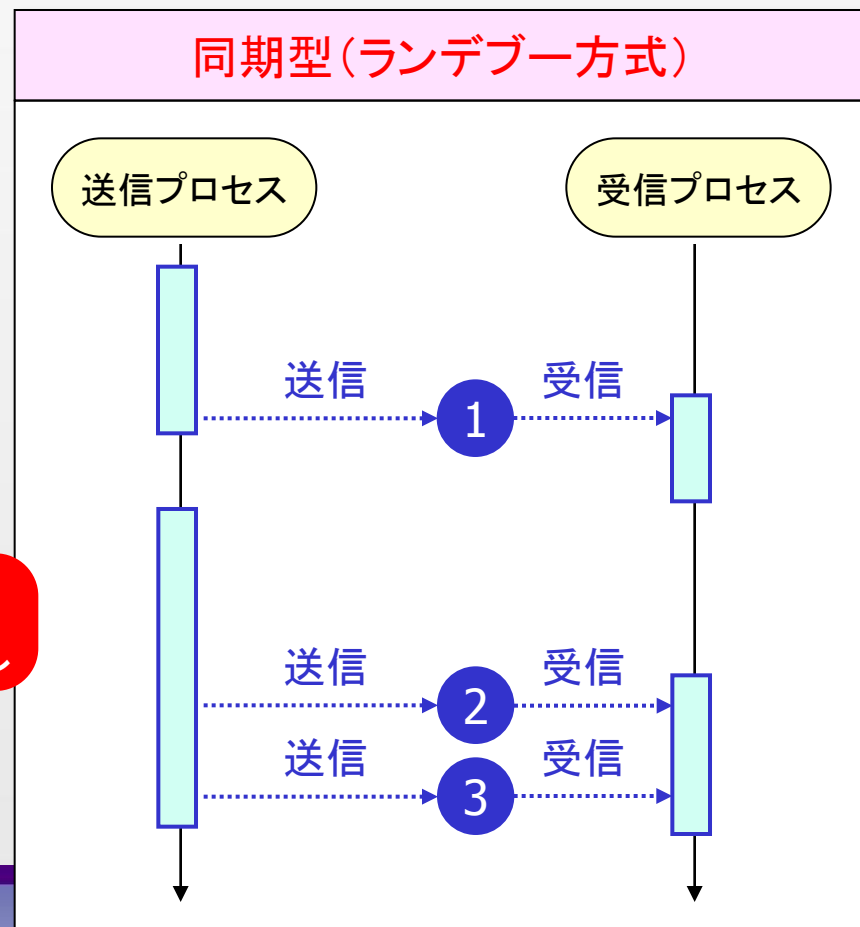
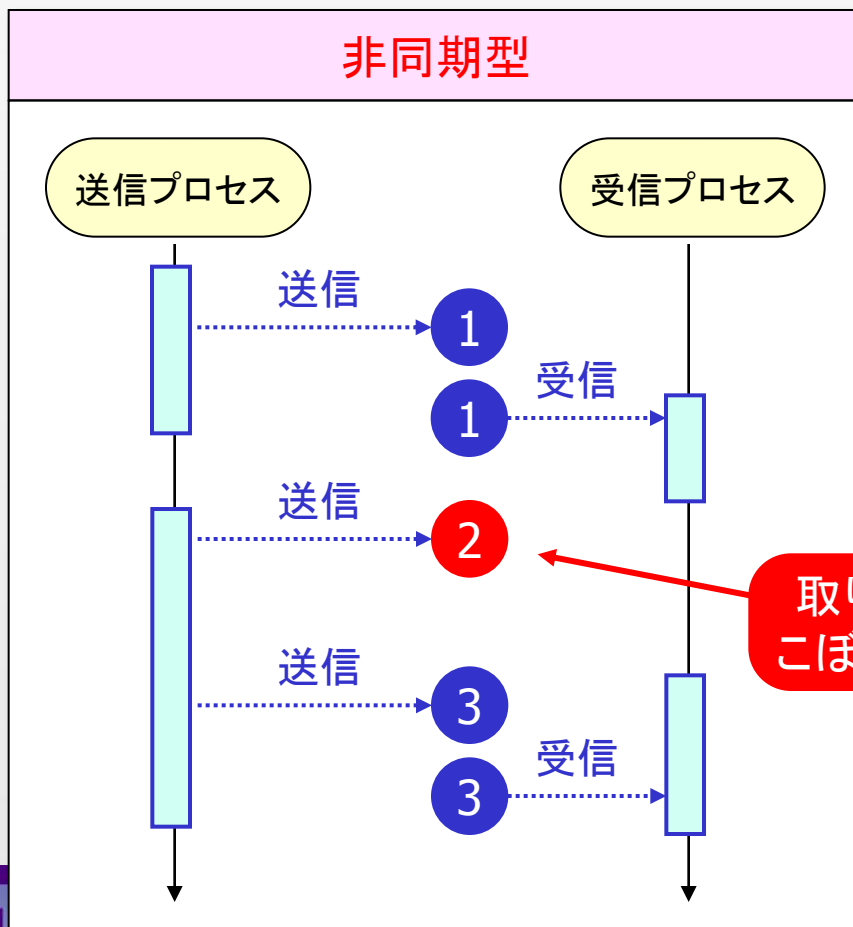


メッセージパッシング通信方式



## メッセージパッシング通信方式

- **非同期型**: 送信と受信は**別々**に起きる ⇒ 処理が軽い(早い)
- **同期型**: 送信と受信は**同時**に起きる ⇒ メッセージの**取りこぼし**がない。







## はじめに

- 並行システムの概論
- JCSP, FDR, CSPとは
- 本講座の特長



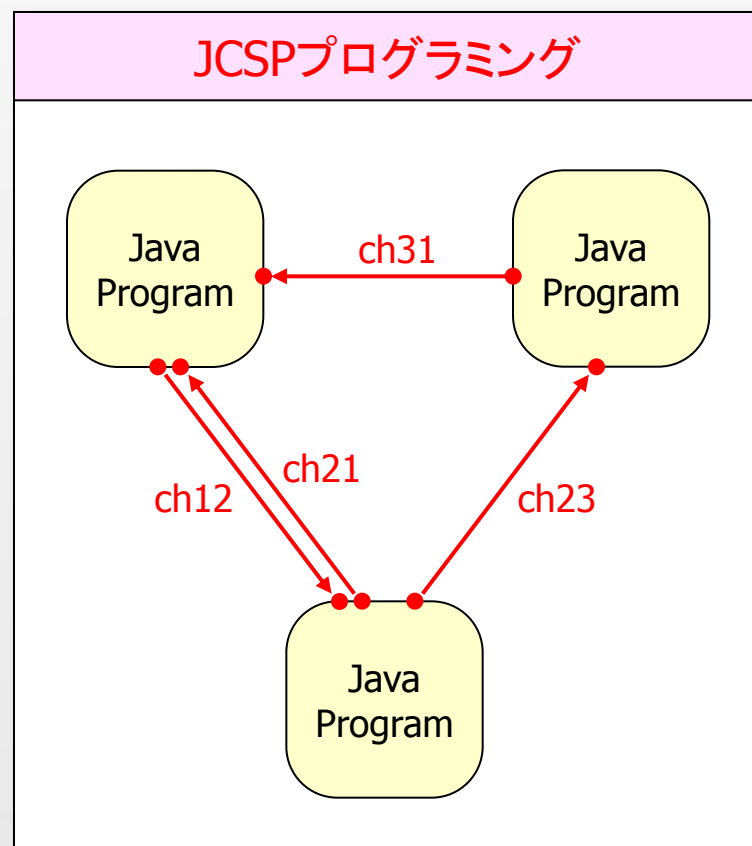
## Javaライブラリ: JCSP

- JCSP: Java で 同期型メッセージパッシング通信を実装するためのライブラリ

新しい言語を覚えなくてよい  
(敷居が低い)

比較的その動作を予測しやすい  
(信頼性を高められる)

信頼性の高い並行システムを構築できる！



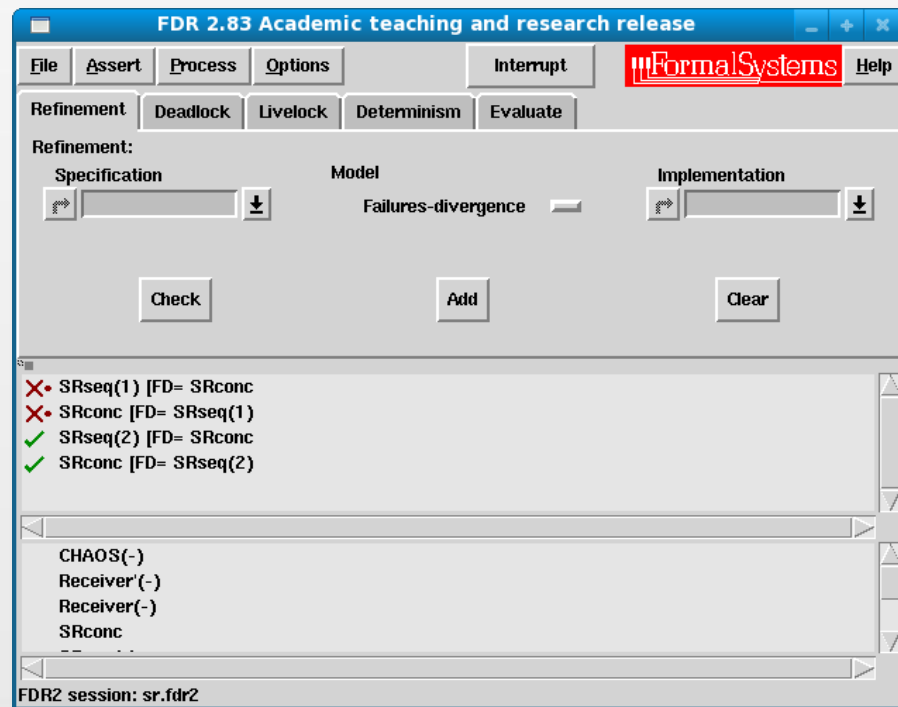
## モデル検査器: FDR

- FDR: JCSPの 同期型メッセージパッシング通信を検証できるツール

### FDRによる検証画面

並行動作の正しさを判定できる  
(さらに信頼性を高められる)

さらに信頼性の高い並行システム  
を構築できる !!





## プロセス代数: CSP

- なぜ**JCSP**の 同期型メッセージパッシング通信を**FDR**で検証できるのか？

⇒ **JCSP** : プロセス代数**CSP** のモデルを**実装**するためのJavaライブラリ

**FDR** : プロセス代数**CSP** のモデルを**検査**するためのモデル検査器

CSP: 同期型メッセージパッシング通信方式を採用しているプロセス代数

プロセス代数: 並行システムを形式的に記述し、解析するための理論

プロセス代数CSPでモデル化し、モデル検査器FDRで検証し、Javaライブラリで実装することによって、理論的に検証された高信頼な並行システムを実装できる !!!

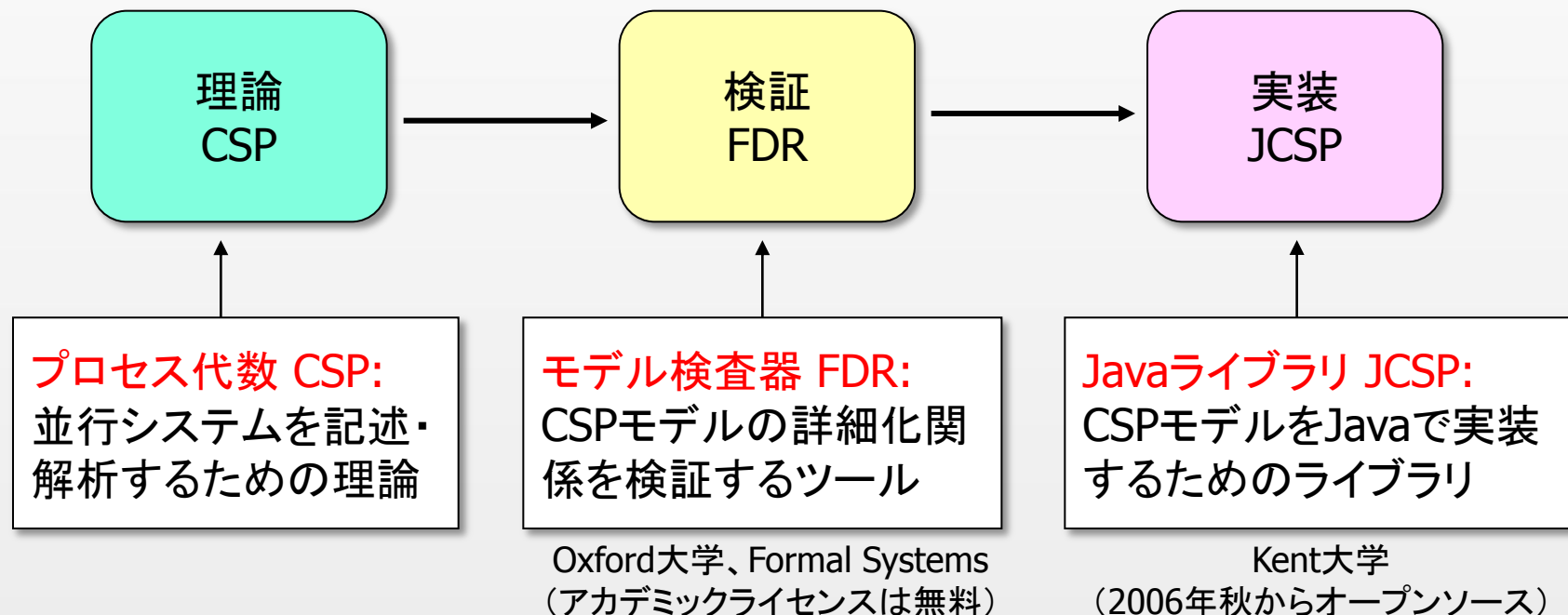


## はじめに

- 並行システムの概論
- JCSP, FDR, CSPとは
- 本講座の特長



## 本講座の3つのキーワード



CSPモデル	システムのCSP記述(形式的な記述)
FDRスクリプト	FDRで読み込み可能な記述(FDR入力言語)
JCSPプログラム	JCSPを利用しているJavaプログラム



## 本講座の構成

### 概要

レベル1

第1章 CSP, FDR, JCSP概論

### 入門

レベル2

第2章 CSP入門

第3章 FDR入門

第4章 JCSP入門

### 基礎

レベル3

第5章 CSP理論

第6章 CSP, FDR検証

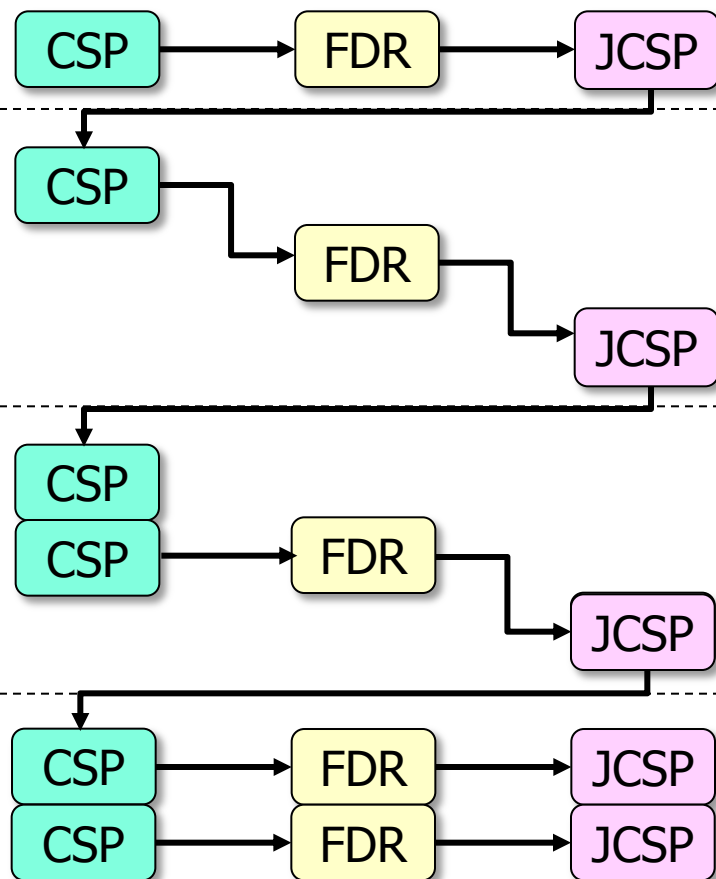
第7章 JCSP実装

### 応用

レベル4

第9章 CSP, FDR, JCSP応用

第10章 CSP, FDR, JCSP実践





## 習得する知識 & 技術

- プロセス代数CSPの基礎知識:

CSPによる**モデル化**、FDRによる**検証**、JCSPによる**実装**の方法を正しく理解するために必要な基礎知識の学習。

- モデル検査器FDRによる検証技術:

オートマトン(クリプキ構造)と時相論理に基づくモデル検査器(SPIN, SMV等)とは一味違った、**プロセス代数に基づく**モデル検査器FDRによる検証方法の習得。

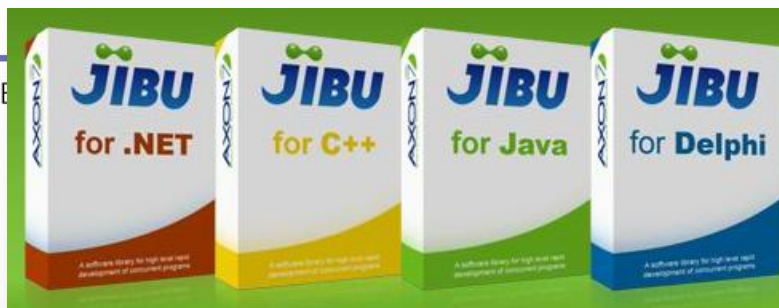
- ライブラリJCSPによる並行プログラミング技術:

CSPモデルに基づく**並行プログラミング(分散プログラミングを含む)**技術の習得。

(**実装を意識したモデル化が重要、検証には理論の基礎知識が必要**)



## Java以外にも有効



[1]

- JCSP (Java) 以外にもCSPモデルを実装するためのライブラリがある。基本的な考え方は同じで、本講座で習得する実装技術は**他のプログラミング言語**にも有効である。

ライブラリ	言語	研究開発元
JCSP	Java	ケント大学 (QuickStone)
C++CSP	C++	ケント大学
CHP	Haskell	ケント大学
JIBU	C++,Java,Delphi,.NET	コペンハーゲン大学 (Axon7)
PyCSP	Python	トロムソ大学 & コペンハーゲン大学
CTJ	Java	トゥエンテ大学

[1] JIBU, AXON7, <http://www.axon7.com/> から写真を引用

## ソフトウェア以外にも...

- Transputer (1981年～1996年、INMOS社)



Transputer[1](Inmos社)

並列コンピューティングに特化した最初の汎用マイクロプロセッサであり、その記述言語OccamはCSPモデルをベースにしていた。残念ながら1996年に製造終了となった。

- XMOS (2008年～、XMOS社)



XMOS[2](XMOS社)

XMOSはイベント駆動型マルチスレッドプロセッサであり、XMOS-XS1-G4では4コア32スレッドによる並列処理が可能である。その記述言語XCはC言語のサブセットに同期型通信チャネルを追加したような言語である。この概念はJCSPに似ており、本講座で習得した技術はXMOS設計にも有効であると思われる。

[1] ウィキペディアの「トランスピュータ」(<http://ja.wikipedia.org/wiki/トランスピュータ>)の写真から引用

[2] XMOS社のウェブサイト(<https://www.xmos.com/products>)の写真から引用



## まとめ

- CSPの基本アイデアはC. A. R. Hoareによって1978年に発表された。
- しかし過去の話ではなくCSPの理論研究は今も続けられている。
- そして最近はCSPの実装関係の研究もより活発になっている。
  - ⇒ マルチコアCPUの登場により並行システムの重要性が高まっている。  
(誰でも簡単に並列プログラミング)
  - ⇒ ネットワークの普及によって分散プログラミングのニーズも高まっている。

CSPの理論、検証、実装の基本的な流れを学習する意義は高い。