

プロセス管理 (Process Management)

- プロセス (Process, タスク: Task) とは何か
- プロセスの状態について
- プロセスの状態遷移について
- 割り込み (SVC割り込み, 入出力完了割り込み, タイマ割り込み) とプロセス状態遷移の関係
- プロセスの構造, プロセスとスレッド, 関連する話題

プロセス(タスク)とは

- **プログラム**: 計算機で処理して欲しい記述.
プロセス: 計算機内部でプログラムを実行する**能動的な実体**(entity*).
- **プロセスは資源割当の対象**(資源を割り当てる単位).
- プロセスは, プログラムの処理に対応し, その動作はプログラムの記述内容に従う. **仮想的なCPU**(すなわち, **プログラムを実行するための専用の概念的CPU**)を含む.
- **1つのプログラム**は, 計算機内で**1つ以上のプロセス**に変換される(プログラムが並行処理を記述できるので).
- **仮想的なCPU**とプロセスの概念とを明確に分離して, プロセスとは別に, 生成・制御できる実体:**スレッド**(Thread)として提供される場合もある. —最近の傾向(最近のOSやJava仮想マシンなど)

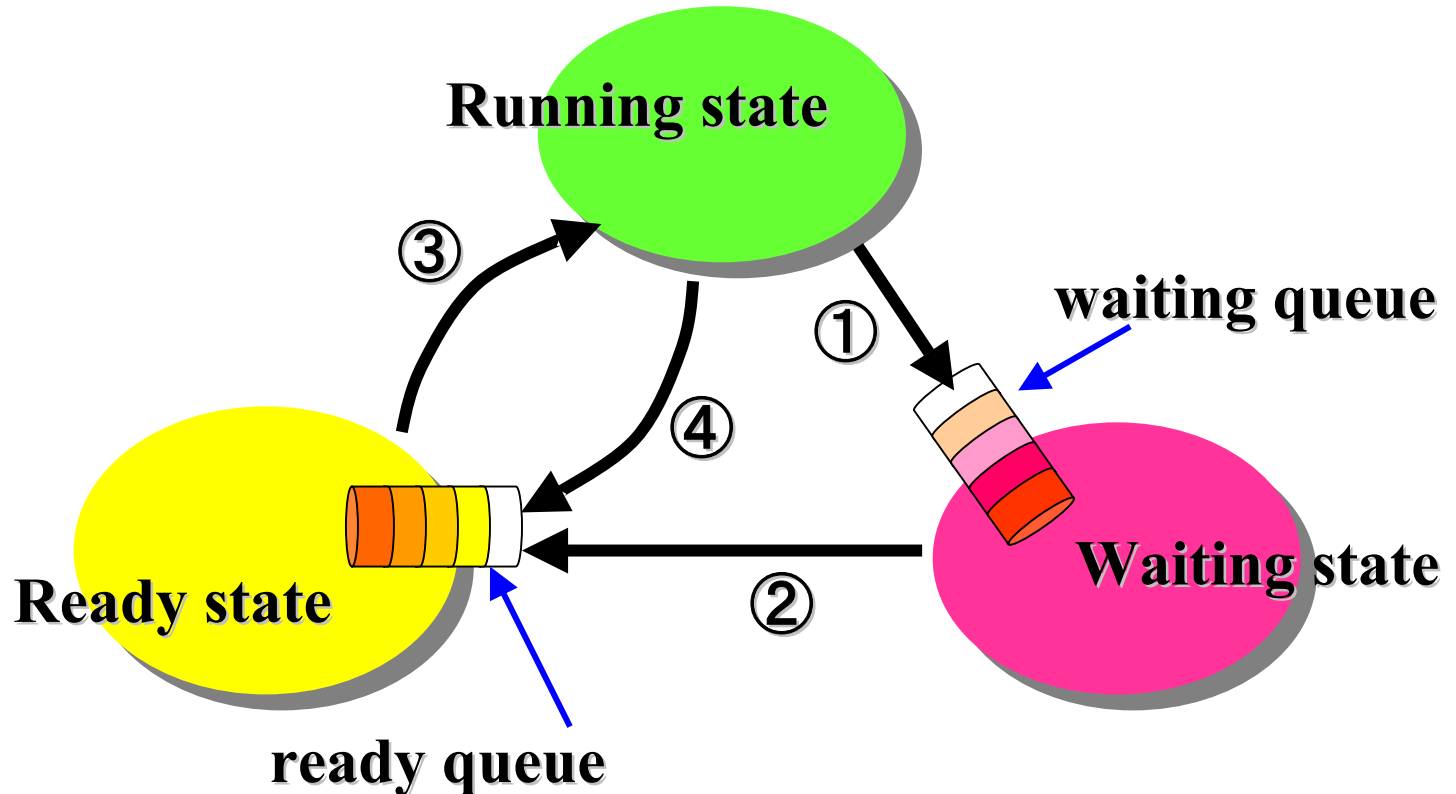
*entity: something that exists as a single and complete unit

プロセス状態とは

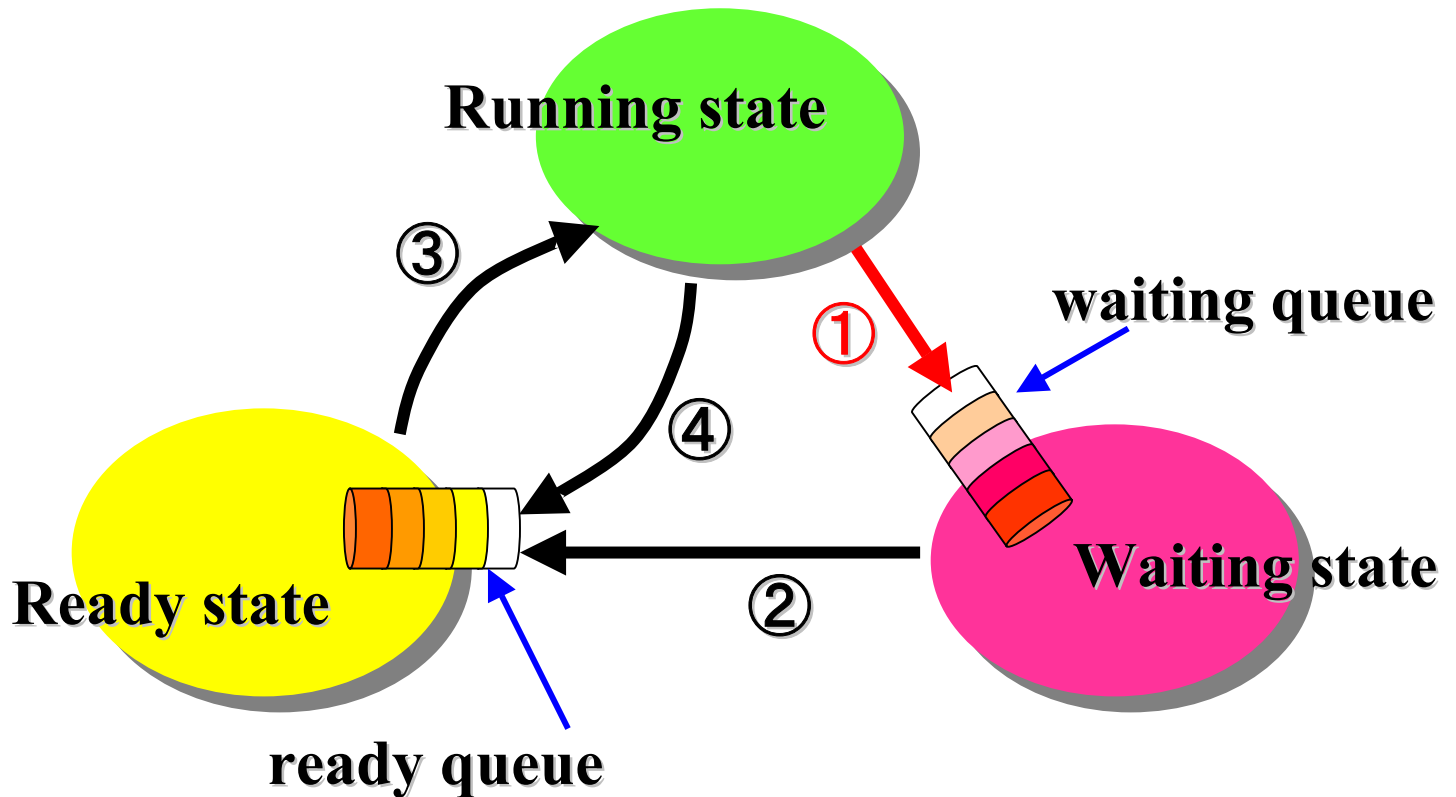
- **実行状態** (running state): 実際にプロセスがプログラムを実行している状態. 実行状態のプロセス数はCPUの数を超えることはない.
- **実行可能状態** (ready state): CPUの使用権を得れば, プログラムを実行できる状態. 実行可能なプロセスは, 一般に, 複数存在し, これらは待ち行列 (**実行可能待ち行列**: ready queue) を形成する. **スケジューラ** (scheduler) は優先順位に基づき, 実行可能待ち行列中のプロセスを実行すべき順に並び替える. **ディスパッチャ** (dispatcher) はスケジューラが選択したプロセスにCPU使用権を割り当てる.
- **待ち状態** (waiting state): 入出力の完了や他のプロセスとの同期など, システムに発生する何らかの事象 (event) や資源の割当を待っている状態. プロセスの実行は継続できず, CPUを必要としない状態. 一般に, 待ち状態のプロセスは複数存在し, **待ち行列** (waiting queue) を形成する.

プロセスの状態遷移とは

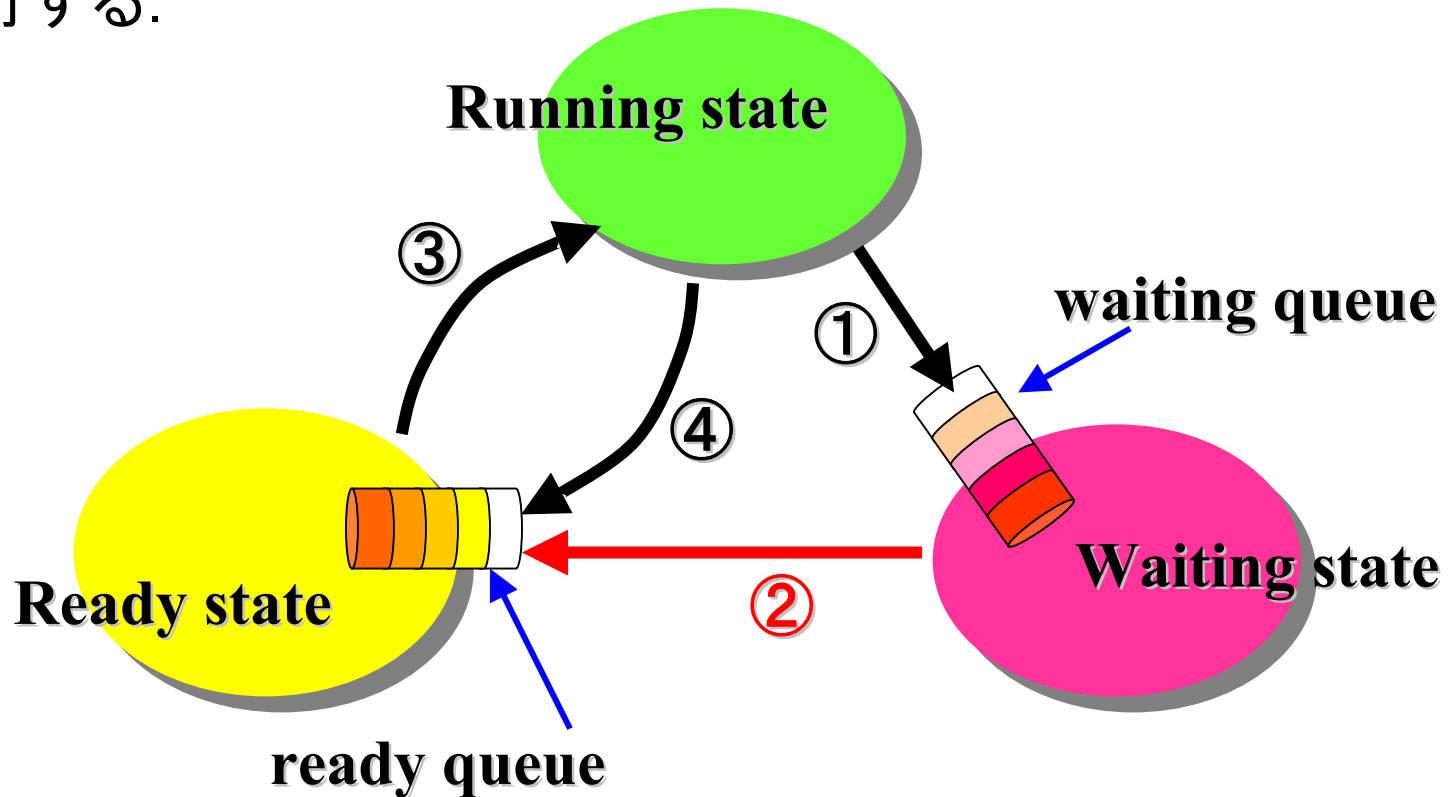
- ① 実行状態から待ち状態への状態遷移
- ② 待ち状態から実行可能状態への状態遷移
- ③ 実行可能状態から実行状態への状態遷移
- ④ 実行状態から実行可能状態への状態遷移



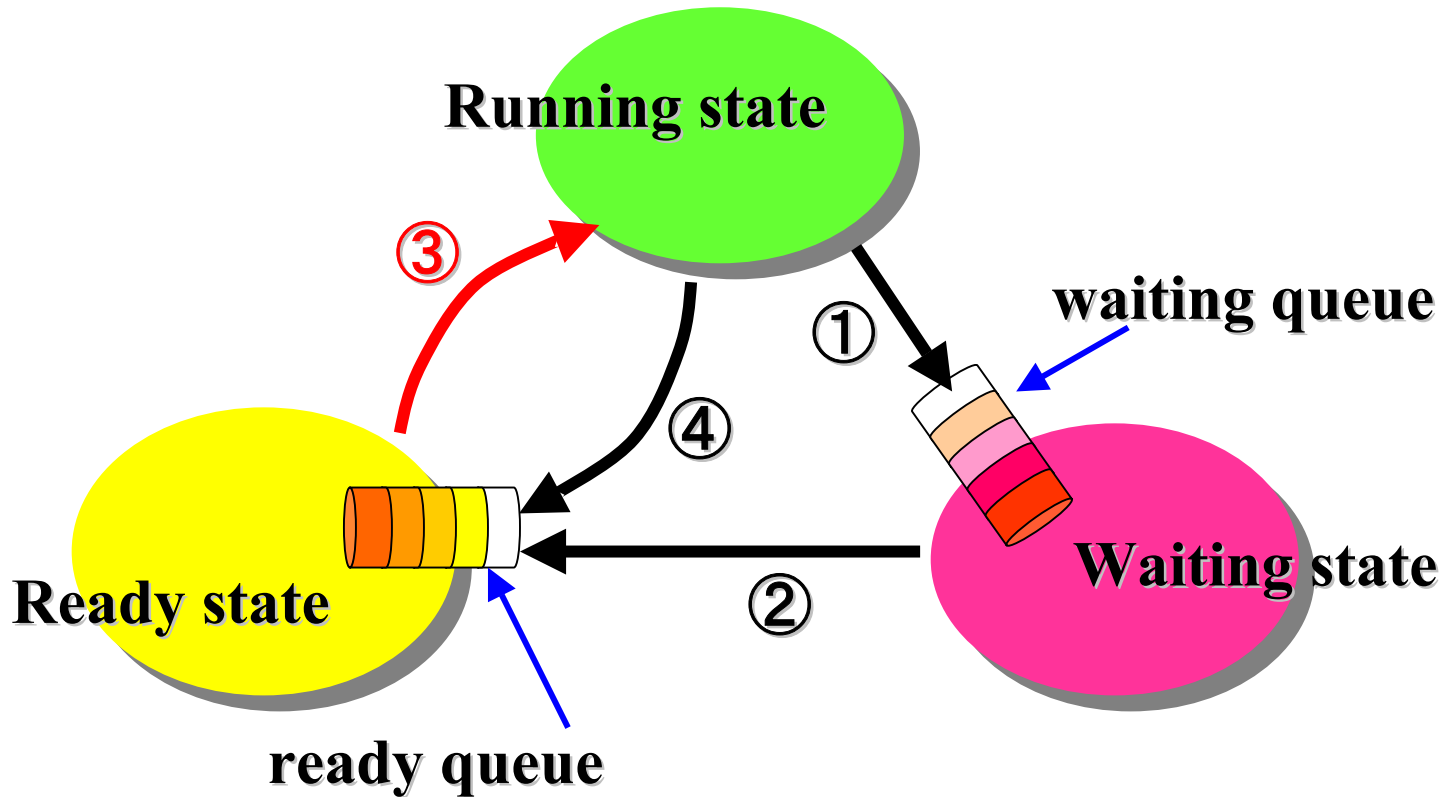
- ① **実行状態** (running state) から **待ち状態** (waiting state) への状態遷移: 実行状態のプロセスが, 例えば, 入力要求をOSに通知するためSVC割込みを発生し, CPU使用権を放棄して, 待ち状態に移行する.



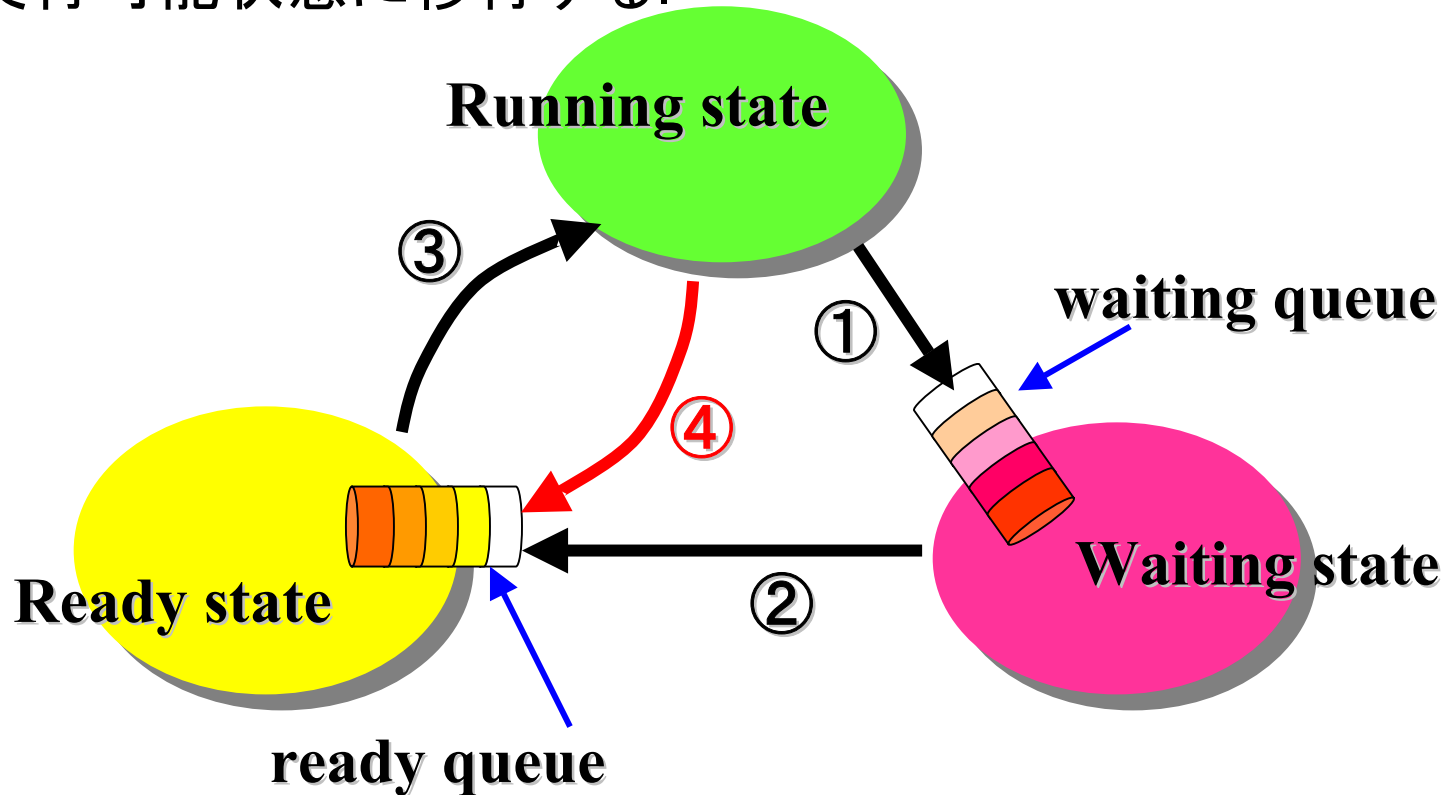
- ② **待ち状態** (waiting state) から **実行可能状態** (ready state) からへの状態遷移: 待ち行列 (waiting queue) にある待ち状態のプロセスが, 入力完了など事象や資源の割当を待機している状態から, 資源を得たり (完了割込みでの通知), 事象の発生を受けて, 実行可能状態に移行する.



③ **実行可能状態** (ready state) から **実行状態** (running state) への状態遷移: 実行可能待ち行列 (ready queue) にある実行可能状態のプロセスが, **スケジューラ**によって優先順位の1位を与えられると, **ディスパッチャ**によって CPU使用権を与えられ, 実行状態に移行する.



- ④ **実行状態** (running state) から **実行可能状態** (ready state) への状態遷移: 実行状態にあるプロセスが, より優先順位の高いプロセスが実行可能状態になると, OSによってCPU使用権を取り上げられ(これを**横取り**: **preemption**と呼ぶ), 実行可能待ち行列 (ready queue) に追加され, 実行可能状態に移行する.

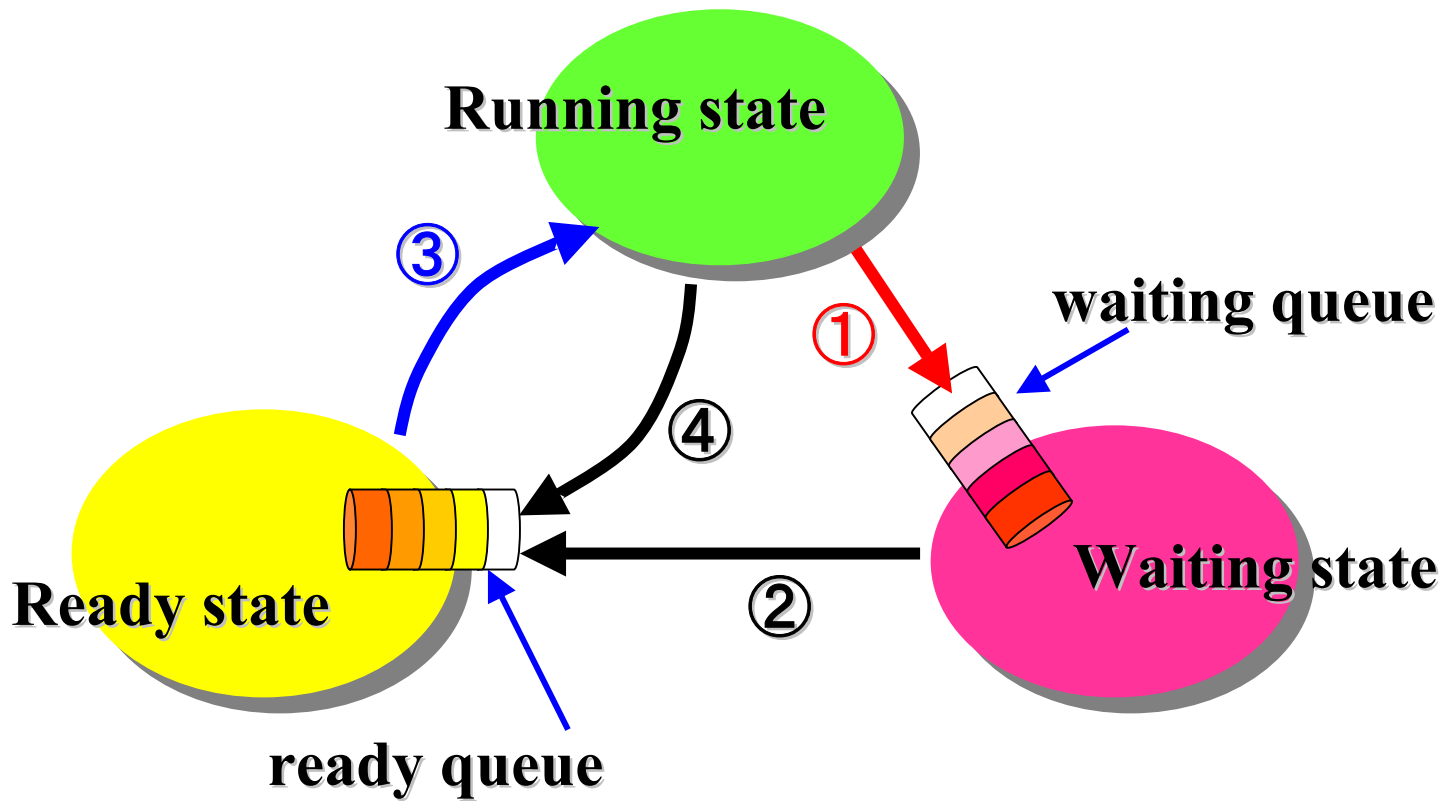
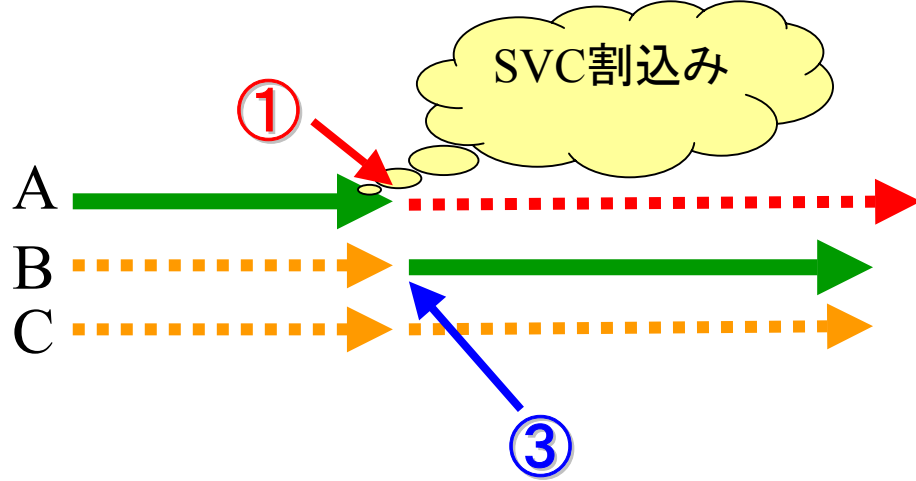


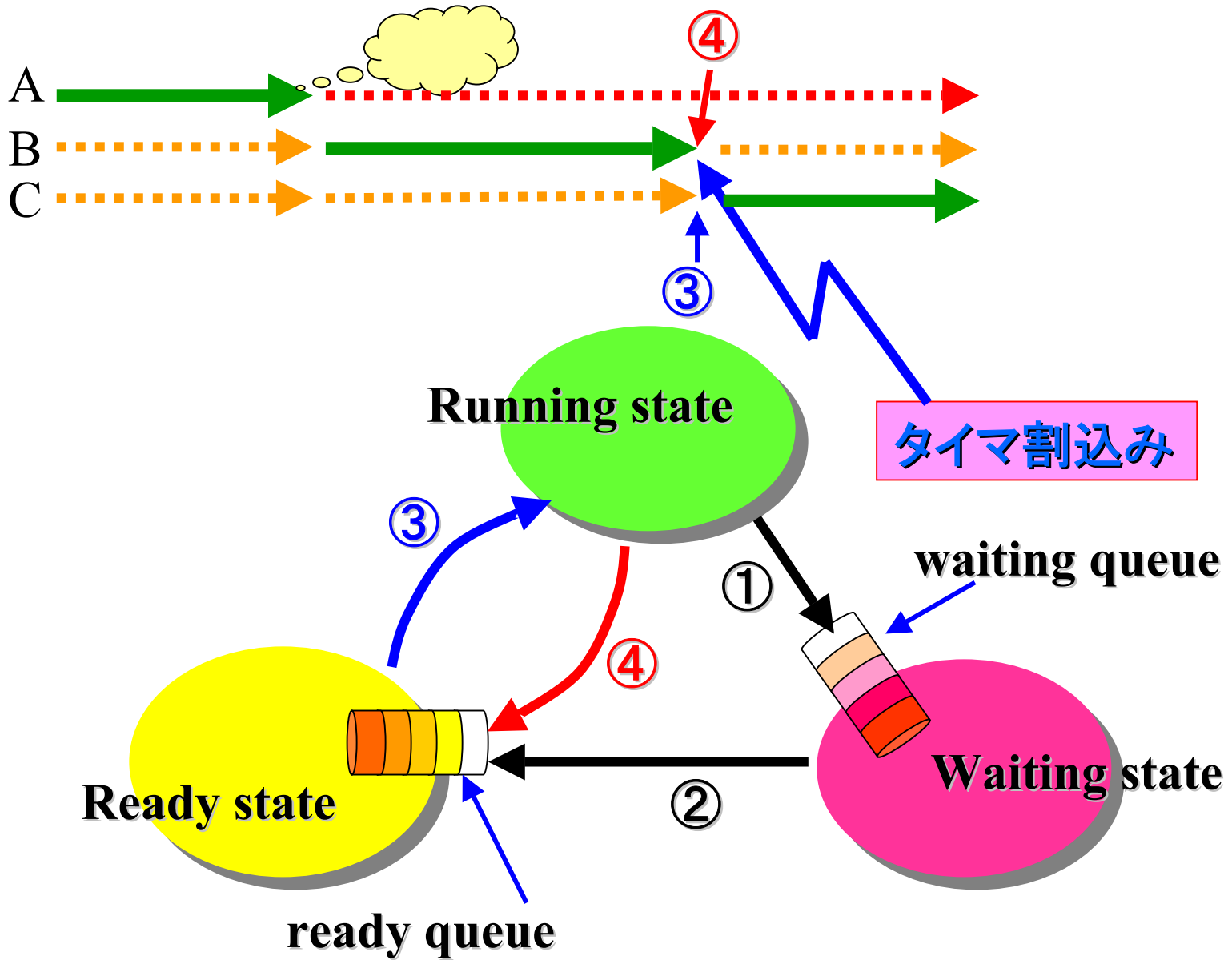
プロセスの状態遷移と割込みの関係

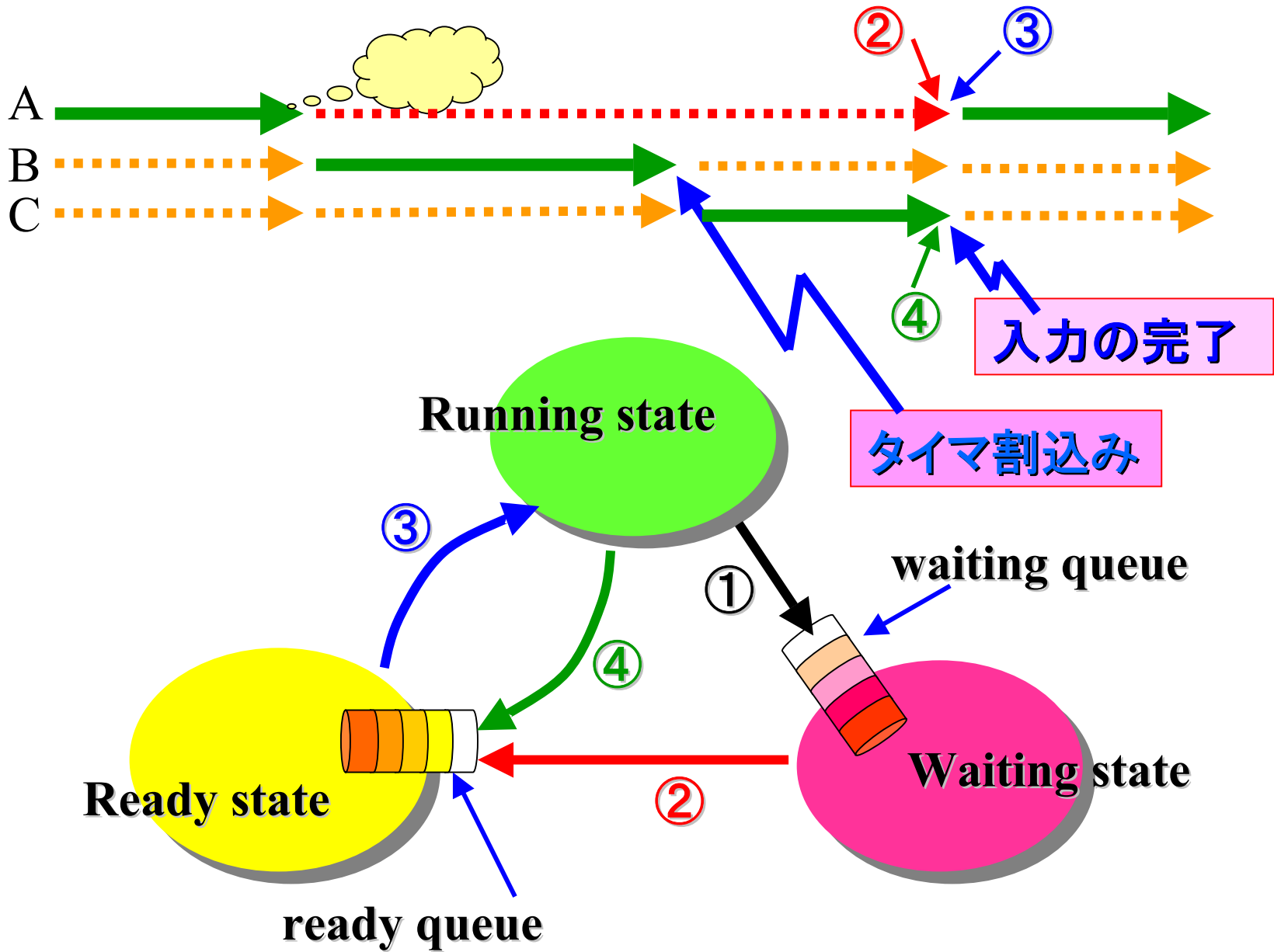
- **プロセスの状態遷移を誘発する割込み**: SVC割込み, 入出力(完了)割込みおよびタイマ割込みによってどのようなプロセス状態遷移が生じるか?
- **プロセスの一生**(プロセスの生成から消滅まで): プロセス状態の変化(含む遷移)を概観して, プロセスの動きを知り, プログラムが計算機内でどのように処理されるかを理解する.

割込みとプロセス状態遷移 との関係

割込みとプロセス状態遷移
を同時に眺めると...



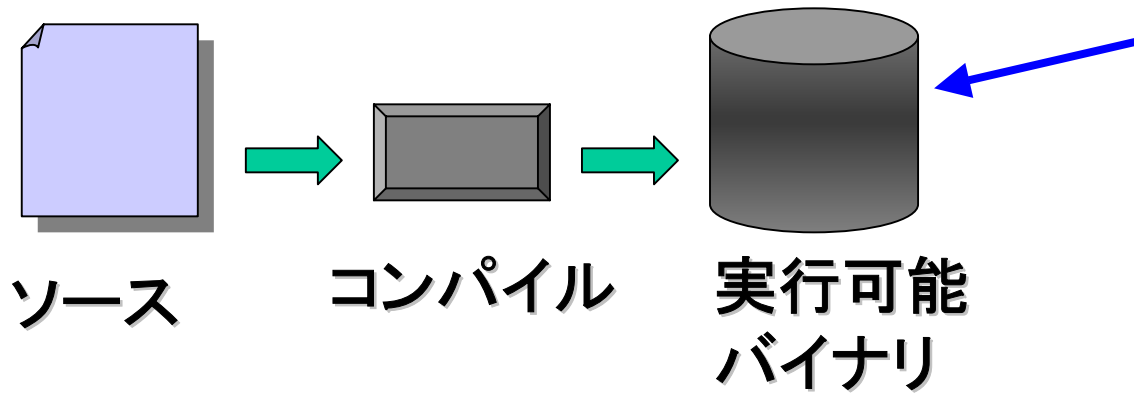




プロセスの一生(プロセス生成からプロセス消滅まで)

あるプロセス*について注目し, そのプロセス状態遷移を見守ると...

*1入力, 1出力のシングルタスク処理(1つのタスク: プロセスを有する)プログラムと仮定する

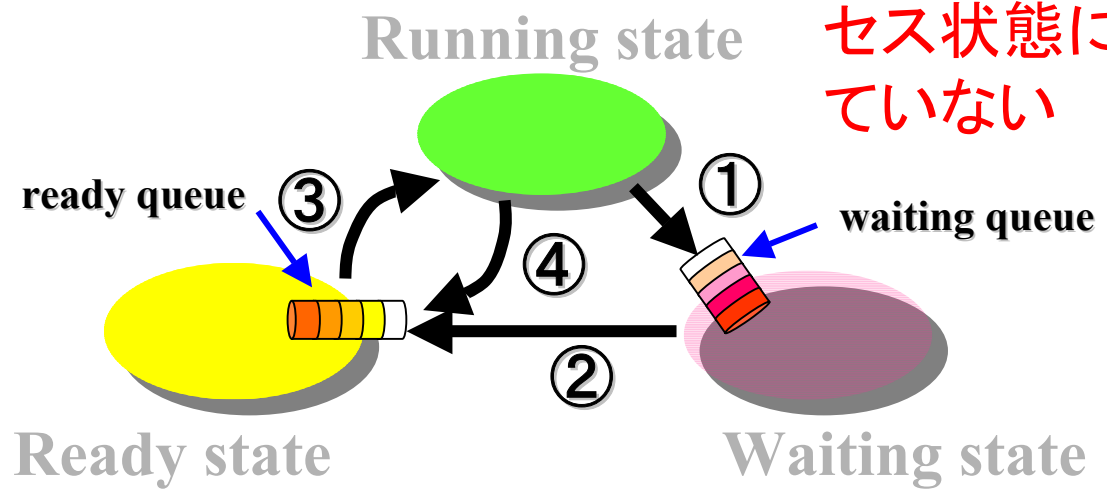


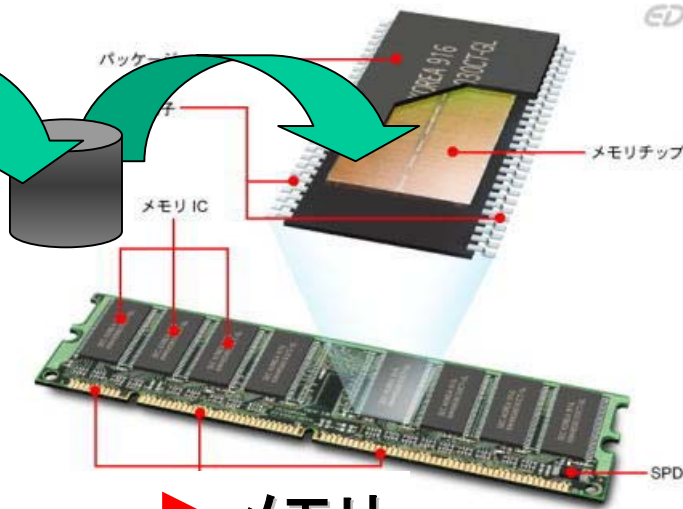
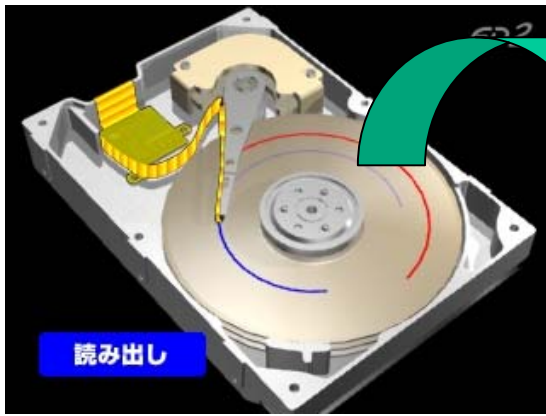
まだディスクの中で、メモリにロードされていない



実行可能バイナリはメモリ上ではないので、どのプロセス状態にも属していない

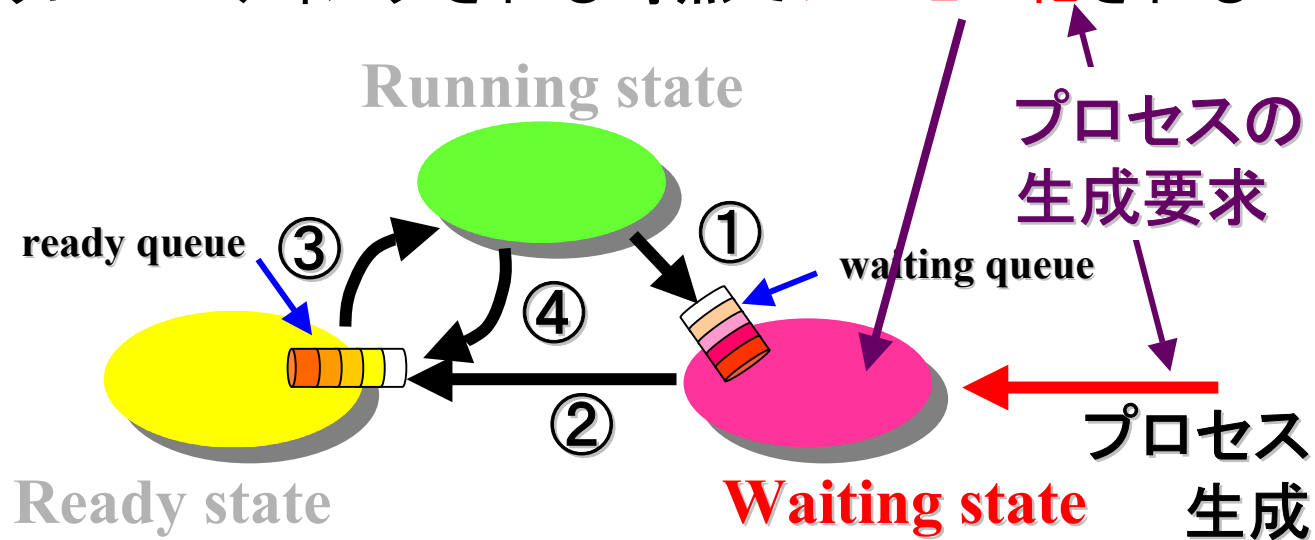
実際はもう少し複雑な過程のなりますが...

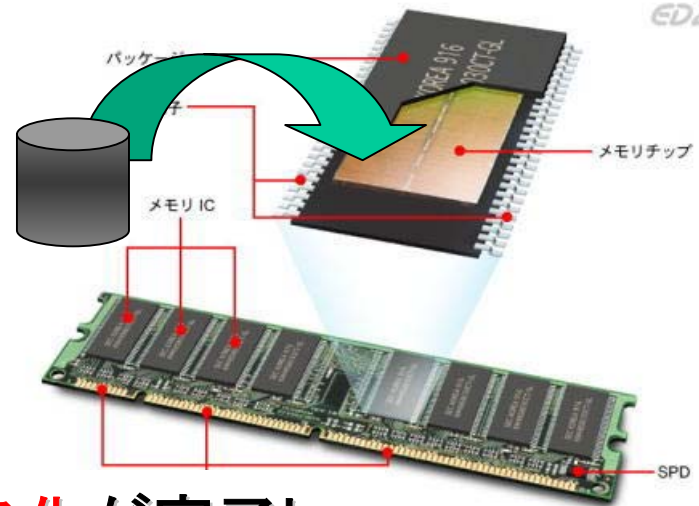




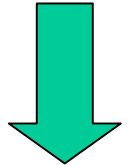
実行バイナリ: ディスク → メモリ

プログラム: メモリにローディングされる時点でプロセス化される

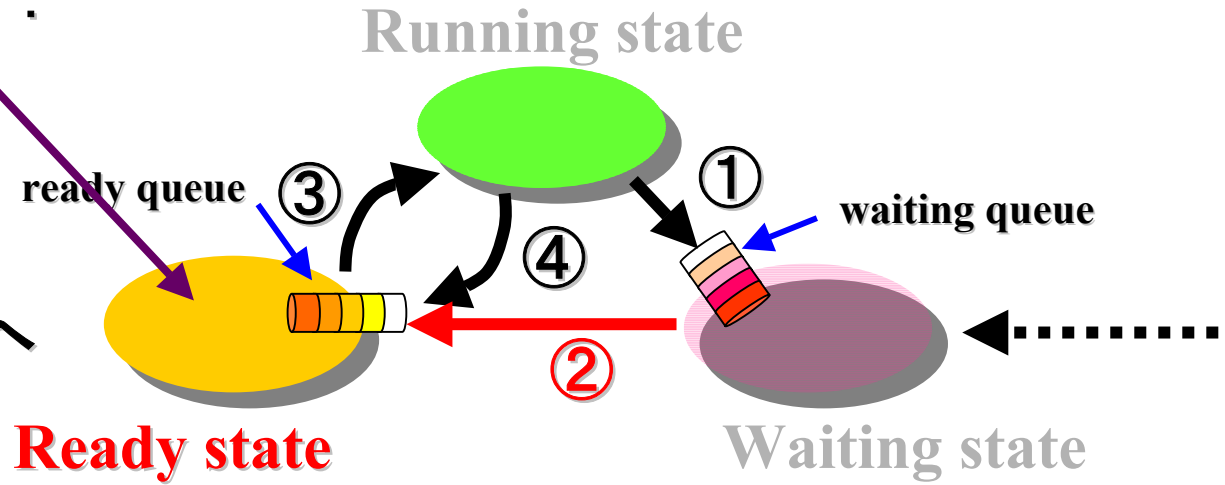


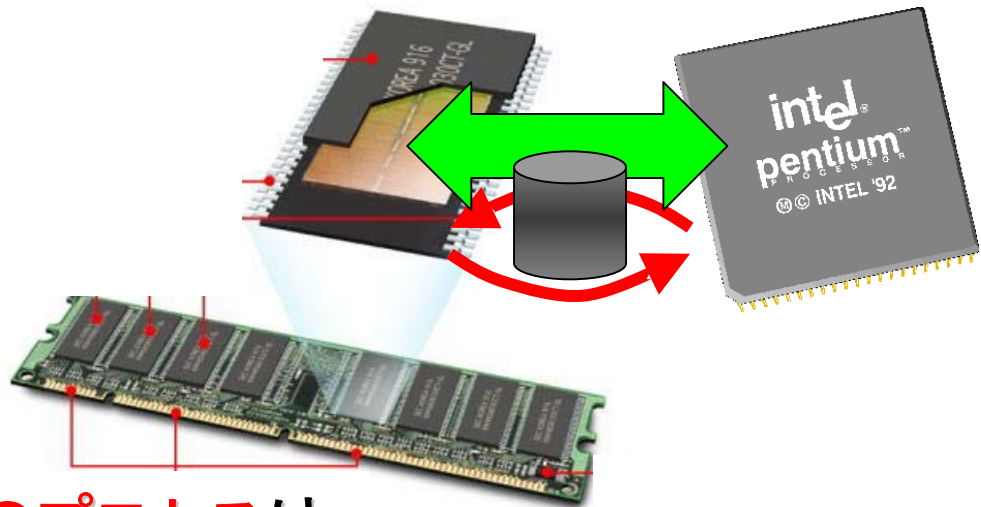


プログラム: プロセス化が完了し,
メモリ上でCPU以外の資源を確保すると.....

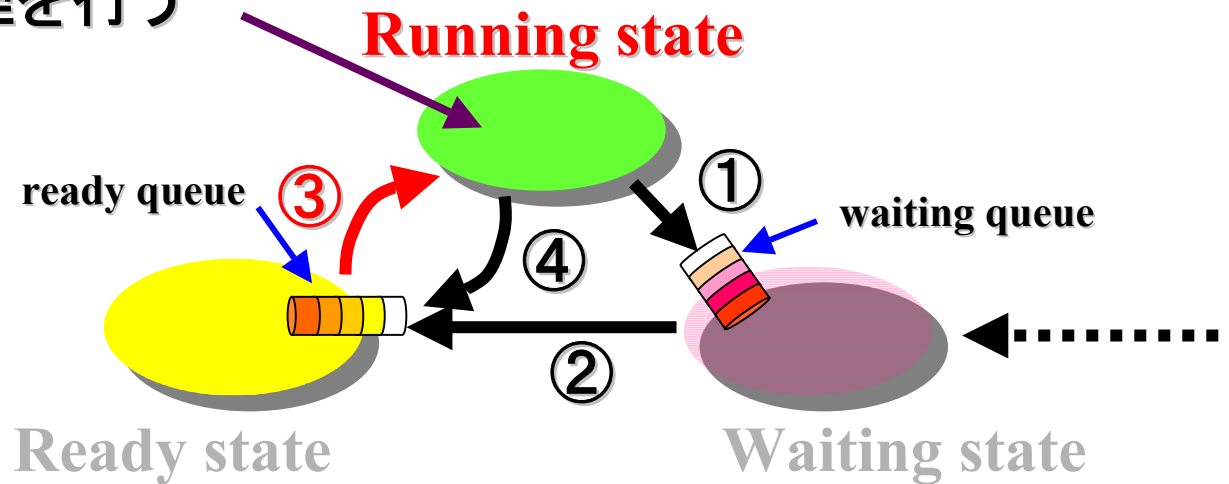


プロセスは
実行可能状態へ



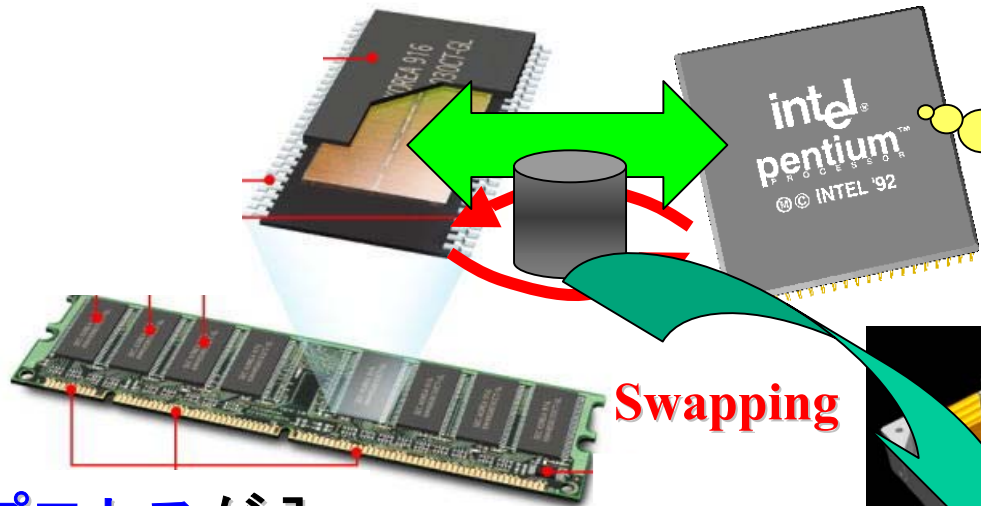


実行状態のプロセスは
メモリ上にあつてCPUを
占有して処理を行う



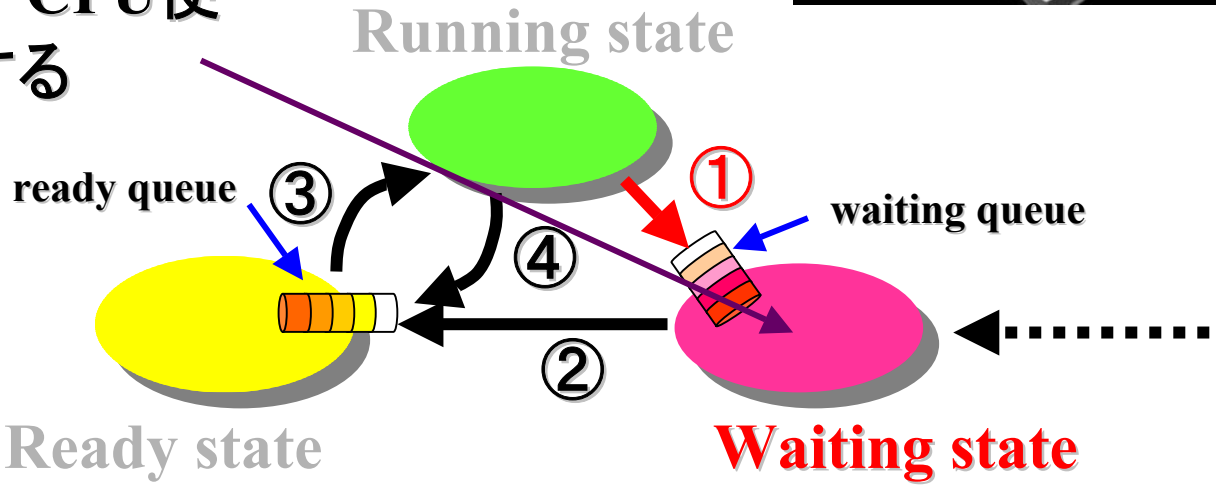
SVC割込み

入力要求



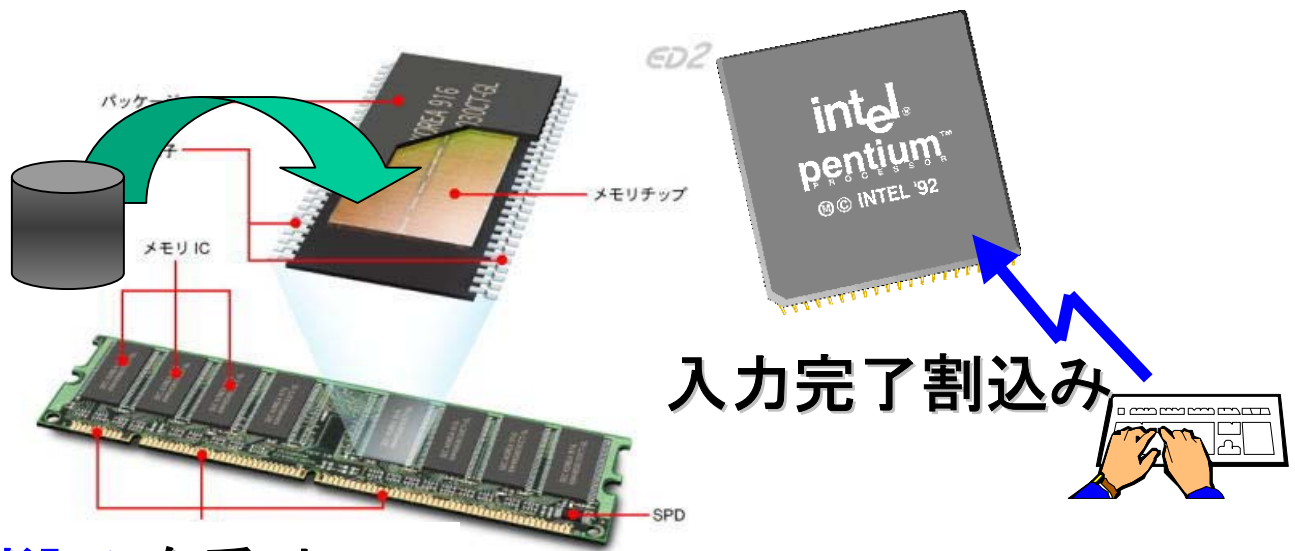
実行状態プロセスが
入力要求(SVC割込み)を
発することで、CPU使
用権を放棄する

待ち状態へ
遷移する

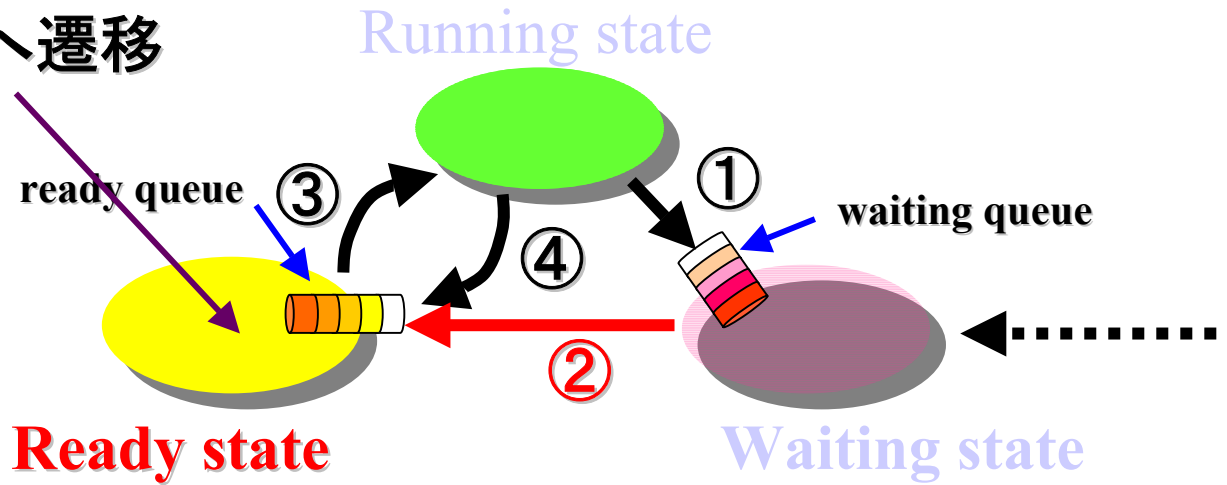


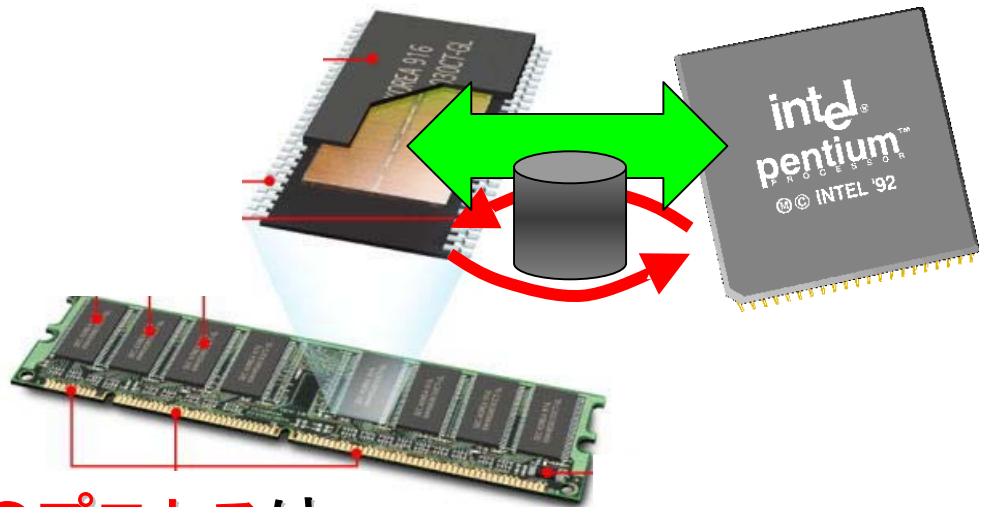
メモリの状況によってディスク
に退避される場合あり

Swapping

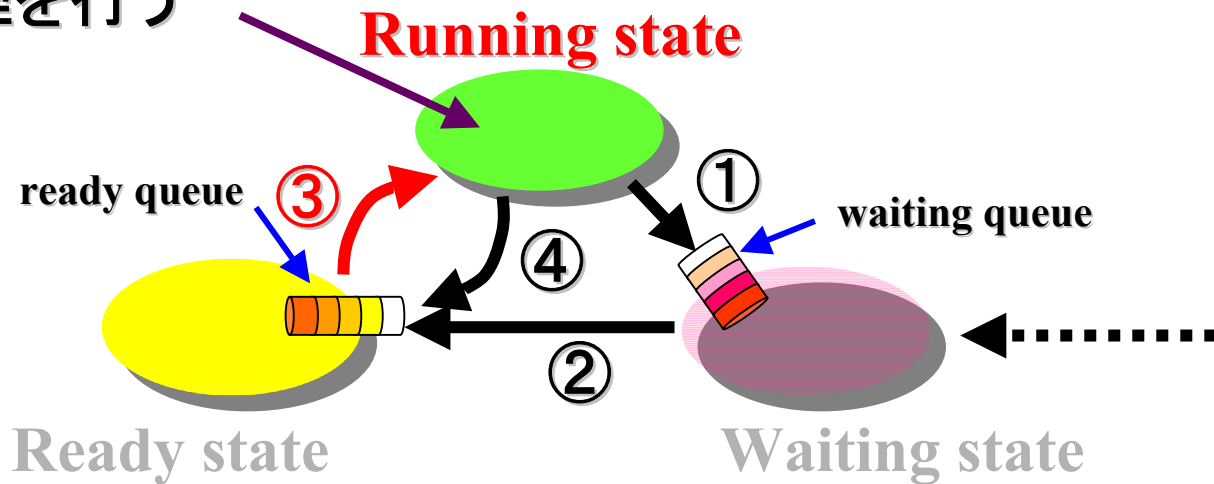


入力完了割り込みを受けて、
待ち状態だったプロセスは
実行可能状態へ遷移

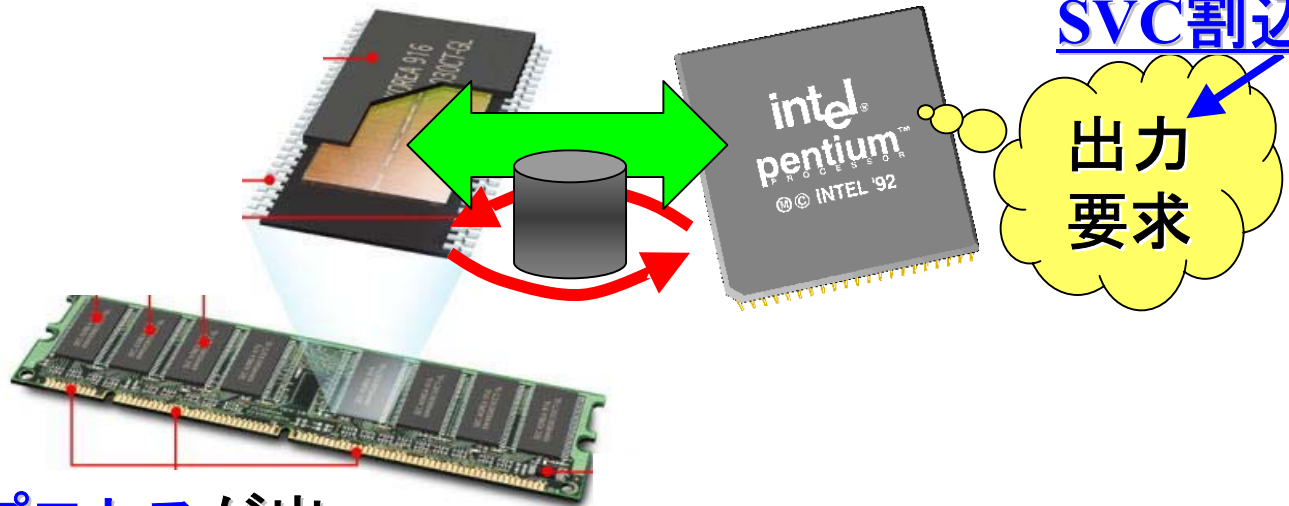




実行状態のプロセスは
メモリ上にあつてCPUを
占有して処理を行う

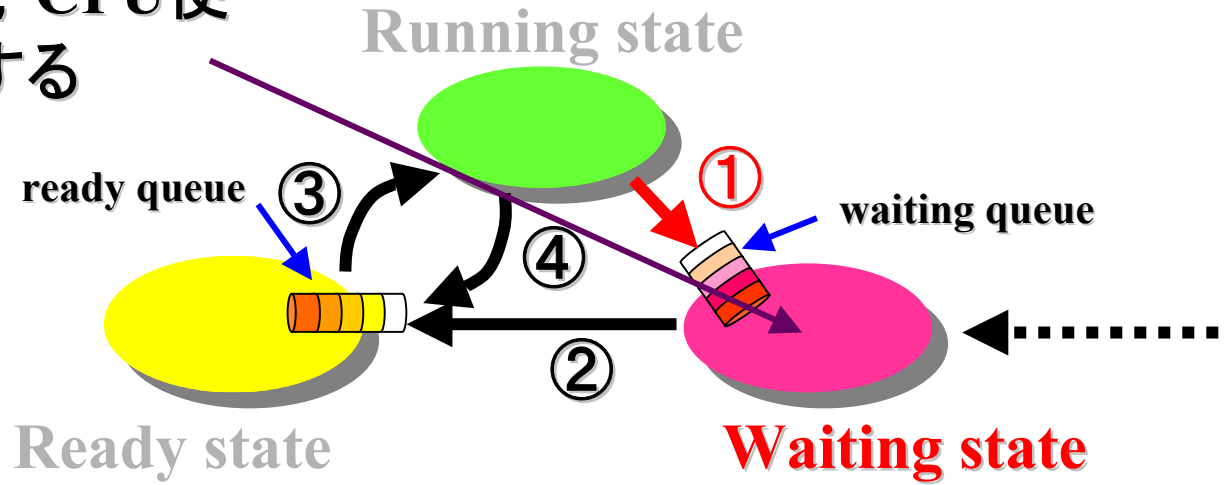


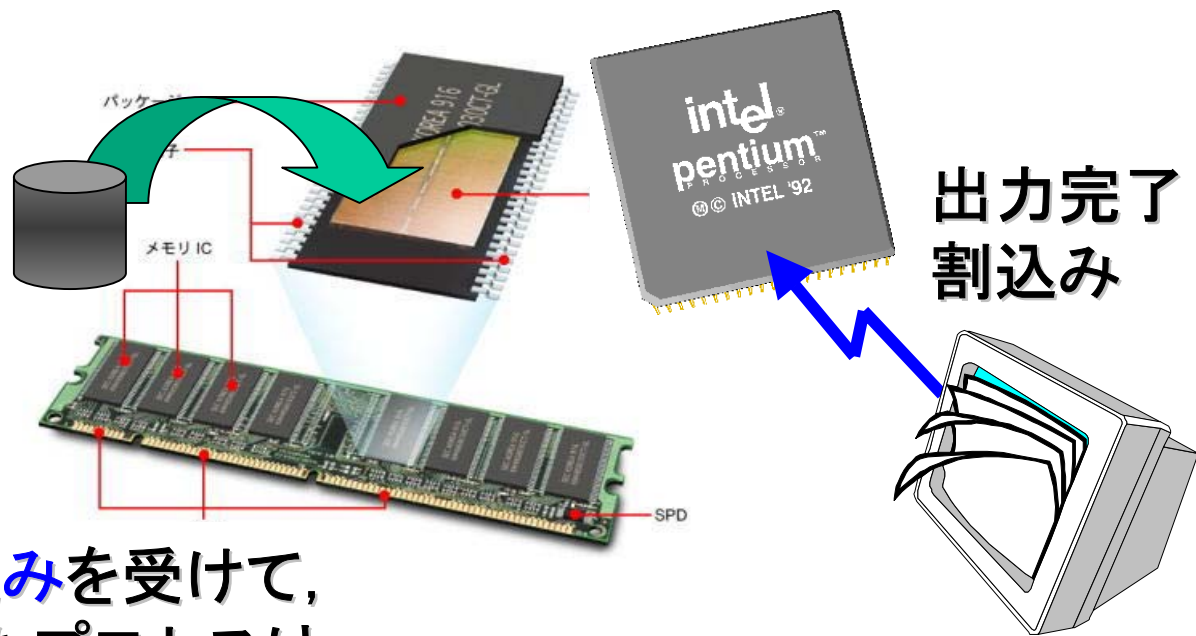
SVC割込み



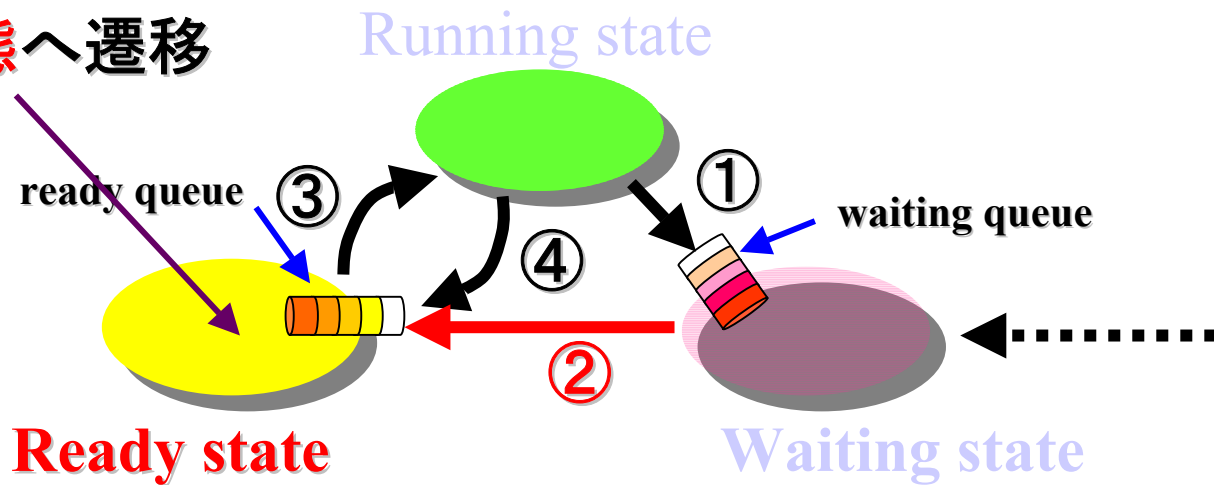
実行状態プロセスが出力要求(SVC割込み)を発することで、CPU使用権を破棄する

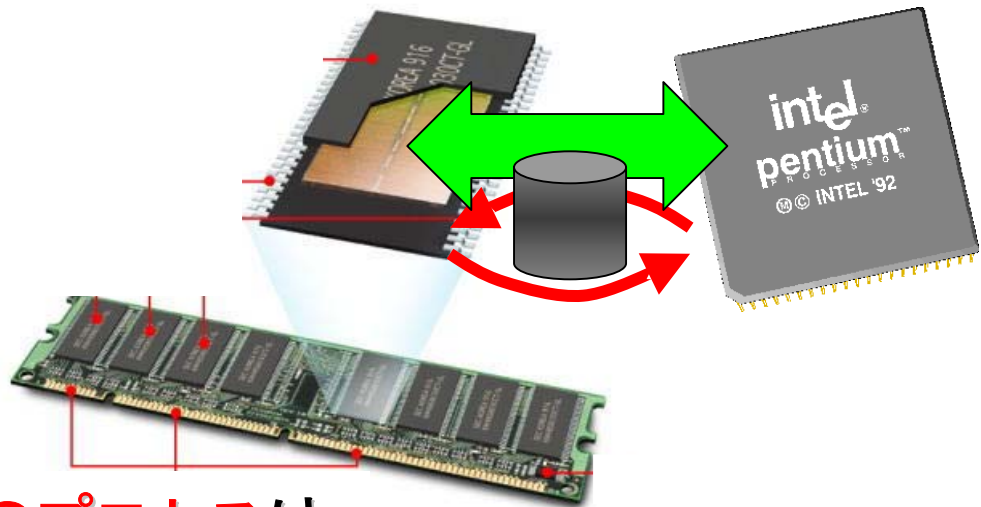
待ち状態へ遷移する



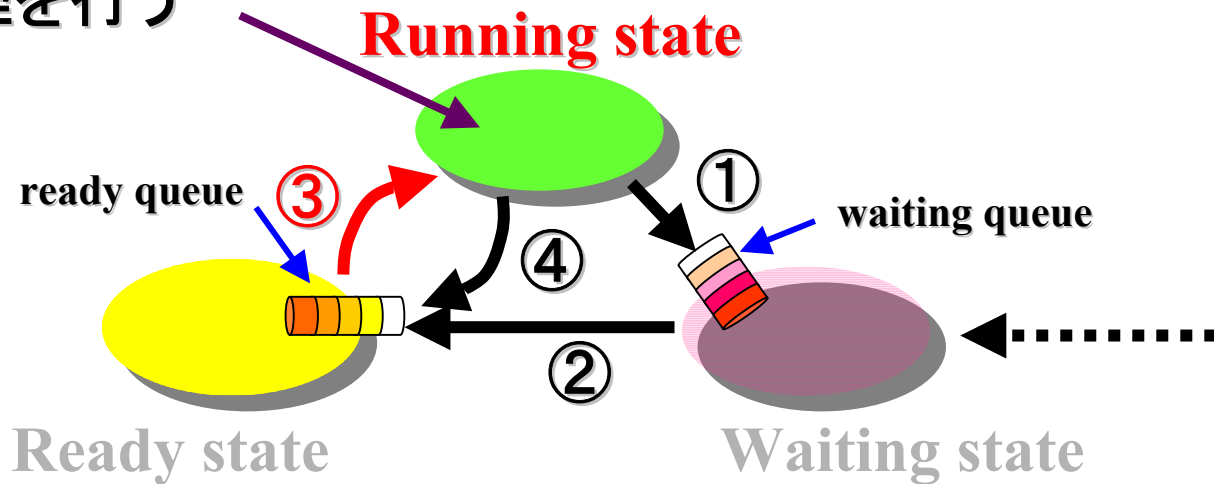


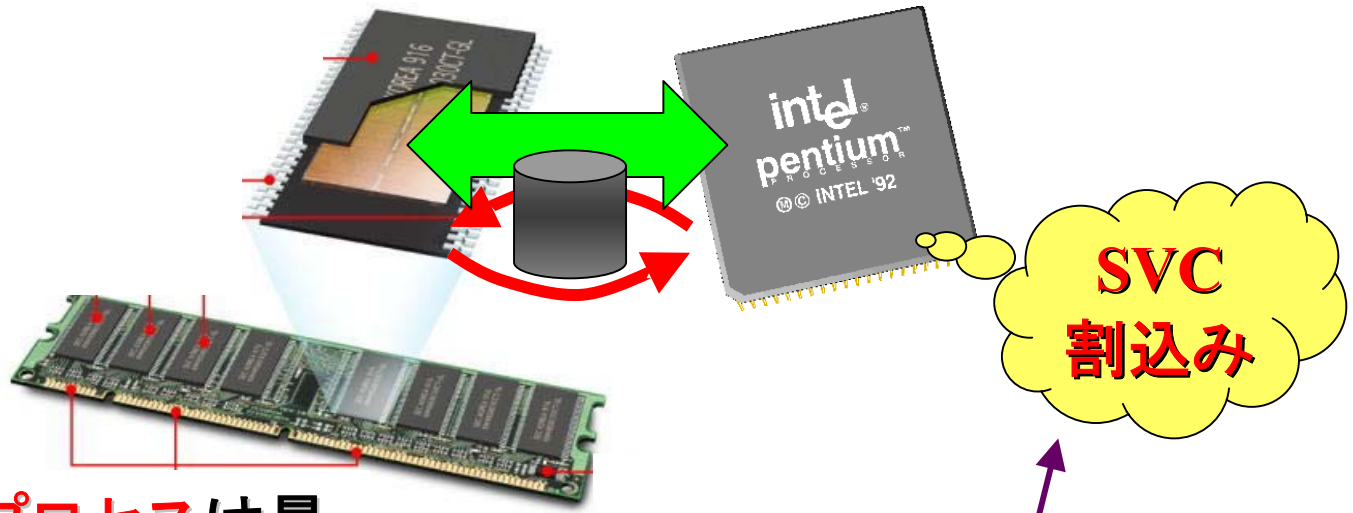
出力完了割り込みを受けて、
待ち状態だったプロセスは
実行可能状態へ遷移





実行状態のプロセスは
メモリ上にあつてCPUを
占有して処理を行う





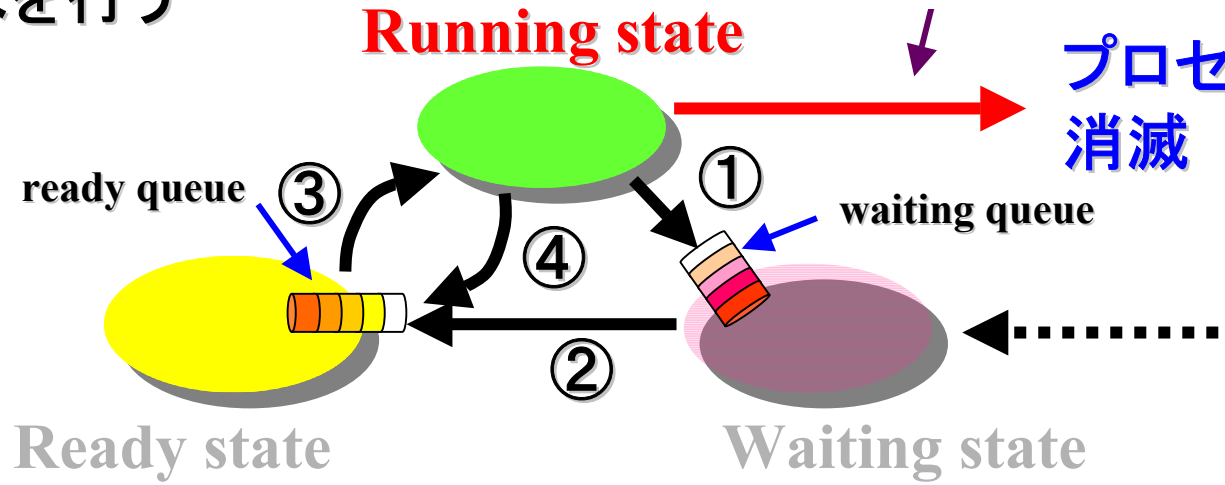
実行状態プロセスは最後にSVC割込みを発生して、終了要求を行う

プロセスの終了要求

プロセス消滅



プロセスはシステムに対し、終了要求を発生して、状態遷移を終え、**プロセス消滅**



マルチプログラミングをベースとしたプロセスの一生

