

オペレーティングシステム

第4回 プロセススケジューラ

情報科学メジャー
鍋木崇史

マルチタスク処理

複数のプロセスを切り替えながら実行すること

- ・プロセスが順次切り替わりながら実行されていく
- ・ユーザ側から見ると複数のプログラムが同時に実行されているように見える
- ・あるプロセスが待ち状態となっても他のプロセスが実行されるため、全体として処理速度が向上する

実行スケジューリング

実行可能状態にある各スレッドに対して優先順位を与え、実行順序を決める

スケジューリングの観点

- 1) タイミング
- 2) 優先度付与基準
- 3) スケジューリングアルゴリズム

タイミングの観点による方式

ノンプリエンプティブ・マルチタスク

各プロセス自身が、自発的に短い時間間隔でOSに制御を戻すもの

プリエンプティブ・マルチタスク

タイマ割り込みを用いて制御を強制的にOSに戻すもの

デバイスコントローラ

(1) タイマ割り込み発生
(割り込みベクタ n を通知)

プロセッサ

実行中のプログラムの

PC PSW

割り込みベクタ・テーブル

割り込みベクタ n に対応した

PC PSW

デバイスコントローラ

(1) タイマ割り込み発生
(割り込みベクタ n を通知)

プロセッサ

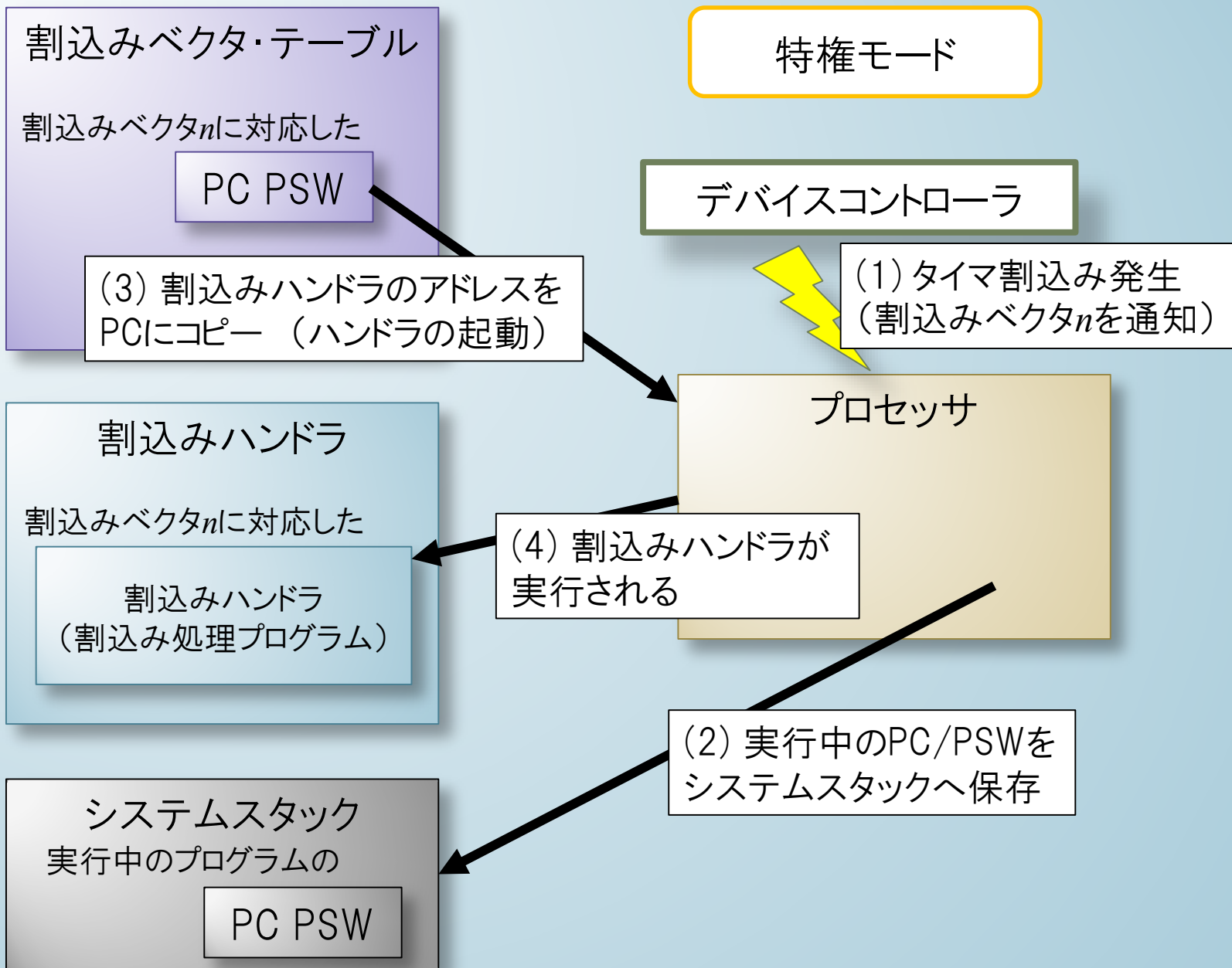
実行中のプログラムの

PC PSW

システムスタック

(2) 実行中のPC/PSWを
システムスタックへ保存





特権モード

切り替え前のスレッド
コンテキスト保存領域

(5) コンテキスト保存

切り替え後のスレッド
コンテキスト保存領域

プロセッサ

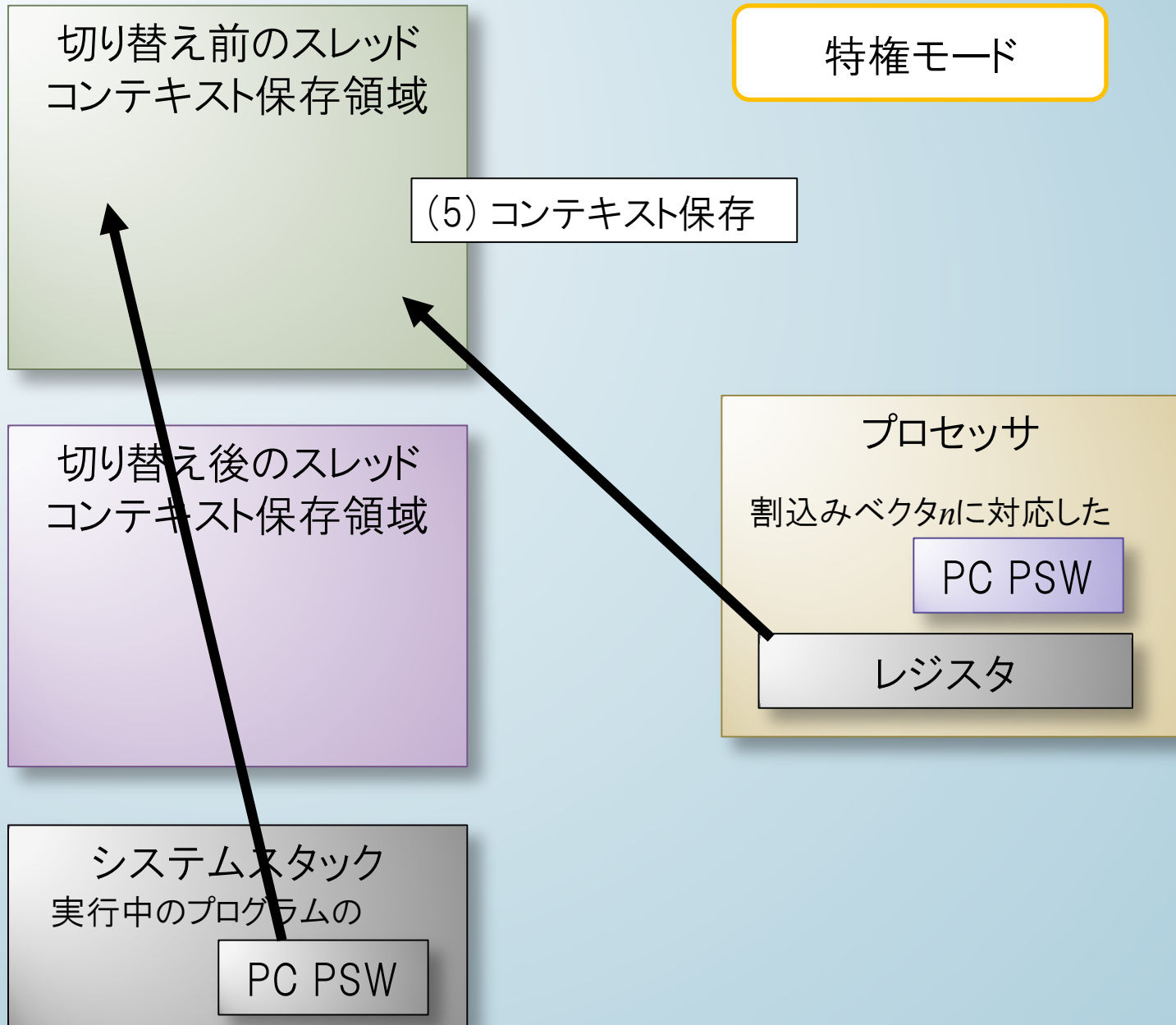
割込みベクタ n に対応した

PC PSW

レジスタ

システムスタック
実行中のプログラムの

PC PSW



特権モード

切り替え前のスレッド
コンテキスト保存領域

PC PSW

レジスタ

切り替え後のスレッド
コンテキスト保存領域

PC PSW

レジスタ

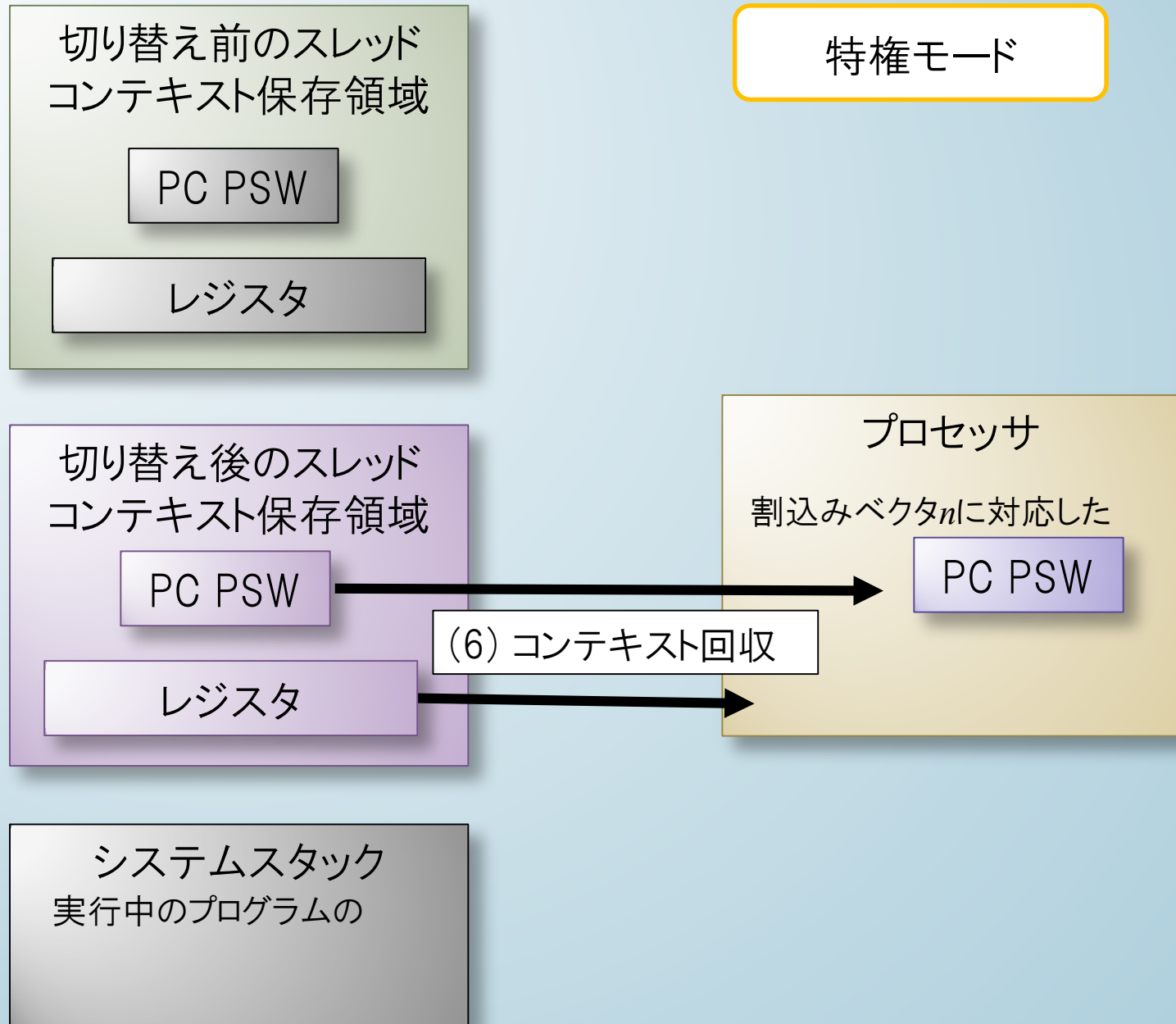
プロセッサ

割込みベクタ n に対応した

PC PSW

(6) コンテキスト回収

システムスタック
実行中のプログラムの



優先度付与基準による観点

応答時間

処理結果が得られるまでの時間

スループット

単位時間あたりの処理効率

プロセッサ利用率

プロセッサが実行状態にある割合

締切

実時間処理においては一定の時間内に応答を返すことが必要
(cf. バッチ処理: 人との対話なしにジョブを順次処理するもの)

公平性

特定のスレッドがいつまでの実行されない状態(飢餓状態)を招く

実行スケジューリング・アルゴリズム

先着順

方法: 実行可能になった順番で高い優先度を与える

長所: 単純

短所: 実行中のスレッドが待ち状態か終了状態になるまで次のスレッドが実行されない → 飢餓状態に陥るリスク

時間順

方法: 処理時間の短いスレッドに高い優先度を与える

長所: 平均応答時間が最小になる

短所: 実行時間をあらかじめ把握している必要がある
一般的に実行時間の予測が不可能

ラウンドロビン

方法: 単位時間にスレッドを切り替え、待ち時間が長いスレッドに高い優先度を与える

長所: すべてのスレッドが平均的に実行できる

短所: 実行していたスレッドのタイムスライス(単位時間)が切れると優先順位が最低になる

実時間スケジューリング

実時間(リアルタイム)

事象が発生してから一定時間後の締め切りまでに
処理を完了できること
→このような処理を目的としたOSをリアルタイムOSと呼ぶ

ハードリアルタイム

締め切りが守られないと処理の価値が著しく低下するものや
OS外に大きな損害を与えるもの
例) 航空機や自動車の制御

ソフトリアルタイム

締め切りが守られなくても処理を行うことの意味があるもの
平均的に締め切りを守ることが重要な場合

実時間スケジューリング・アルゴリズム

締切順

方法： 締切までの余裕の時間が短いものほど先に処理をさせる

特徴： スケジューリングのタイミングでスレッドの優先順位が変化する

例) 時刻10に下表のような状態のとき、優先順位はどうなるか？

	実行可能になった時刻	残り実行時間	処理完了 締切時刻	優先順位
A	1	6	18	2
B	3	8	30	4
C	4	1	35	5
D	7	2	13	1
E	9	4	25	3

レート・モニタック

方法： 処理が起動される頻度が多いスレッドに高い優先度を与える

特徴： 横取り可能な固定優先度スケジューリングで最適

課題1

システムコール fork について調査し、報告せよ。

※教科書 Linuxのしくみ p28-30あたりを参考にしてください

課題2

教科書 Linuxのしくみ p40-51 の実験を実施する。

Moodleのtask1.cをコンパイルし、次の実験条件でプログラムを動作させて、作図ツールを利用して図04-03～図04-08を作成して考察せよ。

```
$ taskset -c 0 ./task1 1 100 1 > task1-1.txt
```

```
$ taskset -c 0 ./task1 2 100 1 > task1-2.txt
```

```
$ taskset -c 0 ./task1 4 100 1 > task1-4.txt
```

```
$ gnuplot
```

```
gnuplot> plot "filename" using 2:3:1 with points pointtype 7 pointsize 2 lc variable
```

```
or
```

```
gnuplot> plot "filename" using 2:3:1 w p pt 7 ps 2 lc variable
```

課題3

下の表の場合、時間順スケジューリングでは優先度がどのように設定されるか記入せよ（必要に応じて教科書 オペレーティングシステム P30～32を参照してください）

	実行可能になった時刻	残り実行時間	処理完了 締切時刻	優先順位
A	1	6	18	
B	3	8	30	
C	4	1	35	
D	7	2	13	
E	9	4	25	