

オペレーティングシステム(OS)

柴山 潔

9. プロセス管理 (3)

- プロセス割り付けとプロセス領域(2)
- プロセススイッチ

プロセス制御ブロック(PCB; Process Control Block)

[定義2.7] プロセス制御ブロック (PCB; Process Control Block)

- (a) あるプロセス(の実行)を管理・制御するための情報&その格納領域;と, (b) そのプロセスの実体 (=プロセス領域) へのポインタ;とを対にした情報 = **プロセスコンテキスト** (process context)

= (OSから見る)「プロセスの概要」= **論理的なプロセスコンテキスト**

- OSは, **プロセスの実体** (=プロセス領域そのもの)ではなく, その「概要」である**PCB**を操作/処理することによって, **間接的にプロセス**を管理&制御
- (OSが行う)プロセス管理 = **PCB**で保持する**プロセス状態(概要)**の管理
- プロセスの切り替え (=プロセススイッチ)は, 実際には, **PCB**の切り替え = **PCB**の退避と回復 (= **プロセスコンテキストスイッチ**)で

PCBの管理(リストとキューによる例)

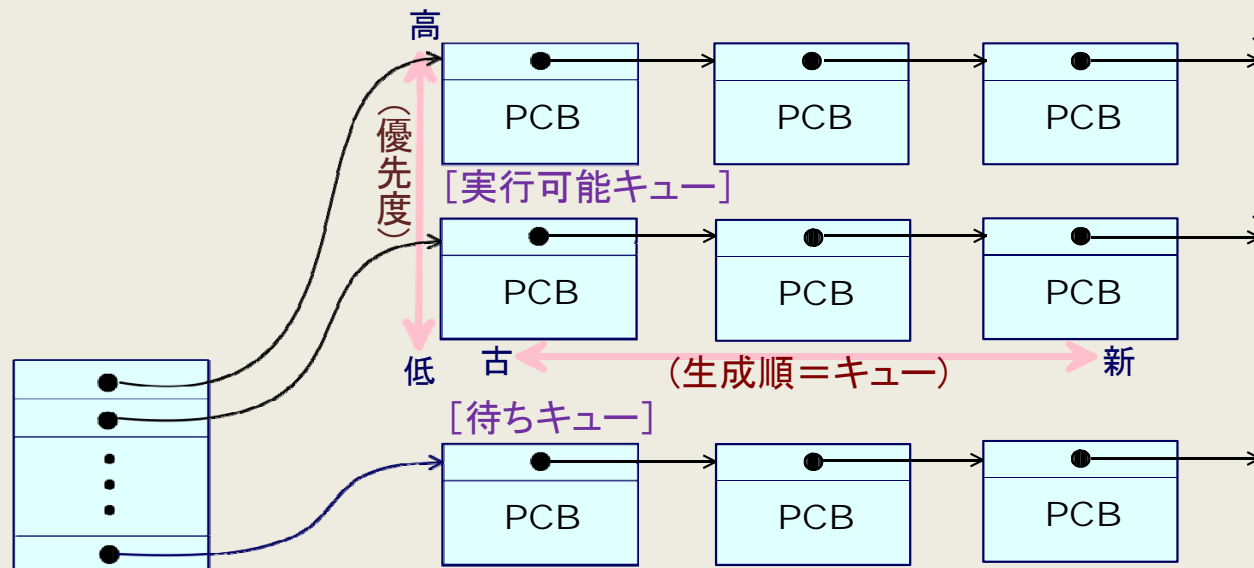
- 実行可能状態と待ち状態のPCBどうしはリスト(list) & キュー(queue)構造で管理

(例)

(a) 実行可能キュー: 実行可能状態のプロセスどうしを連結したPCBリスト

(b) 待ちキュー: 待ち状態のプロセスどうしを連結したPCBリスト

- 実行可能キューや待ちキューでは, プロセス実行の優先度やプロセスの生成順をキュー構造で保持

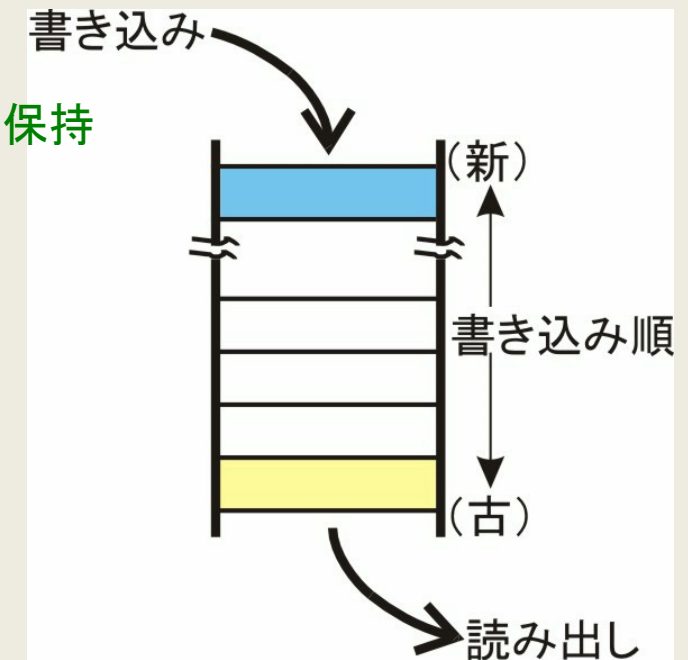


キュー(queue)

- 書き込みと読み出しの独立したアクセスポートを両端に備える1次元メモリ(構造)
 - **FIFO (First In First Out)**: “一番最初に格納したものを一番最初に読み出す”順でアクセス
 - “書き込み順序を保持して, その書き込み順(正順)で読み出す”機能を自然に装備

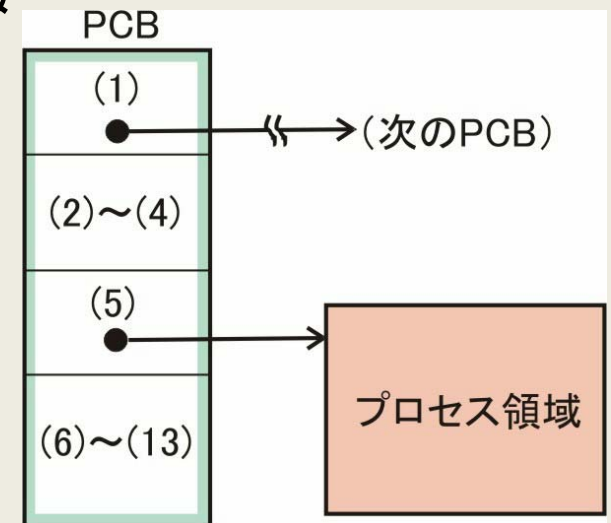
[OSでの適用例]

- (a) 実行可能/待ち状態プロセスのPCBリスト
- (b) FCFSのスケジューリングアルゴリズムでのプロセス実行順序の保持
 - ◆ メインメモリを流用&ソフトウェア機能で実現



プロセス制御ブロック(PCB)の構成例(1)

- (1) 次のPCBへのポインタ
- (2) プロセス識別番号(ID)
- (3) プロセス状態
- (4) スケジューリング情報: (例) 優先度; 生成時刻/生成順; ライフタイム (生成から消去までの経過時間)
- (5) プロセス領域へのポインタ: 対応するプロセス領域である, (A) コード領域; (B) データ領域; (C) ヒープ領域; (D) スタックフレーム領域; それぞれへのポインタ
- (6) プログラムカウンタ(PC)の退避領域
- (7) 汎用レジスタの退避領域



プロセス制御ブロック(PCB)の構成例(2)

- (8) プロセス保護情報: OSのメモリ保護機能用情報(=プロセス権限/ケーパビリティ(capability))
- (9) プロセッサ管理情報: (例) プロセッサ状態; 各種のハードウェア機構の状態(=コンディション/条件); タイマ
- (10) メモリ管理情報: (例) 当該プロセスが生成したプロセス(=子プロセス)に関するMMへの割り付け情報
- (11) ファイル管理情報: (例) オープンしたファイル; ファイル保護情報
- (12) 入出力管理情報: (例) 割り込みや事象に関する情報;

◆(8)~(12): 実際には, 各情報の実体(内容)の格納先アドレスとそのサイズ(→可変)

(13) ユーザ情報

プロセッサ状態ワード(PSW; Processer Status Word)

[定義2.8] プロセッサ状態ワード (PSW; Processer Status Word)

- プロセススイッチに際して退避が必要となる内部装置(プロセッサとメインメモリ)内の**各種のハードウェア状態の組(セット)**
 - ある時点(*)で**PSW**を**退避** → 後刻にその**PSW**を**回復**=内部装置のハードウェア状態は退避した時点(*)に**復帰**

= (OSから見る)「当該時点でのハードウェアの概要」= 物理的な**プロセスコンテキスト**

- **割り込み処理**では, 割り込みハンドラによる割り込み要因ごとの処理の前後に, **割り込まれるユーザプロセス**を実行している内部装置の状態である**PSW**



切り替え(退避/回復, スイッチ)

割り込む**OSプログラム**を実行する内部装置の状態である**PSW**

- **PSW**の切り替え(=プロセスコンテキストスイッチの一部)= **内部装置ハードウェアの切り替え**

プロセッサ状態ワード(PSW)の構成例

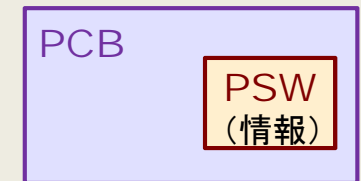
- (a) プロセッサ状態
- (b) コンディション(条件)
- (c) プログラムカウンタ(PC)
- (d) 汎用レジスタ
- (e) メモリイメージ(memory image): そのプロセスが使用しているメインメモリ内容(実メモリ空間) or その完全なコピー
 - 主として, プロセス領域(全部) = プログラム(コードとデータ), ヒープ, スタックフレーム
- (f) メインメモリについての管理情報
- (g) 割り込みの優先度

◆ (a)~(c)だけを“狭義のPSW”と, その狭義のPSW + (d)~(g)を“(ハードウェアとして見る)プロセスコンテキスト”と区別することも

PCBとPSW(実際)

■ **PCB** (プロセス制御ブロック) = 論理的なプロセスコンテキスト

➤ PCBには, PSW(情報)を埋め込む



■ **PSW** (プロセッサ状態ワード) = 物理的なプロセスコンテキスト

➤ OSによるプロセス(コンテキスト)スイッチにおいて, 「メインメモリへの退避」& 「メインメモリからの回復」の各操作対象とするPCBのうちのハードウェア状態(情報)

(例)

- PCBの(9)プロセッサ管理情報はPSWの(a)プロセッサ状態や(b)コンディションを含む

プロセススイッチ (定義と実際)

[定義2.9] **プロセススイッチ (process switch)**

● 実行中状態のプロセスを切り替える (スイッチする) OS機能 (システムサービス)

■ 実際のプロセススイッチ (広義 = ユーザプロセスのスイッチ) ($X \rightarrow Y$)

➤ 割り込みをきっかけ

● 唯一プロセッサ (時間) 上で,

(1) 切り替え元のユーザプロセス(X)



(2) OS



(3) 切り替え先のユーザプロセス(Y)

を順次連続実行



プロセススイッチ(広義)の実際的手順 —ユーザプロセスX→Yの例—

➤ 割り込み = ユーザプロセスが使用しているプロセッサ(時間)のOSによる強制的な乗っ取り/取り上げ

(1) ユーザプロセスX → OS(カーネル)の切り替え

(2) OS(カーネル)による(ユーザ)プロセス管理

(2-1) カーネル機能による割り込み処理

(2-2) システムサービス機能によるプロセススケジューリング(狭義)

(2-3) カーネル機能によるプロセスディスパッチ } →切り替え先のユーザプロセスYを選定
プロセスディスパッチ

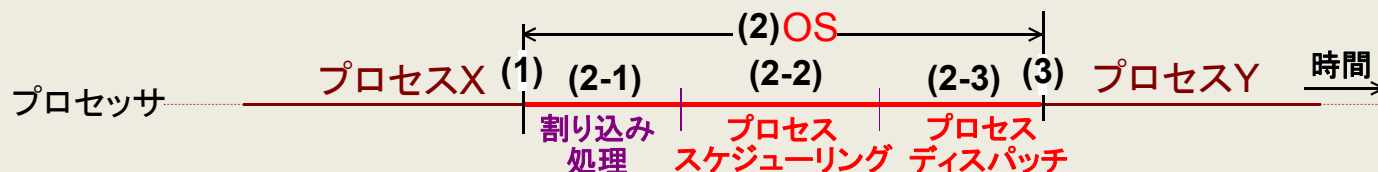
(3) OS(カーネル) → ユーザプロセスYの切り替え

➤ (2-3)を兼ねている機能 = プロセスディスパッチ(狭義)

= OSが使用しているプロセッサ(時間)のユーザプロセスYへの明け渡し/譲り渡し

(a) X = Y ならば, プロセスX(=Y)の再開

(b) X ≠ Y ならば, プロセスYのディスパッチ



プロセススケジューリングとプロセスディスパッチ

[定義2.10] プロセススケジューリング (process scheduling)

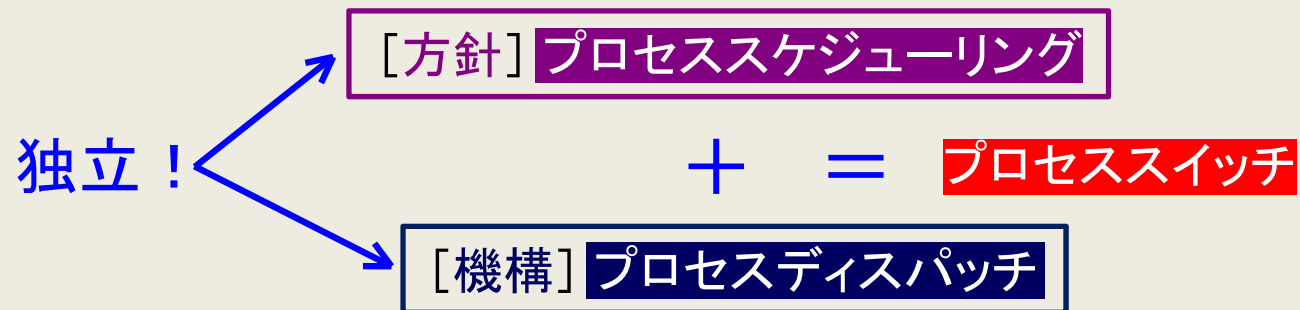
- プロセススケジューラが、スケジューリングアルゴリズムにしたがって、次に実行する(=プロセススイッチにおける切り替え先の)プロセスを選定するOS(システムサービス)機能
- プロセススケジューラ(process scheduler) = プロセススケジューリングというシステムサービスを実現する機構

[定義2.11] プロセスディスパッチ (process dispatch)

- プロセスディスパッチャが、プロセススイッチに際して、プロセスをプロセッサに割り付けるOS(カーネル)機能

- ディスパッチ(定義2.2)とプロセスディスパッチ(定義2.11)は本質的には同義
- 「あるプロセスをプロセッサに割り付け」というプロセスディスパッチ(定義2.11)によって生じる「そのプロセスの実行可能状態→実行中状態遷移」がディスパッチ(定義2.2)

【まとめ】プロセススイッチ機能の設計 —方針と機構の分離の例(再考)—



- プロセススイッチ機能: プロセススケジューリング方針にしたがって動作するプロセスディスパッチ機構

プロセススイッチ(実際)(再掲)

- 割り込みをきっかけとするユーザプロセス間の**プロセススイッチ**

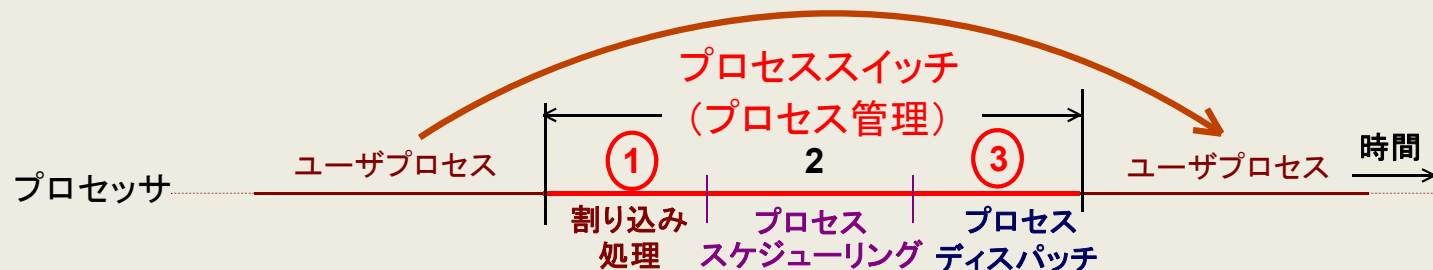
↓ (実際には) **プロセス管理**手順(1~3) = 広義の **プロセススイッチ**

① **割り込み処理**: [ユーザプロセス→OS(カーネル)] スイッチ

2. **プロセススケジューリング**(狭義): スイッチ先ユーザプロセスの**選定**

③ **プロセスディスパッチ**: [OS(カーネル)→ユーザプロセス] スイッチ

➤ ①と③の[ユーザプロセス⇔OS(カーネル)] スイッチ = 狭義の **プロセススイッチ**



プロセススイッチとプロセスコンテキストスイッチ

(具体的には)

■ プロセススイッチ(狭義)

$$= \left\{ \begin{array}{l} \text{“PCB”というソフトウェアプロセスコンテキストの切り替え} \\ + \\ \text{“PSW”というハードウェアプロセスコンテキストの切り替え} \end{array} \right.$$

➤ どちらも/併せてプロセスコンテキストスイッチ

↓ (実際には)

■ プロセススイッチ機構

= プロセスコンテキストスイッチ機構

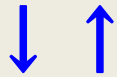
= PCB & PSWの切り替え機構

(上級)

プロセススイッチの実際的な手順 —プロセスコンテキストスイッチ機構の動作の観点(1)—

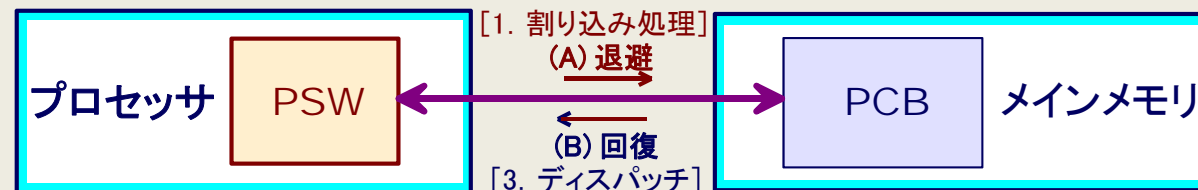
(A) プロセススイッチ手順1の“割り込み処理”におけるプロセッサ内PSWのメインメモリ内PCBへの退避

- スイッチ元の(=現在実行中)のユーザプロセスのPSW(プロセッサ内)を旧プロセスコンテキストとしての当該ユーザプロセスのPCB(メインメモリ内)に退避
= 割り込み処理(←1.4.3項)手順における手順6のソフトウェア状態の退避



(B) プロセススイッチ手順3の“プロセスディスパッチ”におけるメインメモリ内PCBのプロセッサ内PSWへの回復

- スイッチ先のユーザプロセスの新プロセスコンテキストとしてのPCB(メインメモリ内)からのスイッチ先の当該ユーザプロセスのPSW(プロセッサ内)の回復
= 割り込み処理(←1.4.3項)手順における手順8のソフトウェア状態の回復



(上級)

プロセススイッチの実際的な手順 —プロセスコンテキストスイッチ機構の動作の観点 (2)—

- PSW全体も一種のプロセッサ状態
- プロセスコンテキストであるPCBはメインメモリ上に置く

↓ (実際)

■ PCBのスイッチ (= (A)退避と(B)回復)

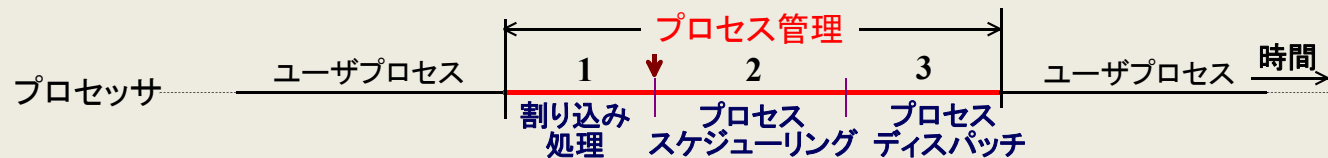
= スイッチ前後の(旧新)PCBを実体とする“PSW(プロセッサ内) ⇔ PCB(メインメモリ上)間転送”

- (A)(B)を併せた“プロセスコンテキストスイッチ”において, (A)での退避先のPCB(旧プロセスコンテキスト)と(B)での回復元のPCB(新プロセスコンテキスト)とが,
 - (a) 同じであれば, 中断しているユーザプロセスの“再開”
 - (b) 異なれば, 相異なるユーザプロセス間での“プロセススイッチ”
- (a)(b)のどちらになるかは, (1) プロセススイッチ手順1を引き起こした割り込み要因が何であるか;
(2) プロセススイッチ手順2でのプロセススケジューリングの結果;によって決定

プロセスの生成

1. プロセス領域の割り付け: メインメモリ上に, プロセス領域を確保
2. PCBの割り付け: メインメモリ上に, PCBの確保
3. PCBの設定: PCBに情報や内容を設定
4. PCBリストの更新: 実行可能キュー(PCBリスト)の最後尾にPCBを挿入
5. プロセス状態の設定: 当該プロセス状態を[実行可能]状態に設定
 - 生成したプロセスは, 当初, 実行可能状態

- プロセス生成の1~5は, 実際には, プロセス管理手順1の割り込み処理と手順2のプロセススケジューリングの間で, OS(カーネル)が実行



(上級)

プロセスの消去(削除) (1)

➤ [実行中/実行可能/実行待ち]状態から遷移

■ 正常終了

1. プロセス領域の解放: 確保&使用しているプロセス領域を空き(未使用)領域としてOSに返却
2. PCBの解放&PCBリストの更新: 実行可能/待ちキュー(PCBリスト)からPCBを削除
3. プロセス状態の設定: 当該プロセスを消去(削除)

➤ 実行中状態プロセスが機能を自然に完了(正常終了)する場合は, その完了時に, 「プロセス機能の終了(完了)」SVC(=ブロック割り込み, ブロック要因)によってOSに通知

(上級)

プロセスの消去(削除) (2)

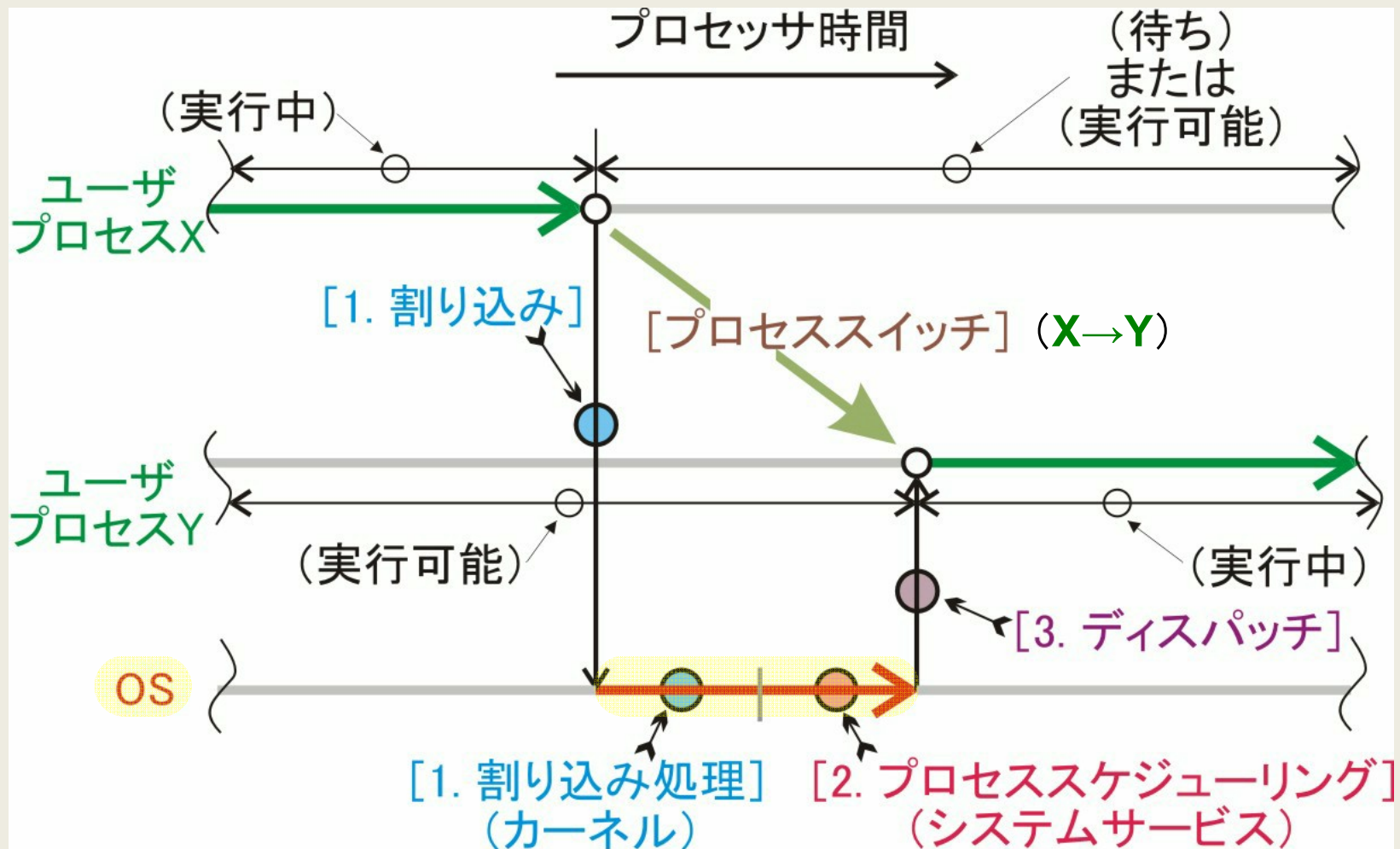
■ 異常終了 (= 中途消滅)

1. PCB&プロセス領域の更新 : PCBやプロセス領域からハードウェア資源に関する情報を削除 (=ハードウェア資源に対する使用要求の強制的な取り下げ)
2. プロセス領域の解放 : 確保&使用しているプロセス領域を空き(未使用)領域としてOSに返却
3. PCBの解放&PCBリストの更新 : 実行可能/待ちキュー(PCBリスト)からPCBを削除
4. プロセス状態の設定 : 当該プロセスを消去(削除)

➤ プロセス消去(削除)の1~3(4)は、実際には、プロセス管理手順1の割り込み処理と手順2のプロセススケジューリングの間で、OS(カーネル)が実行



【まとめ】マルチタスキングとプロセススイッチ(図説)



【まとめ】マルチタスキングとプロセススイッチ —ユーザプロセスX→Yのプロセススイッチ(実際)(0)—

1. 割り込みによる「プロセスX → OS」スイッチ

2. プロセススケジューリングによるスイッチ先プロセスの選定

3. プロセスディスパッチによる「OS → プロセスY」スイッチ

■ マクロ(巨視的)な観点でのユーザプロセス間のプロセススイッチ

↓ (実際には)

= OSを仲介するユーザプロセス間のプロセススイッチ

= 2回の「OS ⇔ ユーザプロセス間スイッチ」によって実現

【まとめ】マルチタスキングとプロセススイッチ —ユーザプロセスX→Yのプロセススイッチ(実際)(1)—

- マルチタスキング → プロセススイッチによって実現

1. 割り込みによる「プロセスX → OS」スイッチ

- 割り込みが受け付けられると、プロセッサ(時間)での実行はユーザプロセスXからOS(カーネル)による割り込み処理に切り替わり、実行中(状態)のプロセスXは**中断**
 - 「プロセスXの**中断**」とは、割り込み要因にしたがって、次のいずれか
 - (a) **ブロック割り込み**の場合: プロセスX自身が、**ブロック**(=実行中状態→待ち状態遷移)して、**実行を中止** → OSがプロセスXからプロセッサ(時間)を譲り受け
 - (b) **ブロック割り込み以外の割り込み**の場合: OSが実行中状態プロセスXの実行を**実行中状態のまま一時中断** → OSがプロセスXからプロセッサ(時間)を**奪取**

【まとめ】マルチタスキングとプロセススイッチ —ユーザプロセスX→Yのプロセススイッチ(実際)(2)—

2. プロセススケジューリングによるスイッチ先プロセスの選定

- OS(システムサービス)が、スケジューリングアルゴリズムによって、スイッチ先のユーザプロセスYを選定
 - 「プロセスYの選定」とは、1の(a)(b)に対応して、
 - (a) 実行可能状態プロセスから1個をプロセスYとして選定
 - (b) 中断している実行中状態プロセスXと実行可能状態プロセスを併せたうちから1個をプロセスYとして選定

【まとめ】マルチタスキングとプロセススイッチ —ユーザプロセスX→Yのプロセススイッチ(実際)(3)—

3. プロセスディスパッチによる「OS→プロセスY」スイッチ

- OS(カーネル)が、「プロセッサ(時間)での実行」を、2で選定したユーザプロセスYに切り替え(実行)
 - 「プロセスYの実行」とは、2の(a)(b)に対応して,
 - (a) 選定したプロセスYをディスパッチ(狭義)(=実行可能状態→実行中状態遷移)
 - (b)
 - (b-1) 中断している実行中状態プロセスX(=Y)を選定した場合、実行中状態を保持しているプロセスXを再開
 - (b-2) 実行可能状態プロセスからプロセスYを選定した場合、中断している実行中状態プロセスXを横取り(=実行中状態→実行可能状態遷移) & 選定したプロセスYをディスパッチ(狭義)(=実行可能状態→実行中状態遷移)