Operating Systems

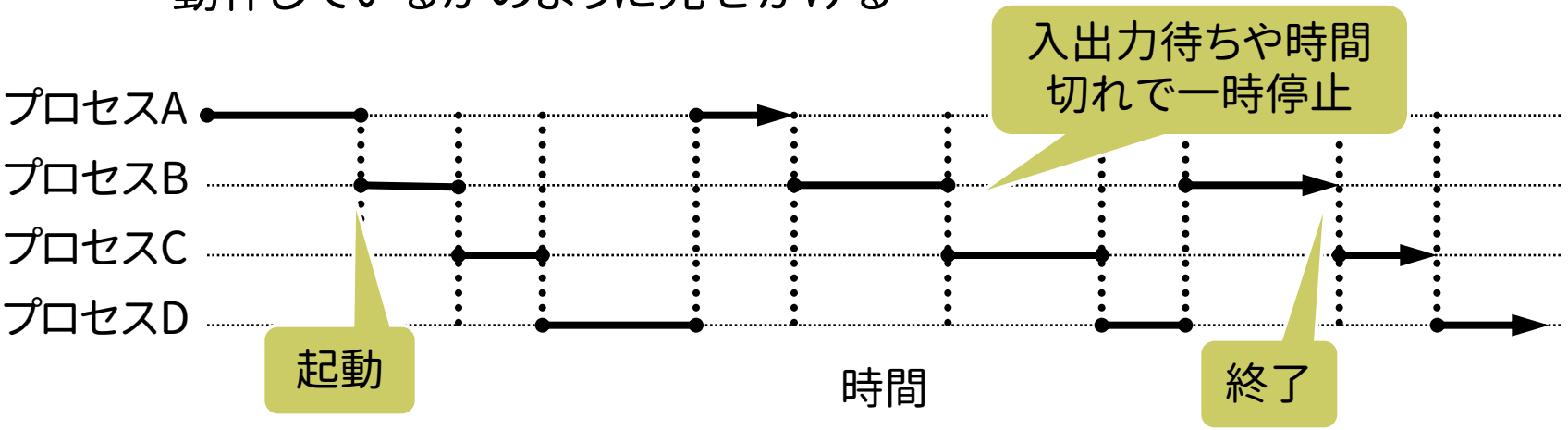
第6回 プロセスのスケジューリング

塩澤 秀和

マルチタスク(マルチプロセス)

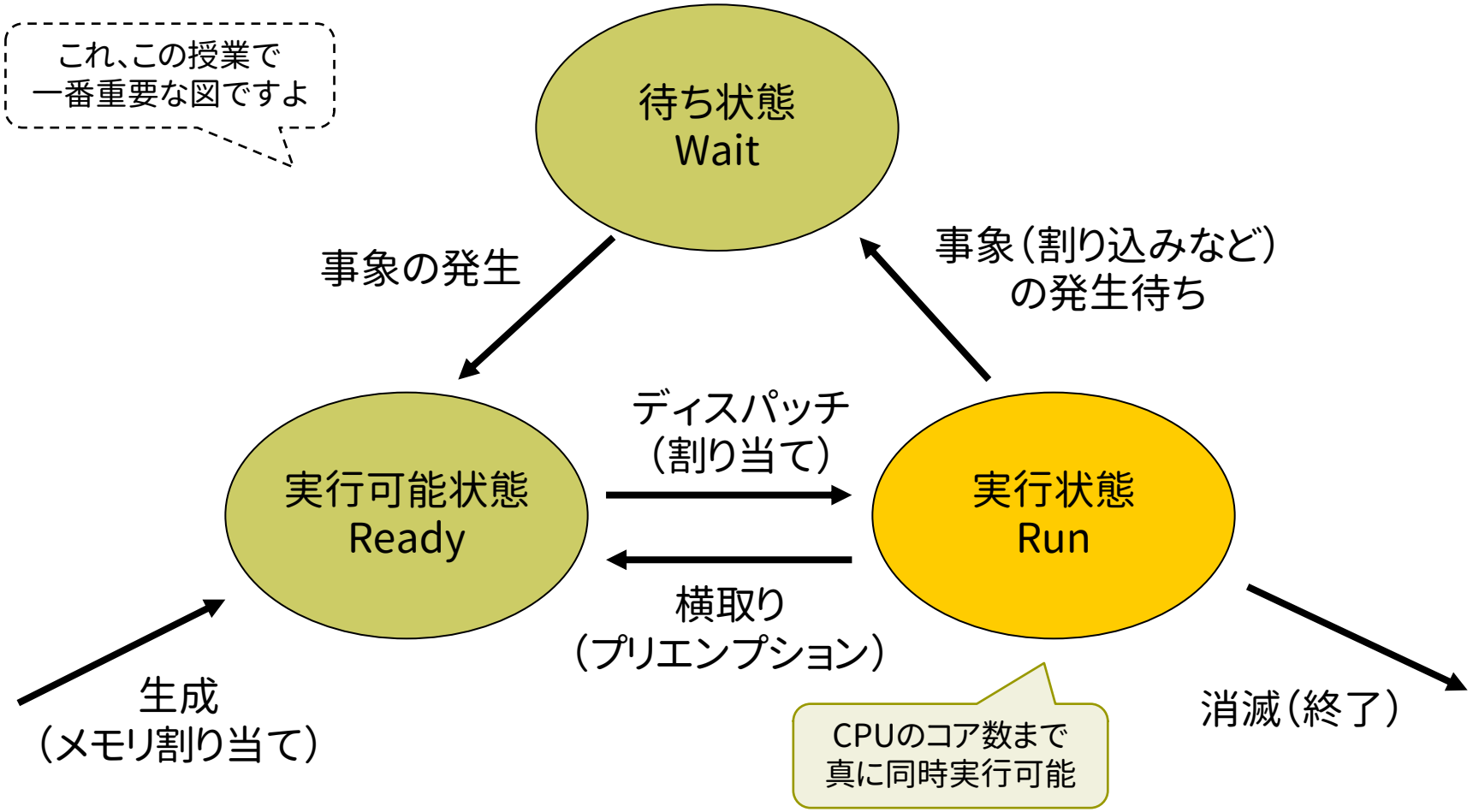
- マルチタスクとは?
  - 1つのCPUで複数のプログラムを(見かけ上)“同時に”実行できる
  - 前のプロセスが終了していなくても,新しいプロセスを起動できる

- マルチタスクの基本アイデア
  - あるプロセスがディスク装置の処理などで瞬間的に止まっている間,別のプロセスに空いているCPUを使わせる
  - さらに,実行プロセスを細かい時間で次々に切り替えて,同時並行に動作しているかのように見せかける



プロセスの状態遷移

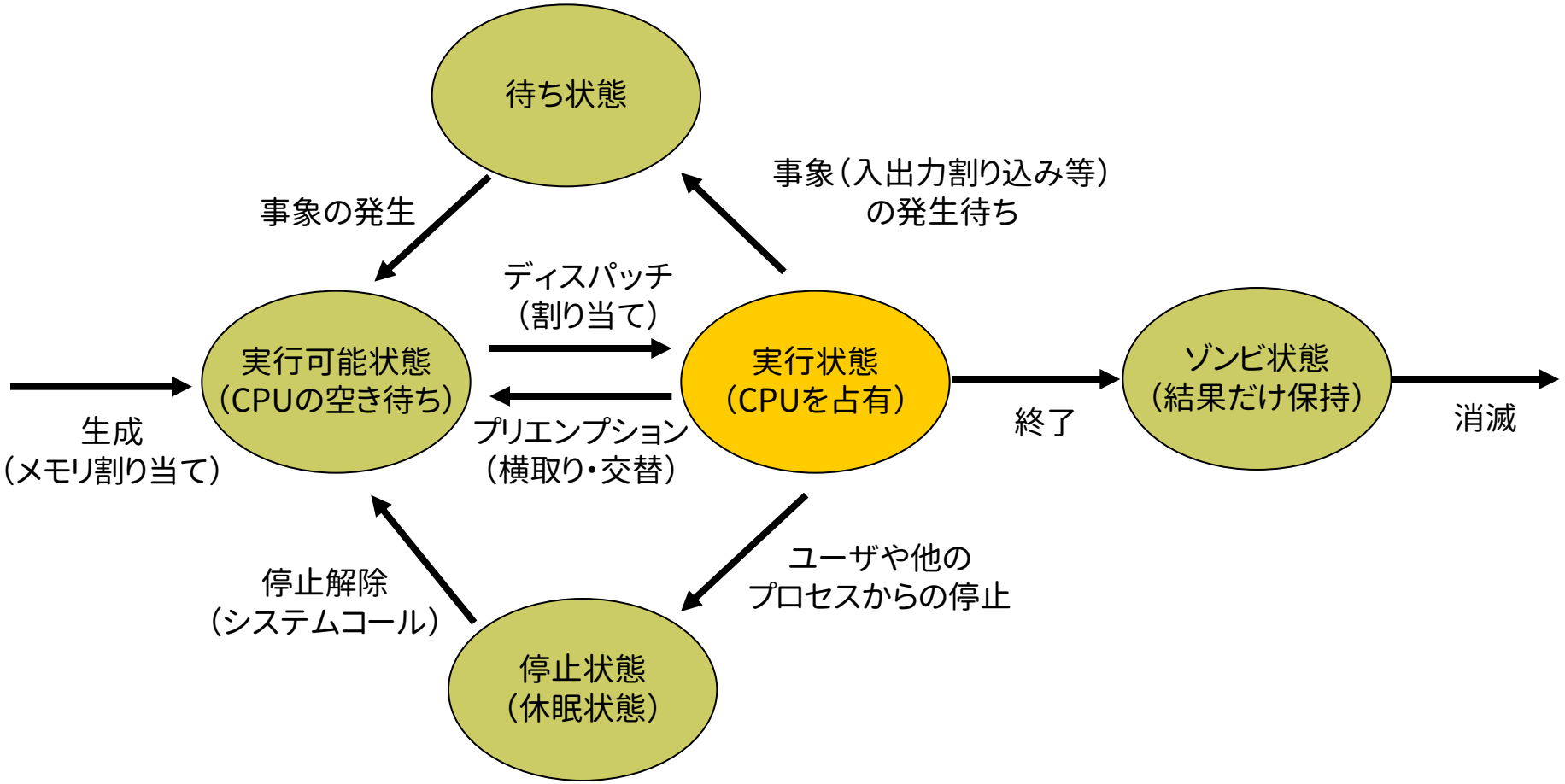
- マルチタスクにおけるプロセス（スレッド）の一生



状態遷移の説明

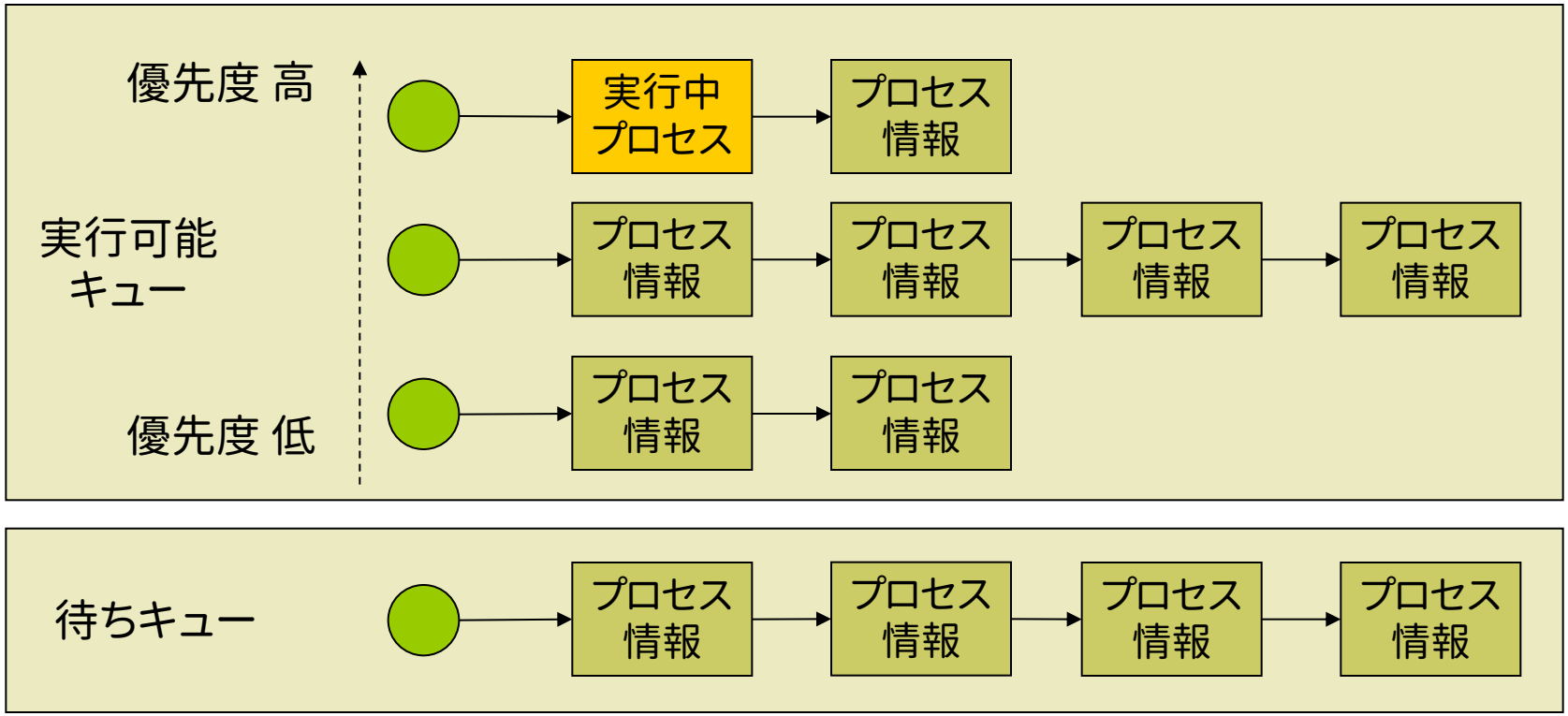
- 実行可能(Ready)→実行(Run)
  - OSがCPUに新しいプロセスを割り当てる(ディスパッチ)
  - 割り当てるプロセスの順番は,プロセスキューで管理されている
- 実行(Run)→実行可能(Ready)
  - OSが現在のプロセスを中断し,順番を次にまわす(プリエンプション)
- 実行(Run)→事象待ち(Wait)
  - プロセスがデバイスなどの応答待ちに入って中断する
  - 事象が発生したら,待っていたプロセスは実行可能状態に戻る
- スケジューリング
  - プロセスがCPUで実行される時間と順番をOSが管理すること
  - 用途に応じた様々なスケジューリングアルゴリズムがある

UNIX/Linuxの状態遷移



プロセスキュー

- プロセス実行順序の管理方法
  - プロセスキュー：カーネル内にあるプロセスの順番表
  - プロセス制御ブロックが実行順につながれた連結リスト構造



キュー（Queue）

- キュー（Queue）とは
  - 「待ち行列」と訳される（代数学の行列（matrix）とは無関係）
  - 何かを待っている行列を表すデータ構造
  - □ ←●●●●●●●●●　（スーパーのレジなど）
- FIFO（先入先出方式）
  - First In First Out
  - 基本的には，先着順に処理されるデータ構造
  - 反対：LIFO（Last In First Out）＝スタック
- 実行可能キュー（OS用語）
  - 別名「レディキュー」「ランキュー」
  - CPUの空きを待っている実行可能状態のプロセスの待ち行列

プロセス スケジューリング

- スケジューリングの役割
  - スケジューリング = スケジュールを決めること
  - CPUでプロセスを実行する時間と順番を管理すること
  - 用途(目的)に応じた様々なスケジューリングアルゴリズムがある
- スケジューリングの目的
  - CPUの利用効率をより高める
  - 人間等への応答時間をよくする
  - 仕事の優先度(緊急度)を反映させる
- プリエンプション(横取り)の有無
  - OSが実行プロセス(CPU利用)を強制的に切り替えること
  - 多くのスケジューリングアルゴリズムは,プリエンプションが前提

スケジューリングアルゴリズム

- 先着順スケジューリング
  - FCFS: First Come First Service
- 最短時間順スケジューリング
  - SJF: Shortest Job First
- 実行期限順スケジューリング
  - EDF: Earliest Deadline First
- 優先度スケジューリング
- ラウンドロビンスケジューリング
  - RR: Round-robin

先着順スケジューリング

- FCFSスケジューリング
  - First Come First Service
  - 最初に来たプロセスから，最初にサービスを受ける
- プリエンプションの有無
  - なし ⇒ 基本的にはシングルタスクのための単純なスケジューリング
- 現実にたとえてみると…
  - ゲームセンターで，ゲーム機に早く並んだ人から順にゲームができる
  - 前の人が終わるまで，次の人はずっと待っている
- 特徴
  - 原理が単純で，実現が容易である
  - 問題点は？⇒ そもそもマルチタスクと言えない

最短時間順スケジューリング

- SJFスケジューリング
  - Shortest Job First
  - 最も所要時間が短い仕事から順番に取りかかる
- プリエンプションの有無
  - なし ⇒ いったん始めたプロセスは中断しない
  - プリエンプションありに改良 ⇒ Shortest Remaining Time First
- 現実世界にたとえてみると…
  - 量の少ない宿題から早く片付ける
- 特徴
  - プロセスの“平均”待ち時間が最小になる
  - 問題点？⇒ 所要時間は実行前に見積もれないことが多い

「残り時間最短」に改良

実行期限順スケジューリング

- EDFスケジューリング
  - Earliest Deadline First
  - “締め切り”が一番近いプロセスから実行する
- プリエンプション
  - あり ⇒ より締め切りの近いプロセスが発生したら，それに切り替える
- 現実社会にたとえてみると…
  - 出発時刻の早い飛行機の人から，優先的に搭乗手続きをする
- 特徴
  - ロボットの制御など，リアルタイム（実時間）で動くシステム向け
  - 問題点？ ⇒ 実装が難しく，全体として最悪のケースが予測しにくい

優先度スケジューリング

- 優先度スケジューリング
  - 設定された優先度が高いプロセスから実行する
  - 優先度が同じ場合は，FCFSなど他のスケジューリングで割り当てる
- プリエンプション
  - あり ⇒ より優先度の高いプロセスが発生したら，それに切り替える
- 現実社会にたとえてみると
  - 社員食堂に並んでいて，上司が来たら順番を譲らなければならない
- 特徴
  - リアルタイムOS向き（ITRONはFCFS+優先度）
  - 問題点？⇒ いつまでも後回しになるプロセスが発生 ＝ 飢餓状態

ラウンドロビンスケジューリング

- Round-Robinスケジューリング
  - タイマー割り込みを利用し，全プロセスを一定時間のタイムスライス（クォンタムタイム，数十～数百ミリ秒）で切り替えて実行していく
  - 「持ち回り」のことを英語ではRound-robinという
- プリエンプション
  - あり ⇒ タイムスライスが経過したプロセスは横取りされ，次に譲る
- 特徴
  - 平等・均一で，常に状態表示が必要な人間相手のシステムに向く
  - 問題点？ ⇒ 時間管理や切り替え自体の処理が多い
- 優先度つきラウンドロビン
  - UNIX，Linux，Windows等ではRRに優先度を組み合わせている

演習課題

- 課題 6a スケジューリングアルゴリズム
  - 学校で以下のような宿題が出された。
  - これらをOSの「FCFS」,「SJF」,「EDF」,「ラウンドロビン」の各方式と同様な考え方でスケジューリングすると,どのような順序に処理してどのような結果になるか説明しなさい。ガントチャートで示すとよい。

科目	出題日	締め切り	かかる時間
A	6月1日	6月11日	5日間
B	6月2日	6月30日	7日間
C	6月4日	6月15日	3日間
D	6月5日	6月8日	1日間

  - 問題の単純化のため,1日に取り組める宿題の数は1つだけとする。
  - また,宿題は出題されたその日から取り組めるものとする。

演習課題

- 課題 6b HOSにおけるFCFS+優先度スケジューリング
  - この課題の狙いは，実際のプログラミングによって，FCFSスケジューリングと優先度スケジューリングの仕組みを理解することである。
  - HOSはシンプルな組み込みOSなので，基本的なスケジューリングは，FCFS（先着順）と優先度によるものである。
  - FCFSはタスクの起動順，優先度はタスクの設定情報によって決まる。

- 手順
  - sample¥win-scheduling を（元のものをコピーして）実行する。
  - 実行結果を示し，なぜそのような動作するのか，スケジューリングの観点から考察（理由を説明）せよ。
  - 各タスクの優先度（T_CTSK構造体の4番目の値）や start 関数での実行順序を変更していくつかの組み合わせを試してみよ。
  - 実行結果を示して，それらの効果について考察せよ。

次回：小テストとプログラミング実習

- 小テスト
  - 第1回～第6回に関する用語問題と記述問題
  - 今年度は，オンラインで実施
- プログラミング実習
  - 第7回の演習課題の説明と実習

HOS（μITRON）のタスク管理

サービスコール	意味	説明
cre_tsk	create task	タスクの生成
acre_tsk		同上（ID自動割り当て）
act_tsk	activate task	タスクの起動
iact_tsk		同上（割り込み用）
can_act	cancel activation	タスク起動予約の取り消し
ext_tsk	exit task	自タスクの終了
ter_tsk	terminate task	タスクの強制終了
chg_pri	change priority	タスク優先度の変更
get_pri	get priority	タスク優先度の参照
rot_rdq	rotate ready queue	実行可能キューの回転
irot_rdq		同上（割り込み用）

HOS（μITRON）のタスク管理

サービスコール	意味	説明
dly_tsk	delay task	自タスクの遅延
slp_tsk	sleep task	自タスクの休止（起床要求待ち）
tslp_tsk		同上（タイムアウトあり）
wup_tsk	wake up task	タスク起床要求
iwup_tsk		同上（割り込み用）
can_wup	cancel wakeup	タスク起床要求の取り消し
rel_wai	release wait	待ち状態の強制解除
irel_wai		同上（割り込み用）
sus_tsk	suspend task	強制待ち状態への以降
rsm_tsk	resume task	強制待ち状態からの再開
frsm_tsk	force resume task	強制待ち状態からの強制再開