オペレーティングシステム(OS)

柴山　潔

7.　プロセス管理 (1)

- プロセス管理とプロセッサ管理
- プロセスの状態とその遷移
- プロセス管理と割り込み(1)

プロセスとプロセッサ

［定義1.3］（再掲）

■ **プロセス**(process)（タスク(task)）:“実際に実行するプログラム（＝動的に生成するマシン命令列やそれが使用するデータの集まり）”

- **OS**による実行制御対象の論理的な単位
  - ➢ プロセッサでプロセスを実行するためには，プロセスをメインメモリにあらかじめ（＝実行前）置き（＝**プロセス割り付け**），プロセッサとプロセスを対応付けることが必要

■ **プロセッサ**: マシン命令を実行する物理的（ハードウェア）機構

- **OS**による実行制御対象の物理的な単位
  - ➢ メインメモリにあるマシン命令列（＝ **プロセス**）を順次実行（＝ **プロセス実行**）

◆**OS**の担当

(A) メインメモリにプロセスを割り付け ＝[**プロセス割り付け**]

(B) プロセッサにプロセスを割り当て ＝[**プロセススイッチ**（switch; 切り替え），**プロセスディスパッチ**（dispatch; 発送）]

時分割制御

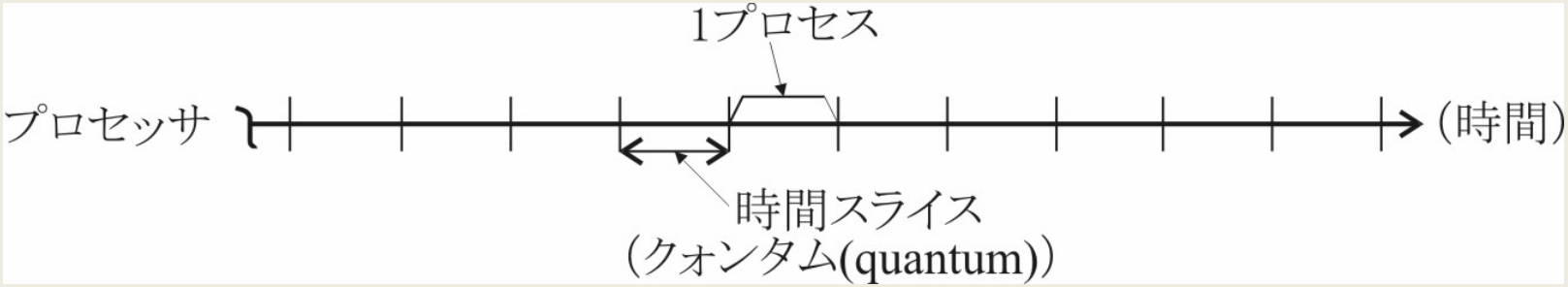
- プロセッサ時間を一定時間（数～数十ミリ秒）ごとに分割，分割した時間単位（→最小単位は時間スライス/クォンタム）をOSが各プロセスの実行に割り当てるプロセッサ管理･制御方式

  ↑

  - （一般の）コンピュータ：プロセッサは単一，一時には唯一プロセスを実行
    ＝ プロセッサの利用時間（＝プロセッサ時間）は1本

■ TSS (Time Sharing System)：時分割制御に従うコンピュータシステム （再掲）

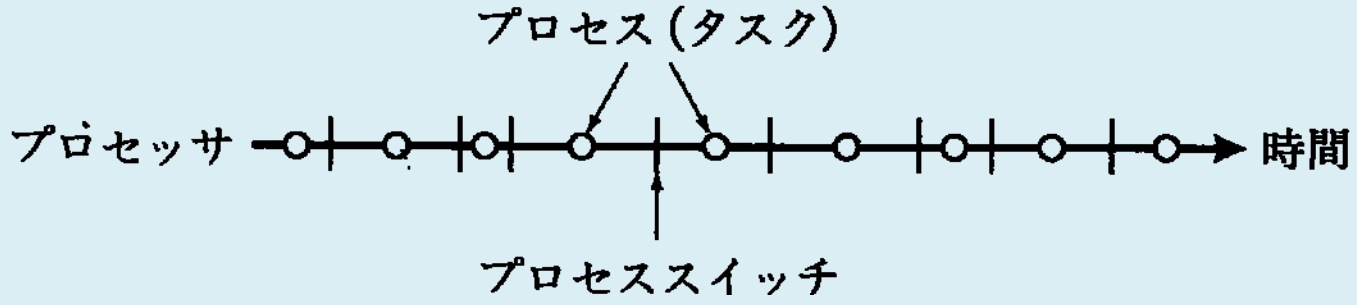
- プロセッサ時間の切り替えは高速 → 人間にとっては“単一ハードウェア（＝プロセッサ）が多数のソフトウェア（＝プロセス）を同時実行”
  - コンピュータシステムでの時間的多重化機能を実現する原理



マルチタスキング

［定義2.1］ **マルチタスキング(multi-tasking)**

- プロセッサとプロセス（タスク）との対応付けを“1プロセッサ対多プロセス”で多重化し，時分割制御によってプロセス（タスク）を切り替えながらプロセッサをはじめとする各種ハードウェア機構を共用する**OS**のプロセッサ管理･制御方式



➢ マルチタスキングの実現

← 多重化したハードウェア機構の効率的利用を**OS**によって時分割制御

- マルチタスキングは**OS**の（時間的管理機能を実現する）原理

← **OS**によるプロセス/プロセッサ管理機能はマルチタスキングを基本に実現

マルチタスキングとマルチプロセッシング（注意）

タスク ＝ プロセス → マルチタスキング ＝ マルチプロセッシング

but, [現代]

■ マルチプロセッシング（multi-processing; 多重処理）: 多数個プロセッサを装備する並列コンピュータなどにおける“多プロセッサ対多プロセス”という多重化

↓ ↑

■ マルチタスキング : 一般のコンピュータ（唯一プロセッサ）における“1プロセッサ対多プロセス”という多重化

マルチタスキングとマルチプログラミング（注意）

- **OS**はコンパイラが実行前に（静的に）生成したプログラム（実際にはマシン命令列）から プロセス（実際にはマシン命令列）を実行時に（動的に）生成

■ マルチタスキング：単一プロセッサ上で複数のプログラムを時分割制御によって切り替え・実行する多重化方式

↑（発展形）

- マルチタスキングの概念が誕生した［第3世代］では，
  ＝ マルチプログラミング（multi-programming; 多重プログラミング）

OSによるプロセス管理機能

(1) プロセスの状態（＝プロセス状態）の管理

(2) 実行プロセスの切り替え　　＝ プロセススイッチ(process switch)

(3) プロセスの生成と消去（削除）

(4) プロセス実行順序の決定　　＝ プロセススケジューリング(process　scheduling)

(5) プロセスからの実行（制御）フロー（＝スレッド(thread)）の生成と管理

(6) 複数プロセスの同期

(7) 複数プロセス間の通信

プロセスの状態（プロセス状態）

➢ **OS**は，プロセスを，生成から消去までの（＝生きている）期間，次の3状態のどれかで管理･制御

(A) **実行可能** (ready)（走行可能）：いつでも（プロセッサで）実行できる状態

(B) **実行中** (running)（走行中，実行(executing)）：現在（プロセッサで）実行している状態

- ➢ ある時刻にプロセッサ（時間）上に存在するプロセスは唯一

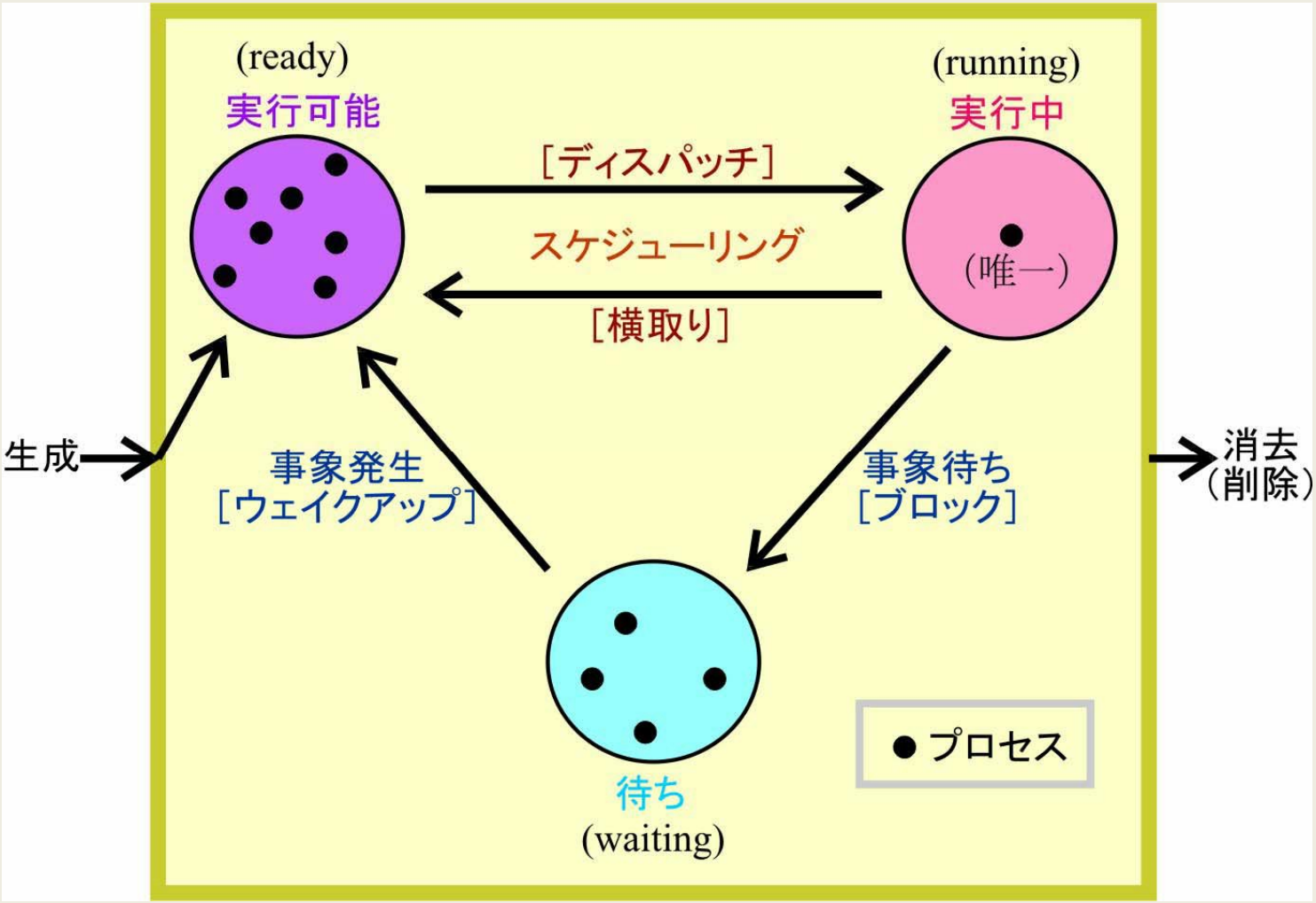
(C) **待ち** (waiting)（閉そく，ブロック(blocked)）：事象の発生を待っている状態

➢プロセスの生成直後は実行可能状態，プロセスの消去は3状態のいずれでも可

■ **OS**による実行（実行中状態に）するプロセスのスケジューリング：プロセスの実行可能(ready) ⇔ 実行中(running) 状態間遷移の管理･制御　（原則）

プロセスの状態と状態遷移（図）

重要



プロセスの状態遷移 —ディスパッチと横取り—

[定義2.2] **ディスパッチ(dispatch)**

- プロセスの**実行可能→実行中**状態遷移

↓ ↑

[定義2.3] **横取り** **プリエンプション(preemption)**

- **OSによる強制的な**(プロセスの)**実行中→実行可能**状態遷移

■ **プロセススケジューリング**

- 実行可能状態のプロセスから1個を選定，それを実行可能→実行中状態遷移させる機能(狭義)
- 実行可能⇔実行中状態遷移(＝「ディスパッチ」とその結果としての「横取り」の総称)(広義)

事象とは？

［定義2.4］ **事象** **イベント(event)**

- プロセスの実行中→待ち状態遷移や待ち→実行可能状態遷移を起こす要因
  - ◆**事象待ち**：実行中→待ち状態遷移（**ブロック**）
  - ◆**事象の発生**：待ち→実行可能状態遷移（**ウェイクアップ**）

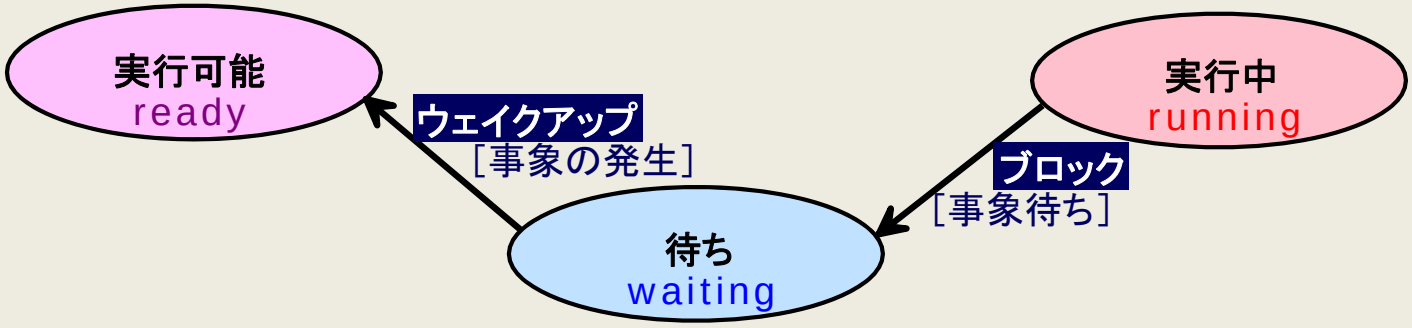
事象によるプロセス状態遷移 —ブロックとウェイクアップ—

［定義2.5］ **ブロック(block)** **事象待ち**

- プロセスの**実行中→待ち**状態遷移
  - 実行中状態のプロセスに関係する事象の発生を待つために行う

［定義2.6］ **ウェイクアップ(wake-up)**

- プロセスの**待ち→実行可能**状態遷移
  - 待ち状態のプロセスが待っている事象の発生による，そのプロセスの待ち→実行可能状態遷移



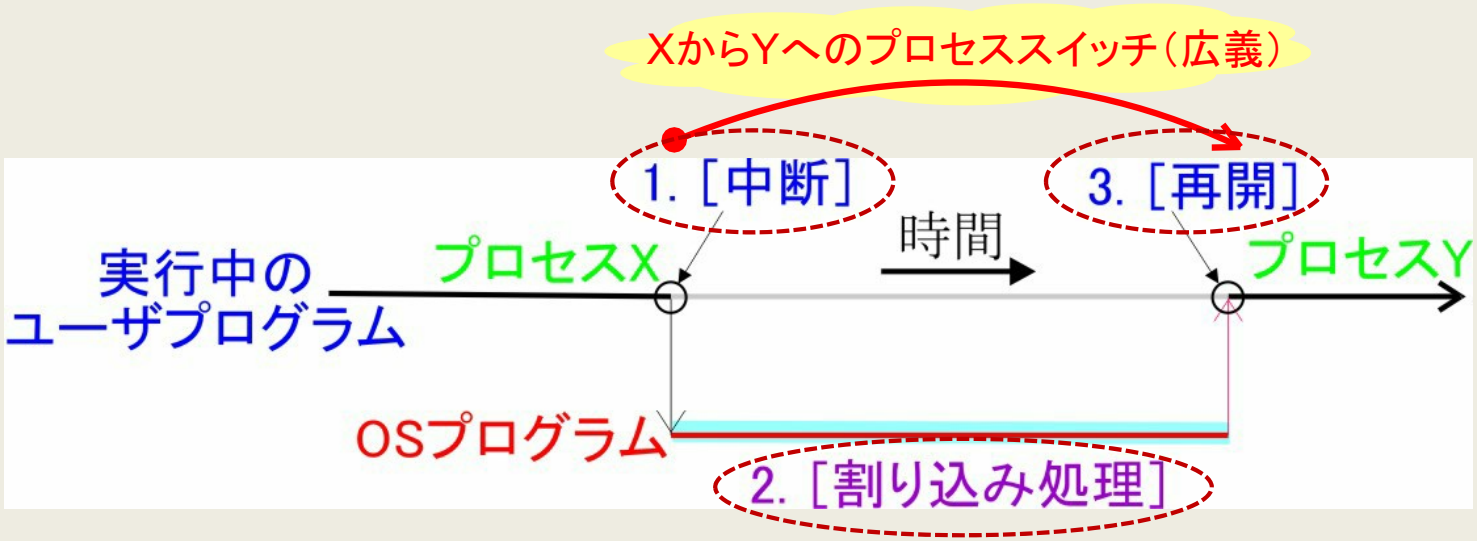
割り込みとユーザプロセスの管理

■ 割り込みとは？
- 「OSプログラムが，OS機能を実行するために，ユーザプログラムからプロセッサ（時間）を強制的かつ動的に奪い取る」仕組み
- 「プロセッサで実行しているユーザプログラムを“割り込み処理”というOSプログラムへ強制的かつ動的に切り替える」仕組み

■ 割り込みとは？ —ユーザプロセスの管理の観点—
- 「あるユーザプロセスから（原則として，別の）ユーザプロセスへの切り替え」の実現
  - ➢ 広義のプロセススイッチ

割り込みによるユーザプロセスの管理手順

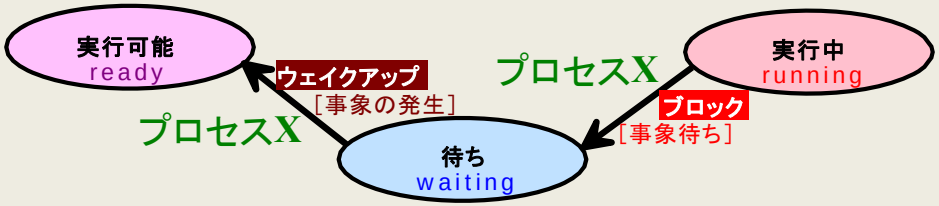
1. 割り込みによるユーザプロセスXの中断：OS（カーネル）が実行中（状態）のユーザプロセス(＝X)からプロセッサ（時間）を奪取し/乗っ取り，そのプロセス(X)の実行を一時中断
2. 割り込み処理：“割り込み処理”というOS（カーネル）プログラム自身を実行
3. ユーザプロセスYの再開：割り込み処理が終了すれば，OS（カーネル）がユーザプロセス(＝Y)にプロセッサ（時間）を譲り，そのプロセス(Y)の実行を開始/再開（原則）



（上級）ブロック（割り込み）/ウェイクアップとユーザプロセス管理

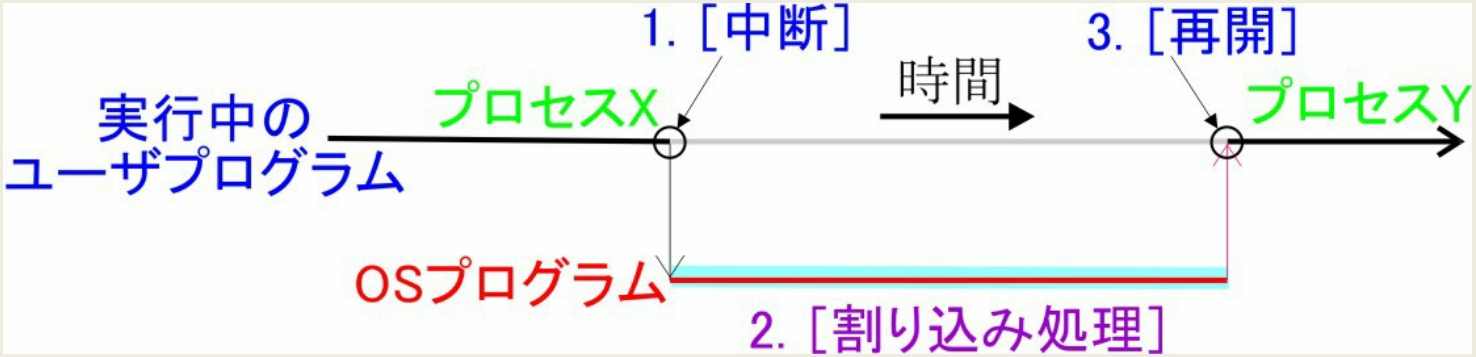
- 管理手順の3.で再開するプロセスYは実行可能状態プロセスから選定（＝プロセススケジューリング）
  → この場合，プロセスY≠プロセスX
- (1)～(3)のブロック割り込み ＝ 内部割り込み
  → 割り込み要因 ＝ 実行中の（＝割り込まれる）プロセスX自身（の実行による(1)～(3)）そのもの）

- プロセスXは「割り込みによるブロック」の要因である“事象待ち”を解く「事象の発生」がなければ，「実行の再開」（実際には，「実行可能状態への遷移」）は不可能
  - 内部割り込みの場合，OSに割り込まれたプロセスXが（実行中状態での中断ではなく）ブロック（＝実行中→待ち状態遷移）する理由
- ブロック割り込みによってブロック（＝実行中→待ち状態遷移）する場合における「ブロックしているプロセスXの再開」とは，正確には，「プロセスXのウェイクアップ（＝待ち→実行可能状態遷移）」



割り込みによるユーザプロセスの切り替え

1. プロセス**X**の中断（ → **2.** 割り込み処理 → ）**3.** プロセス**Y**の再開



- 割り込みによるユーザプロセスの管理手順（1～3）において，OSは「ユーザプロセスの切り替え（＝広義のプロセススイッチ）機能」も実行

↓

  - 手順3において再開するプロセスYが，手順1において中断するプロセスXと
    (A) 同じ(X＝Y) → プロセスX の再開
    (B) 異なる(X≠Y) → プロセスYのディスパッチ（＝“プロセスX→プロセスY”のプロセススイッチ）

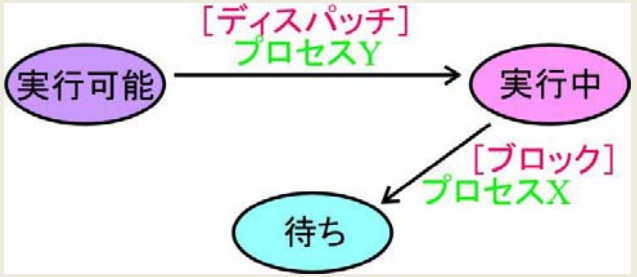
プロセス状態遷移の管理の実際
—プロセススイッチ（広義）の場合—

1. プロセスXの中断（→ 2. 割り込み処理 →）3. プロセスYの再開
（(A) プロセスX の再開：プロセスX(＝Y)が実行中状態を保持したまま再開）
(B) プロセスYのディスパッチ：プロセスX→プロセスYという実行中状態プロセスの切り替え
＝広義のプロセススイッチ

■ プロセススイッチに伴うプロセスXとプロセスYの状態遷移（実際）

(1) [ブロック & ディスパッチ]
- プロセスX：実行中→待ち状態（＝ブロック）
- プロセスY：実行可能→実行中状態（＝ディスパッチ）
  ＝ 実行中状態プロセスがX→Y（切り替え）

(2) [横取り & ディスパッチ]
- プロセスX：実行中→実行可能状態（＝横取り）
- プロセスY：実行可能→実行中状態（＝ディスパッチ）
  ＝ 実行中状態プロセスがX⇔Y（入れ替え）



➢ 実際のこれらの「プロセス状態（遷移）の管理」そのものは，割り込み処理の終了時に実行

【まとめ】割り込みによるプロセススイッチ（実際）

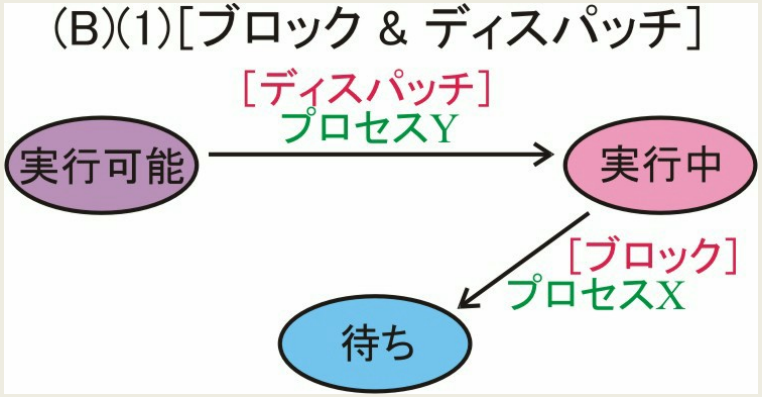
■ 割り込みによるプロセススイッチで実行中状態になるプロセスは？

(A) 中断しているプロセスXの再開

(B) （広義の）プロセススイッチ（プロセスX→プロセスY）の実行

(1) [ブロック（プロセスX） & ディスパッチ（プロセスY）]

(2) [横取り（プロセスX） & ディスパッチ（プロセスY）]



（上級）

中断プロセスの再開とプロセススイッチ（注意）

（厳密には）

(A) 中断しているプロセスXの再開
(B) プロセススイッチ（プロセスX→プロセスY）の実行

（しかし，）

- ユーザプロセス→OS（割り込み処理）→ユーザプロセス
  ＝ 実行中状態プロセス（OSも含めて）というプロセッサ（時間）の使用者の切り替え（スイッチ）
  ＝（広義の）プロセススイッチ機能は，(A)も(B)も，OS機能として同じ
  &
- それを実現するプロセススイッチ機構は共用

■ プロセススイッチ：(A)&(B)を統一的に実現する機能や機構

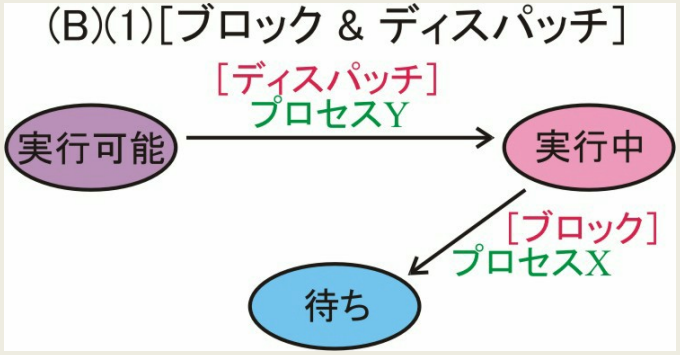
（上級）

割り込みとプロセススケジューリング (1)
—（相異なるプロセス間）プロセススイッチの場合—

- **(B)(1)**[ブロック&ディスパッチ]，**(B)(2)**[横取り&ディスパッチ]とも，割り込み処理としてのプロセス管理手順の3「割り込み処理の終了時におけるプロセス管理」では，「実行可能状態プロセスのうちのどれをディスパッチ（＝実行可能→実行中状態遷移）させるか」＝「プロセススイッチ先の決定」が必要

■ **プロセススケジューリング（狭義）**：実行可能状態プロセスからのディスパッチするプロセスの選定

■ プロセススケジューリング（広義）：ディスパッチ機能とその結果としての横取り機能とを併せた**OS**機能
（参考，再掲）



（上級）

割り込みとプロセススケジューリング (2)
―プロセス再開の場合―

■ プロセスXが「ブロック（実行中→待ち状態遷移）しない（＝(B)(1)でない）」&「プロセススケジューリングが横取り（実行中→実行可能状態遷移）を許さない（横取りなしor横取り不可能）（＝(B)(2)でない）」の場合（＝(B)でない）

↓

➢ プロセスXは（中断するが）実行中状態から遷移せず
- プロセス状態遷移は(A)

↓

➢「割り込み処理の終了時におけるプロセス管理」では，「実行中状態で中断しているプロセスXを再開」

実際には，
- (A)は「プロセスXの再開」
  - ➢「プロセススイッチ先の決定」というプロセススケジューリング（狭義）と「実行可能状態のプロセスYを実行中状態にする」というプロセスディスパッチはいずれも不要

（上級）中断プロセスの再開とプロセススケジューリング（注意）

(A) 中断しているプロセスXの再開
(B) プロセススイッチ（プロセスX→プロセスY）の実行

(A)
- プロセススケジューリング（狭義）によるプロセスXの選定
  &
- プロセスディスパッチによるプロセスXの実行

↓

➢ **(A)**「プロセスXの選定・実行」; **(B)**「プロセスYの選定・実行」; ともに「ユーザプロセスの選定・実行」
＝ プロセススケジューリング（広義）

【まとめ】割り込みによるプロセス管理手順（概要）(1)
―割り込み処理も含めたプロセス管理フロー―

1. **割り込み処理**：実行中状態プロセスからOS（カーネル）に切り替え，割り込み要因に対する処理を開始 ＝プロセススイッチ（広義）の開始
   - ➢ カーネル（OS機能）が実行
2. **プロセススケジューリング（狭義）**：実行可能状態プロセスのうちから実行するプロセスを選定（原則）
   - ➢ プロセススケジューラというシステムサービス（OS機能）が，それぞれ実行
3. **プロセスディスパッチ**：2で選定したプロセスを実行可能→実行中状態遷移　＝プロセススイッチ（広義）の終了
   - ➢ カーネル（OS機能）が実行

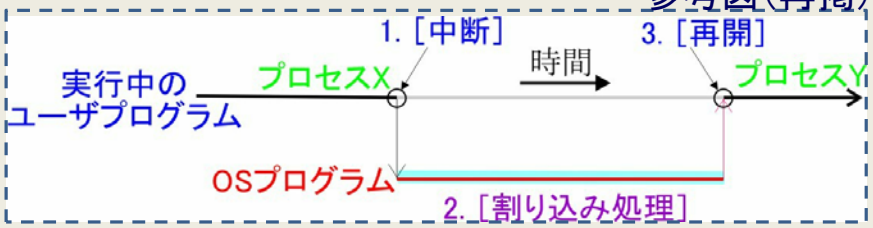
プロセススイッチ（広義）　＝ 1～3

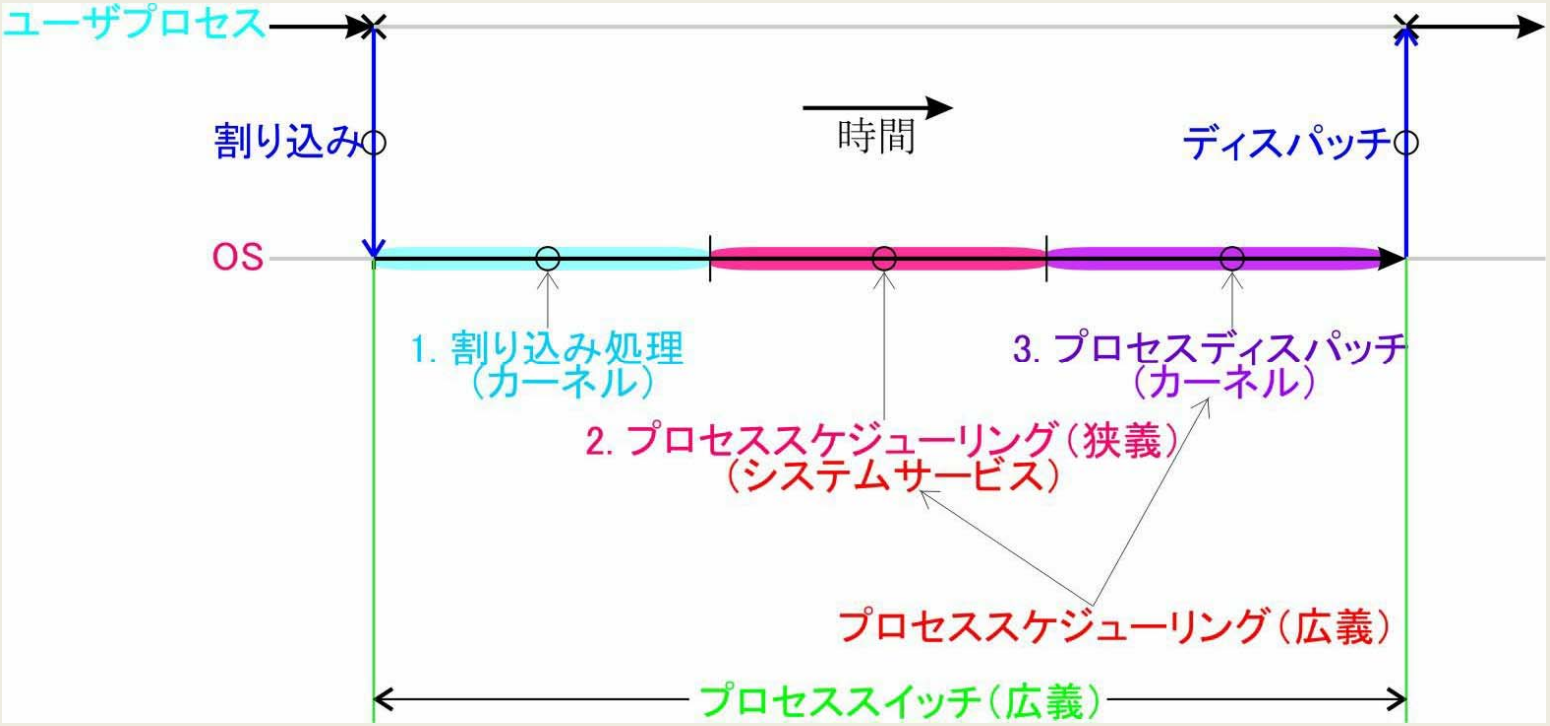
（例外）

- ●「割り込まれたプロセスへの復帰は行わない」割り込み（例：ハードウェア障害，リセット）では，1の割り込み処理で普通は「OSのリブート」を開始 → 1～3 のプロセス管理手順（特に，2と3）は実行せず

【まとめ】割り込みによるプロセス管理手順（概要）(2)
―割り込み処理も含めたプロセス管理フロー（図説）―

参考図（再掲）





（上級）【まとめ】割り込みによるプロセス管理手順（概要）(3)
—プロセス状態遷移との関係— (1)

(1) [手順1（割り込み処理）において]

- ブロック割り込み（＝ブロックを引き起こす割り込み）の場合に
  → ブロック（事象待ち）
- ウェイクアップ割り込み（＝ウェイクアップを引き起こす割り込み）の場合に
  → ウェイクアップ（事象発生）

---

(2) [手順2（プロセススケジューリング）において]

- 横取りがある場合に → 横取り

---

(3) [手順3（プロセスディスパッチ）において]

- 「中断している実行中状態プロセス」がない場合に → ディスパッチ
- 手順2において横取りがあった場合に → ディスパッチ

（上級）【まとめ】割り込みによるプロセス管理手順（概要）(4)
—プロセス状態遷移との関係— (2)（注意）

- 実際の「プロセス状態（遷移）の管理（※）」機能は，OS（カーネル）がプロセス管理手順1（割り込み処理）と2（プロセススケジューリング）との間で実行

  （※）の例
  - ◆ ブロックによる当該プロセスの実行中→待ち状態遷移 (1)
  - ◆ ウェイクアップによる当該プロセスの待ち→実行可能状態遷移 (1)
  - ◆ プロセスの生成や消去に伴うプロセス状態（遷移）

- (1)において，割り込まれた実行中状態プロセスがブロックしない（＝ブロック割り込みでない）場合，OSは，当該実行中状態プロセスを一時中断 → 1～3の手順を遂行
  - ➢ ただし，さらに，プロセス管理手順2でのプロセススケジューリングが横取りを許さない（＝横取りなし）場合には，(2)での横取りや(3)でのディスパッチは実質的には実行せず →「一時中断している実行中状態プロセスを再開」でプロセス管理手順1～3 を終了

（注意） （再掲）
- 再開もプロセスディスパッチに含める