

# Asynchronous Exceptions

原案: 田中

英文: 菅原

解答: 田山、青木

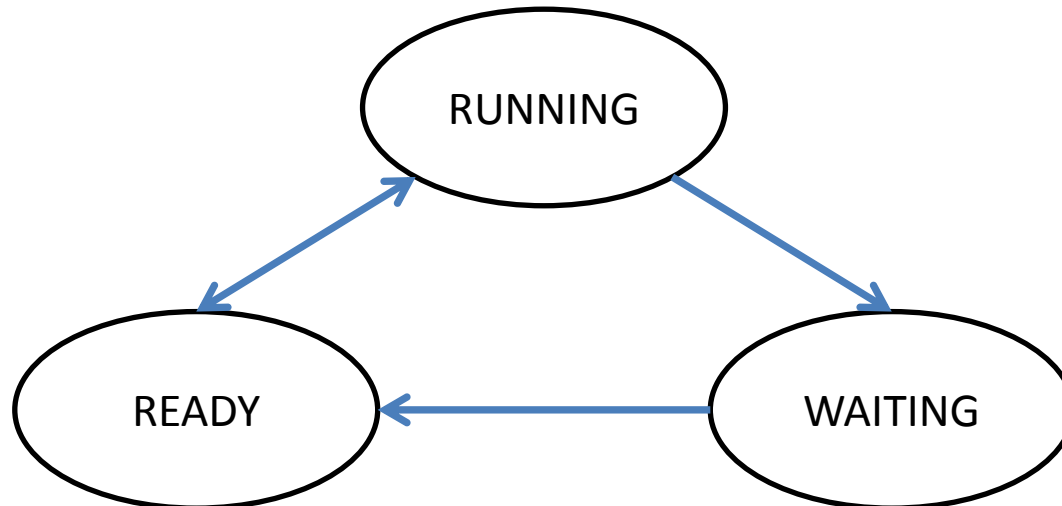
解説: 青木

# 問題概要

- SMP(対象型マルチプロセッサ)のシミュレーションを行う
  - 全てのスレッドの終了時刻を求める
  - 時間やリソースの上限を超える場合も検出
- 複数のCPU、スレッド、コードブロック
  - スレッドは1つのコードブロックをもつ
  - スレッドはコードブロックを上から順にendまで実行
  - 各CPUには高々1つのスレッドが割り当てられる

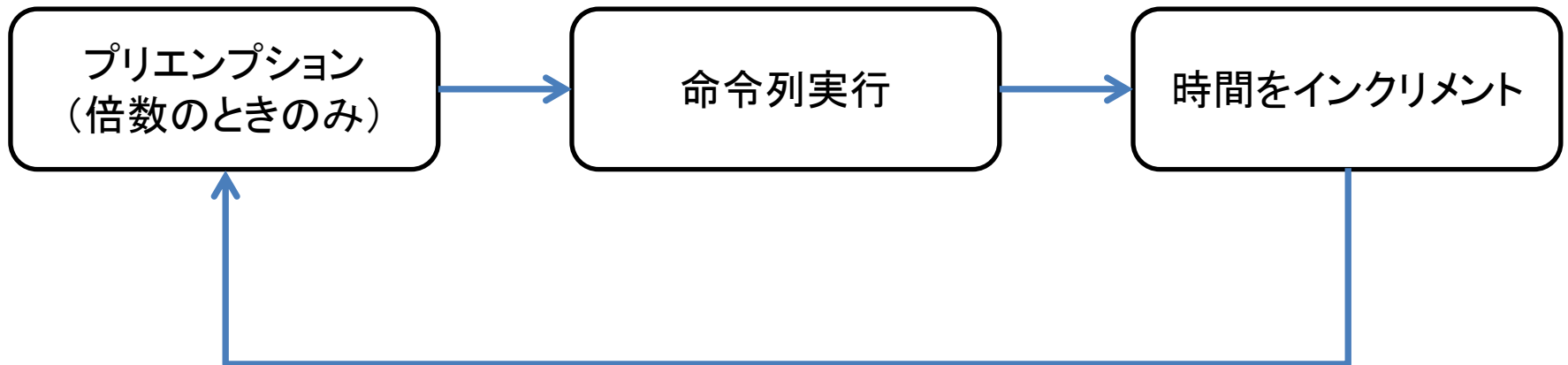
# スレッド

- スレッドは、3つの状態
  - RUNNING CPUに割り当てられている状態
  - READY 実行待ち行列に入っている状態
  - WAITING ロックしている状態



# スケジューリング

- CPUは、スレッドを切り替えながら実行
- プリエンプション以外でもスレッド割り当て発生
  - end, killThread, lock, yield命令の後



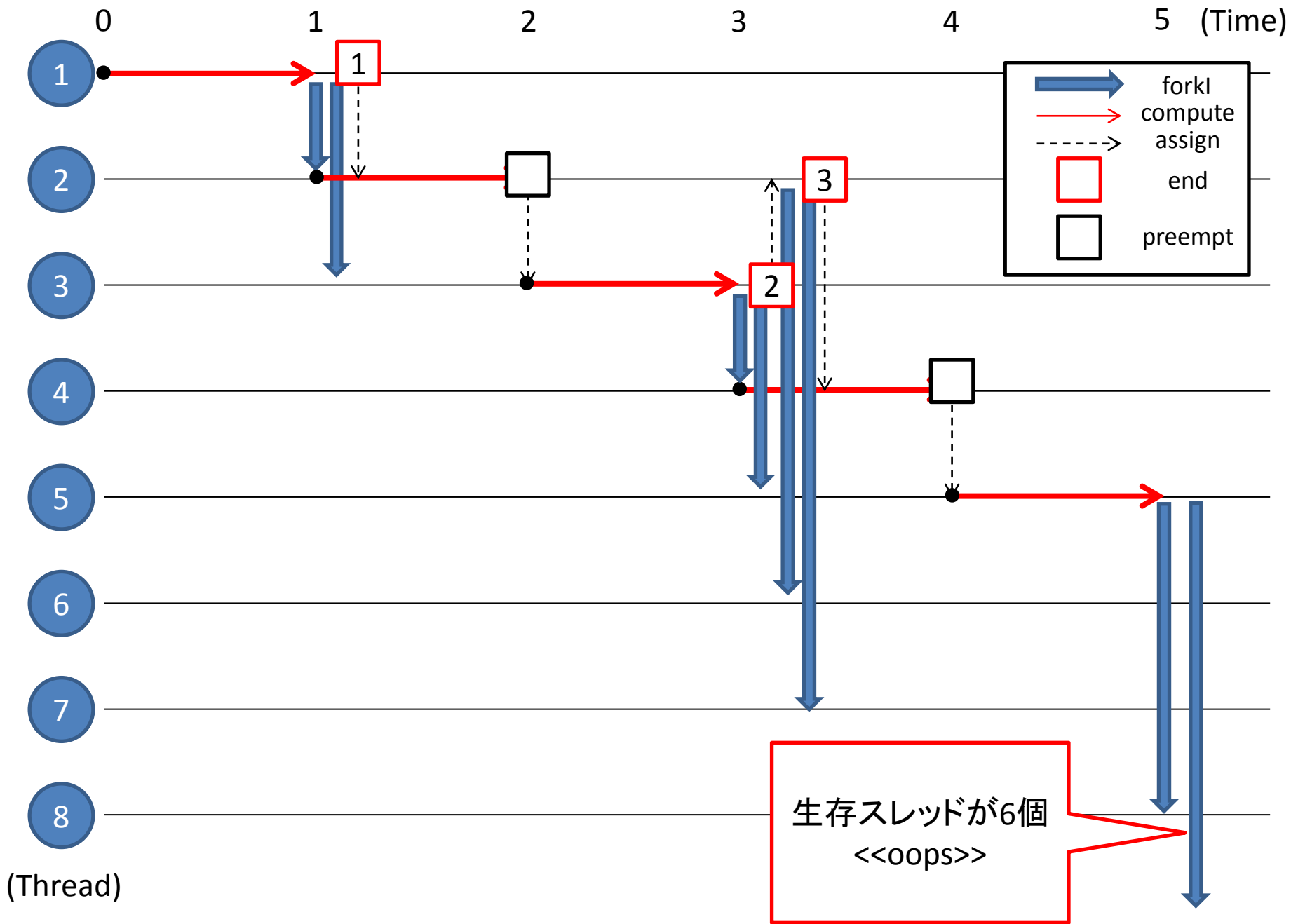
# 命令

- computeのみが、時間を消費する命令
- compute以外の命令は、時間を消費しない
  - 同時刻・同CPUで複数の命令が実行され得る

# 例: Sample 4の場合

- 自分のコードブロックを2つforkするプログラム

```
5 5
1
2
1
semaphore 1
1
codeBlockA:
compute 1
hoge <- fork1 codeBlockA
hoge <- fork1 codeBlockA
end
```



# 想定解法

- がんばって実装する



# ジャッジ解

- 田山 C++: 819行、21619バイト
- 青木 Java: 566行、19051バイト
- 実装量が多い

# 結果

- Submit: 0