知能制御システム学

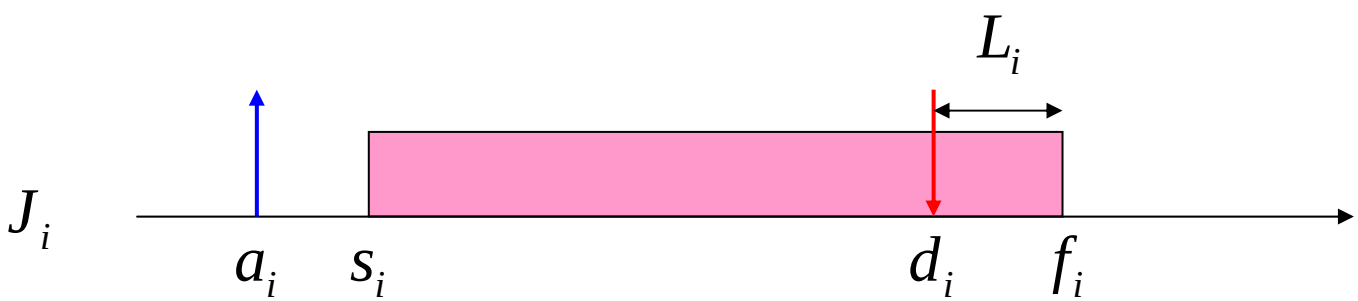
リアルタイムシステム (2)

東北大学 大学院情報科学研究科
鏡 慎吾
swk(at)ic.is.tohoku.ac.jp

2006.07.11

復習: EDF スケジューリング

任意の arrival time を持つ独立なタスクの集合が与えられたとき，各時刻において，実行可能なタスクのうちデッドラインが最も早いものを実行するアルゴリズムは，最大遅延時間を最小化する意味で最適である



$$L_{\max} = \max_i (f_i - d_i)$$

これはタスクが周期的だろうが，非周期的だろうが，一般的に言える

周期的タスク

軌道計画
~ 1[s]

ダイナミクスモデル計算
~100 [ms]

モータ制御 ~1[ms]

レーザ飛行時間距離計測
~100[ps]

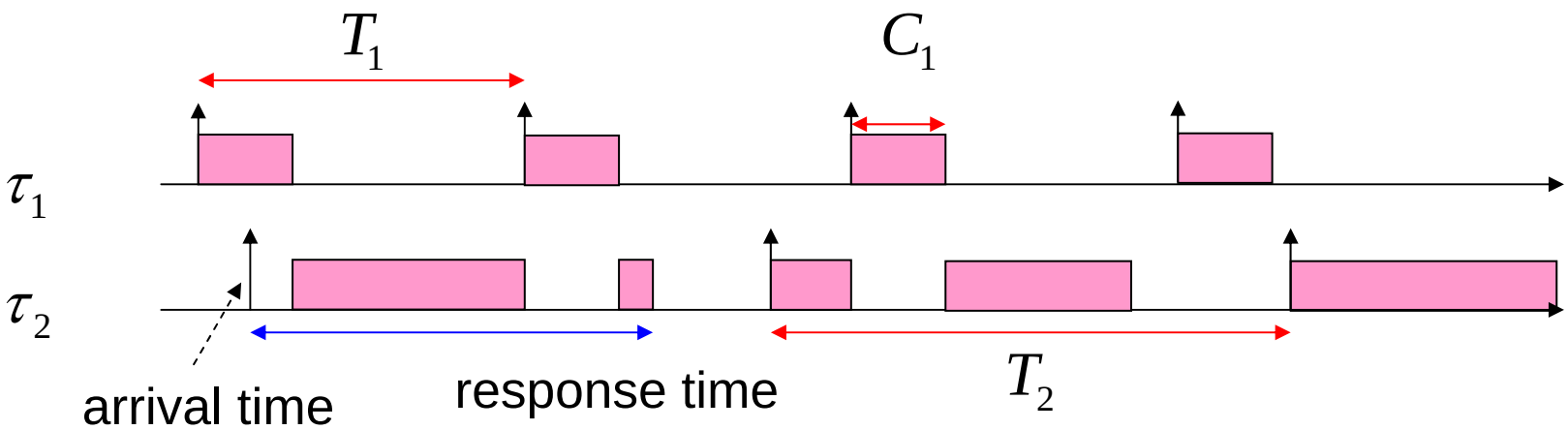
音声波形生成
~20 [μs]

力覚センサ処理
~1 [ms]

画像処理
~33 [ms]

音声信号処理
~100 [ms]

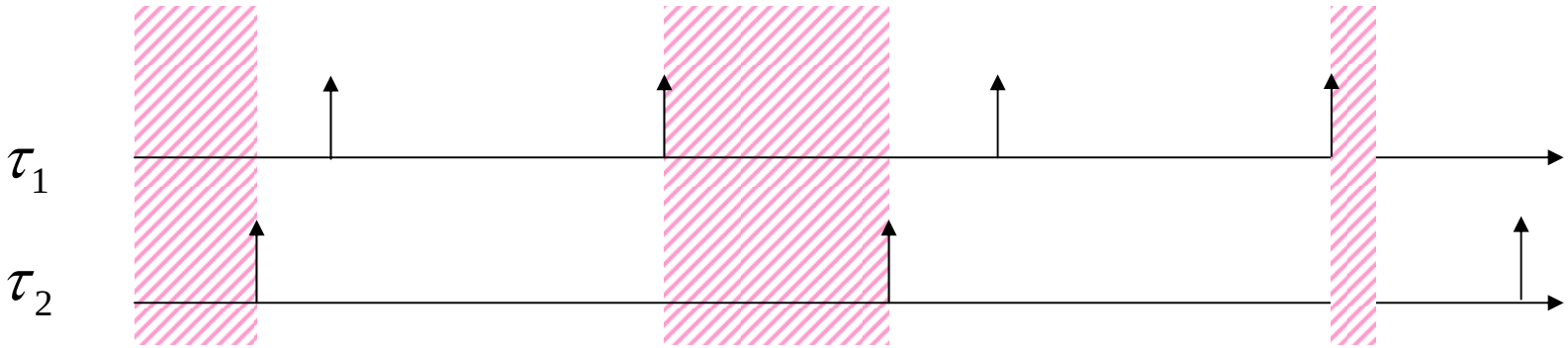
周期的タスクの実行例



注) 周期 = 相対デッドラインと考えている
相対デッドライン < 周期の場合に適用可能なアルゴリズムも存在する
(Deadline Monotonic)

固定優先度と動的優先度

EDFの場合



網掛けの時間帯は，τ_2 が優先，それ以外は τ_1 が優先
時刻によってタスクの優先度が変わっている（動的優先度）

タスクごとに優先度を固定で決められる方が実装は簡単
（固定優先度）

Rate Monotonic スケジューリング

周期的タスクの集合に対して，周期の短い順に（レートの高い順に）優先度を割り当てるスケジューリングを Rate Monotonic (RM) **と呼ぶ．**

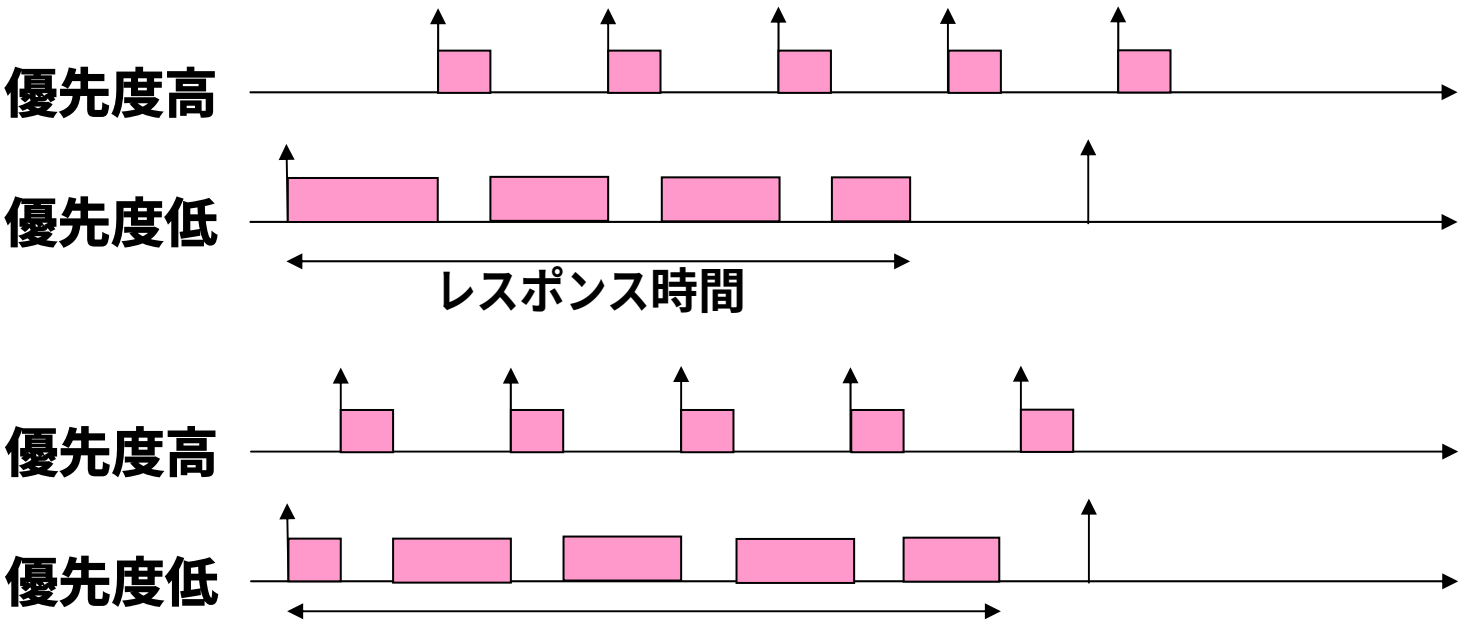
注) 固定優先度となる

RM **によるスケジュールが実行不可能であれば，他のどんな固定優先度割り当てによるスケジュールでも実行不可能である．この意味で，**RM**は最適である．**

準備

あるタスクのレスポンス時間が一番長くなるのは，それが自分より高い優先度のタスクすべてと同時に投入されたときである．

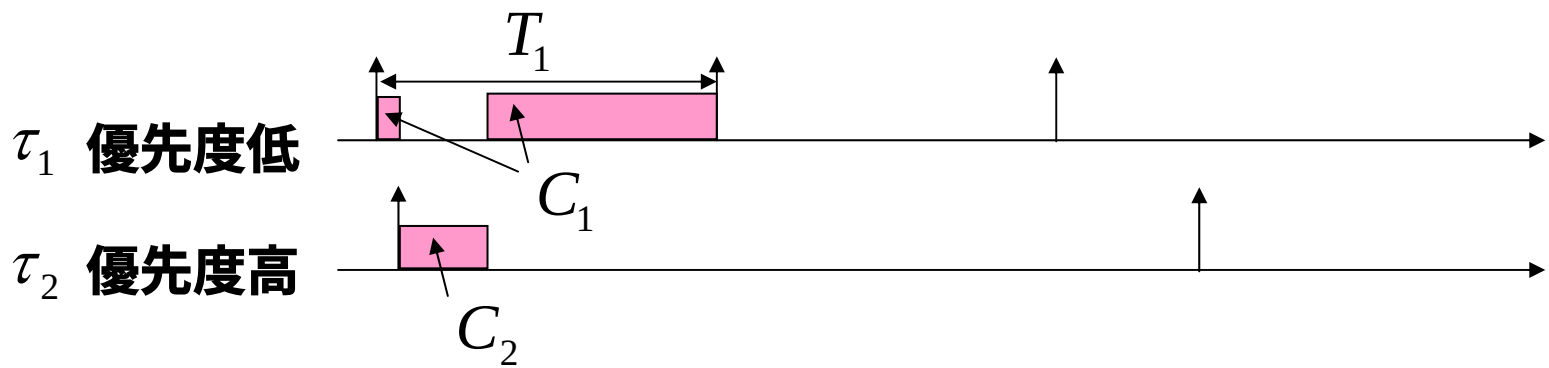
要するに，全部のタスクを一斉に投入するケースが一番厳しい．この場合にスケジュール可能ならば，他の場合でも大丈夫．



高優先度タスクの投入が早くなるほど
レスポンス時間が長くなる（かぶりやすくなる）

タスクが2つの場合

タスクが2つの場合を考える．RMではない優先度割当をしたときに実行可能であるなら，RMのときにも必ず実行可能であることを示す．



RMでない優先度割当で実行可能なら

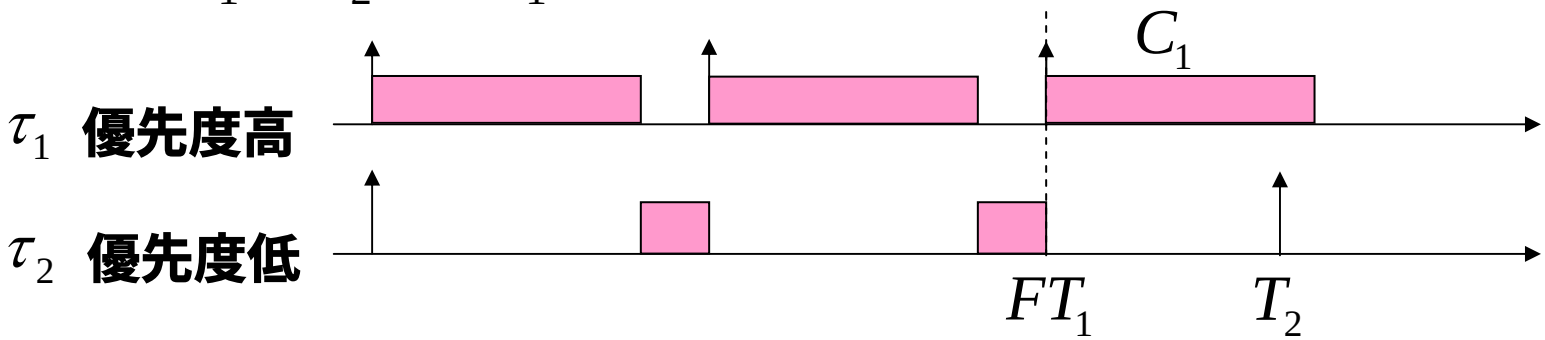
$$C_1 + C_2 \leq T_1$$

でなければならない．この条件が成り立つとき，RMな優先度割当によるスケジュールは常に実行可能であることを示せばよい．前ページの考察から，両タスクが一斉に投入される場合に実行可能であればよい

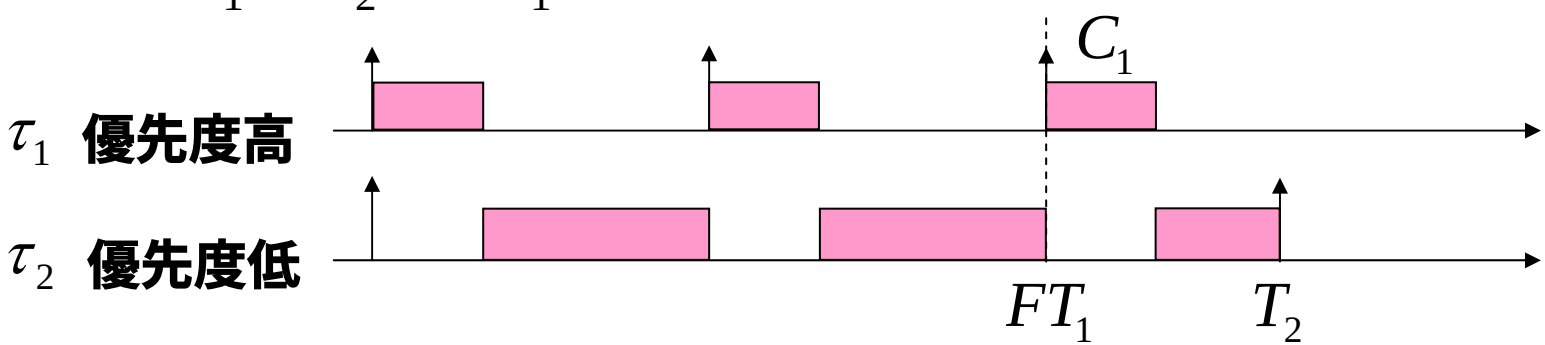
RMのときの場合分け

T_2に含むことのできるT_1の数をFとする　$F = \lfloor T_2 / T_1 \rfloor$

Case1: $C_1 \geq T_2 - FT_1$



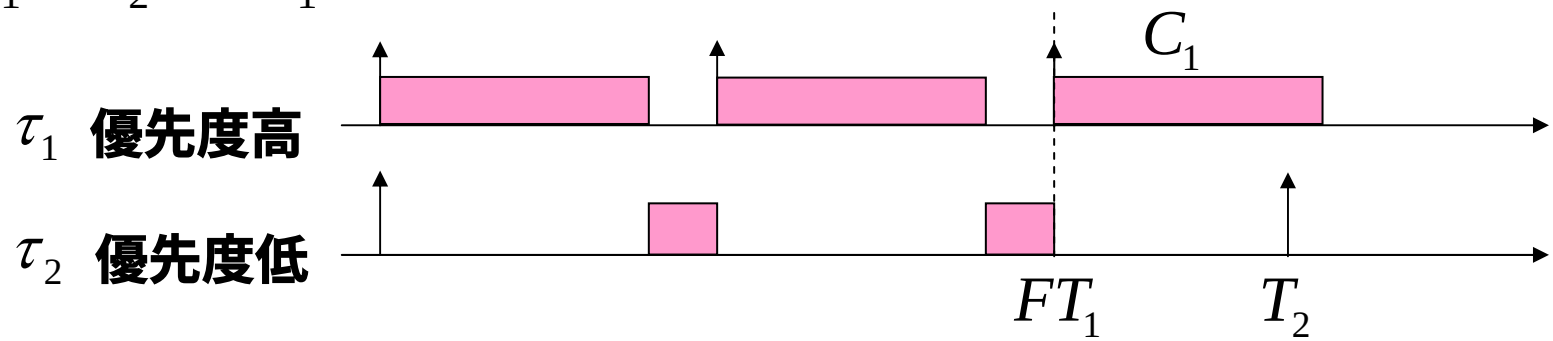
Case2: $C_1 \leq T_2 - FT_1$



この図では $F = 2$

Case1

$C_1 \geq T_2 - FT_1$ **の場合**:



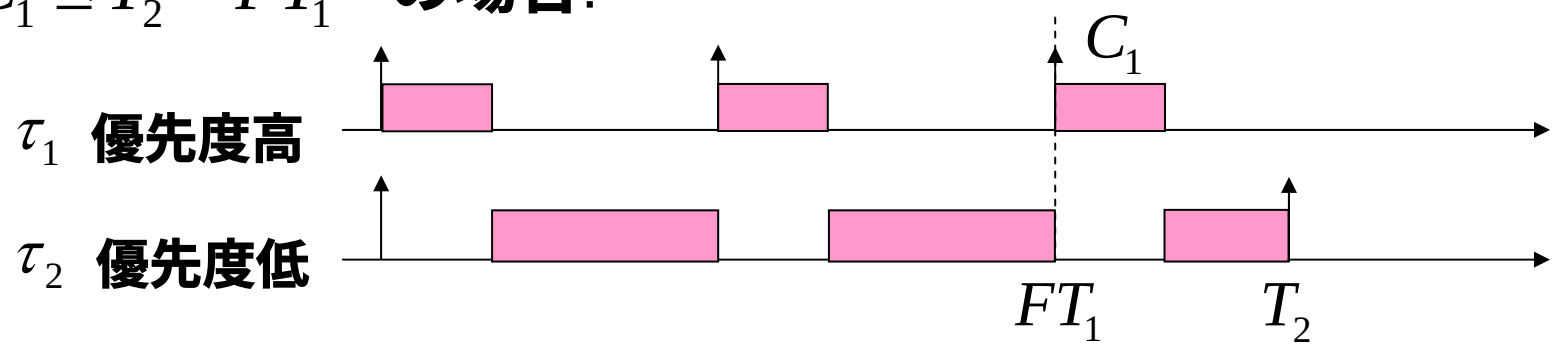
$FC_1 + C_2 \leq FT_1$ **であれば実行可能と言える**

RM**でなくて実行可能，つまり** $C_1 + C_2 \leq T_1$ **ならば**

$$FC_1 + FC_2 \leq FT_1$$
$$FC_1 + C_2 \leq FT_1 \qquad (\because F \geq 1)$$

Case2

$C_1 \le T_2 - FT_1$ **の場合**:



$(F+1)C_1 + C_2 \le T_2$ **であれば実行可能と言える**

RM**でなくて実行可能，つまり** $C_1 + C_2 \le T_1$ **ならば**

$$FC_1 + C_2 \le FT_1 \qquad \text{(Case1参照)}$$

$$FC_1 + C_1 + C_2 \le FT_1 + C_1 \qquad \text{(両辺に } +C_1\text{)}$$

$$(F+1)C_1 + C_2 \le FT_1 + C_1 \le T_2 \qquad (\because C_1 \le T_2 - FT_1)$$

タスクが複数の場合

あるRMでない優先度割当によって実行可能なタスクの集合

$$\tau_1, \tau_2, \cdots, \tau_n$$

を考える. 優先度の高い順に並んでいるとする.

隣接する τ_i, τ_j を考えて, $T_i > T_j$ だったら順序を入れ替える.

この入れ替えを行っても, 実行可能なままであることが今までの議論から分かる. (これらより高い優先度のタスクに割り当てられた時間を除いて考えればよいし, これらより低い優先度のタスクについては, これらの順序を入れ替えたところで残り時間が変わるわけではない)

この入れ替えを繰り返せば RM な優先度割当に行き着く. つまり何らかの実行可能な優先度割当が存在するならば, RMによる優先度割当は実行可能であることが示された.

RMスケジューリングが可能な十分条件

プロセッサ使用率を以下で定義する. (n **はタスク数**)

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i}$$

あるタスク集合のプロセッサ使用率が以下を満たせば,
RMによるスケジュールは実行可能である

$$U \leq n(2^{1/n} - 1) \rightarrow \ln 2 \cong 0.69 \ \ (\text{as} \ \ n \rightarrow \infty)$$

注) 必要条件ではない

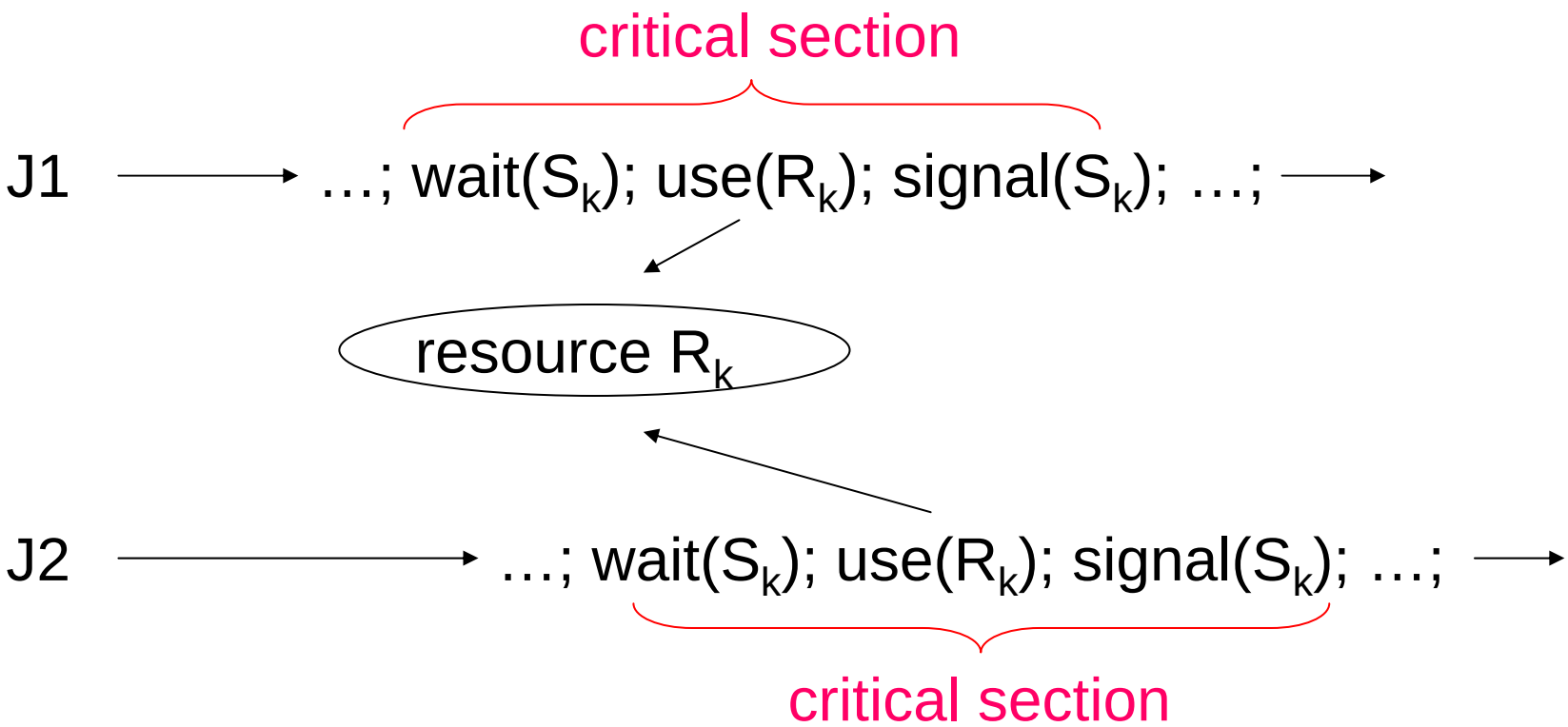
EDFの周期タスクスケジュール可能性

ある周期タスクの集合がEDFでスケジュール可能な必要十分条件は以下で与えられる

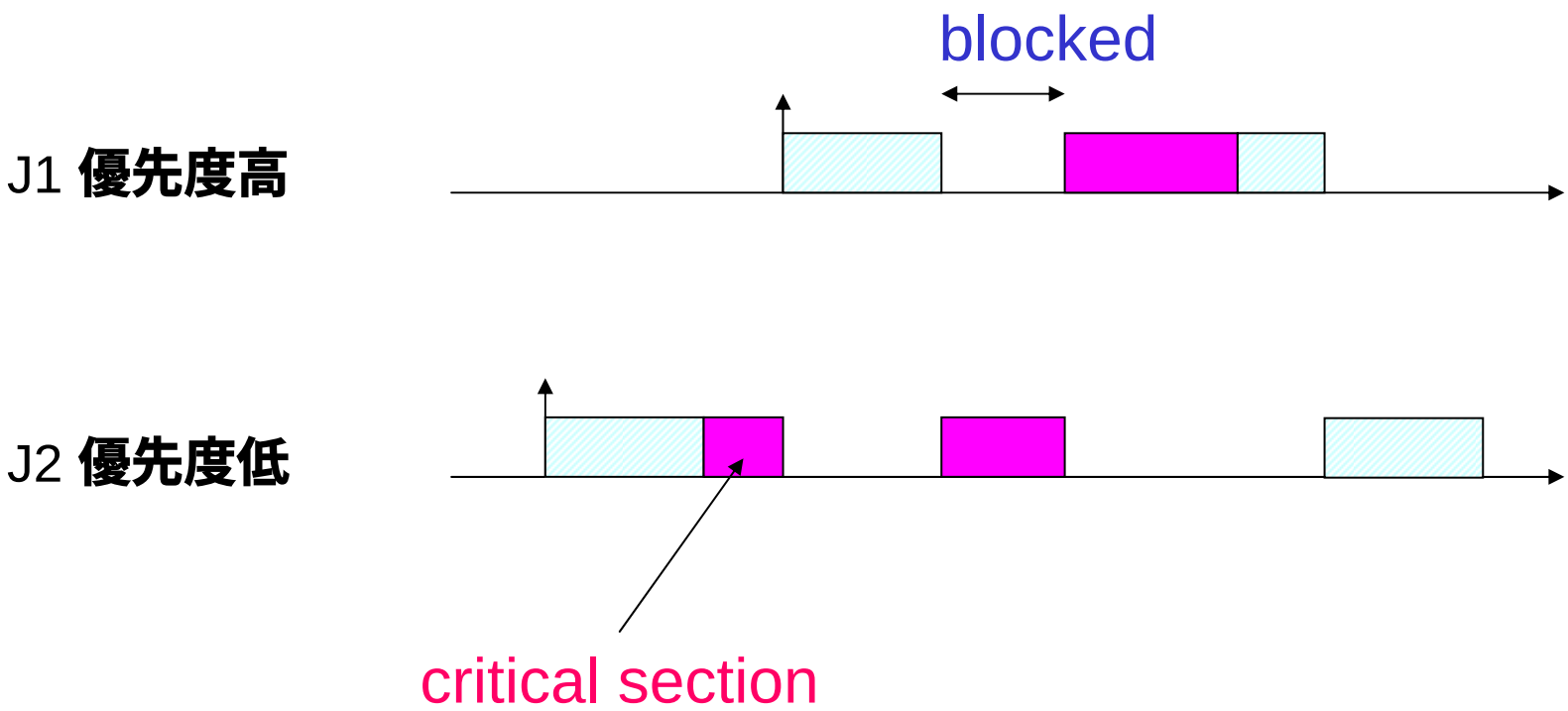
$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$$

タスク間で共有資源がある場合

同時にアクセスできない共有資源がある場合

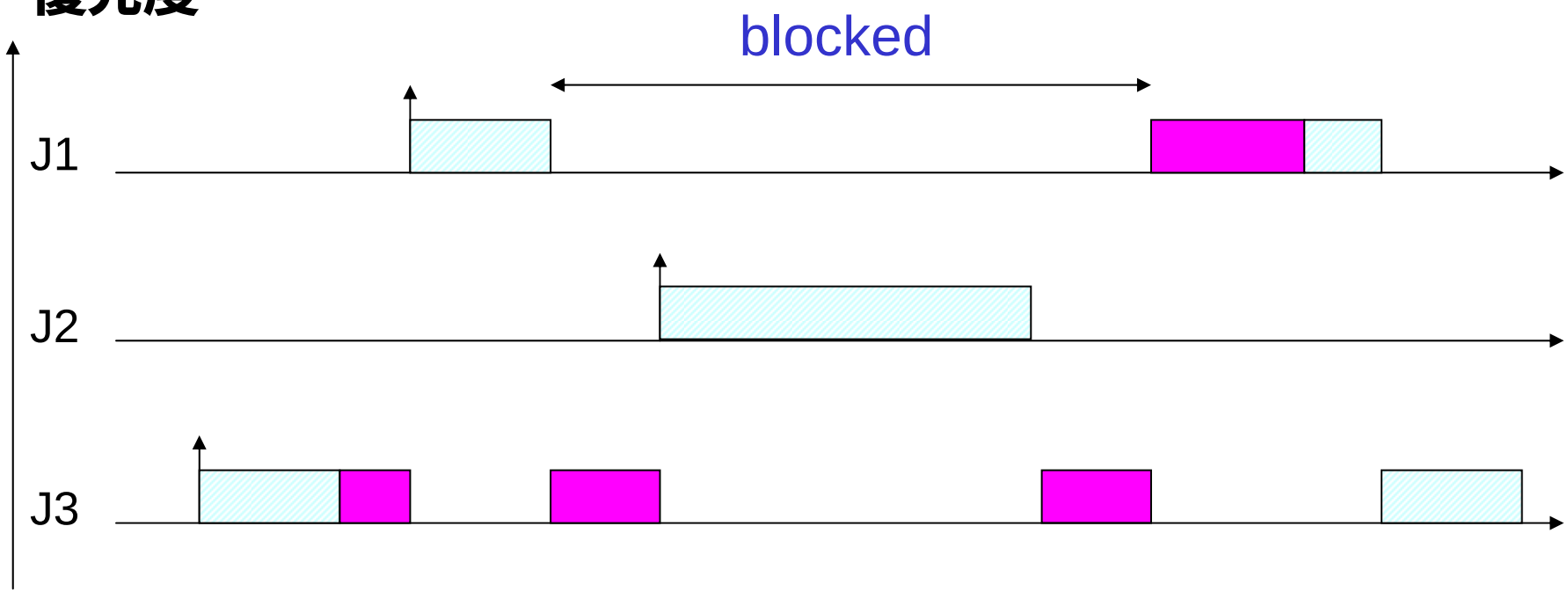


実行のブロック



J2がクリティカルセクションを抜けるまで，J1は待たなければならない
これはクリティカルセクションなのでしかたない
(e.g. クリティカルセクションはできるだけ短くする，など)

優先度逆転問題

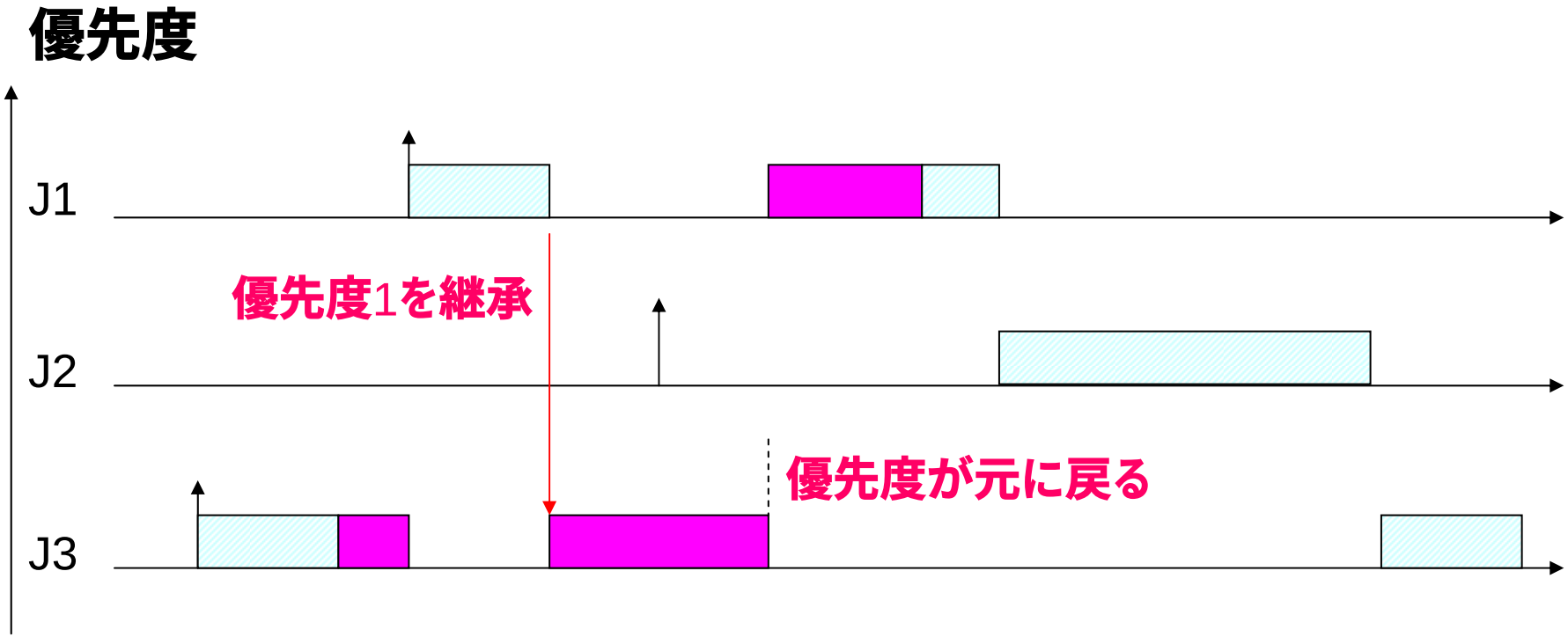


J2 の非クリティカルセクションによって J1 がブロックされてしまう

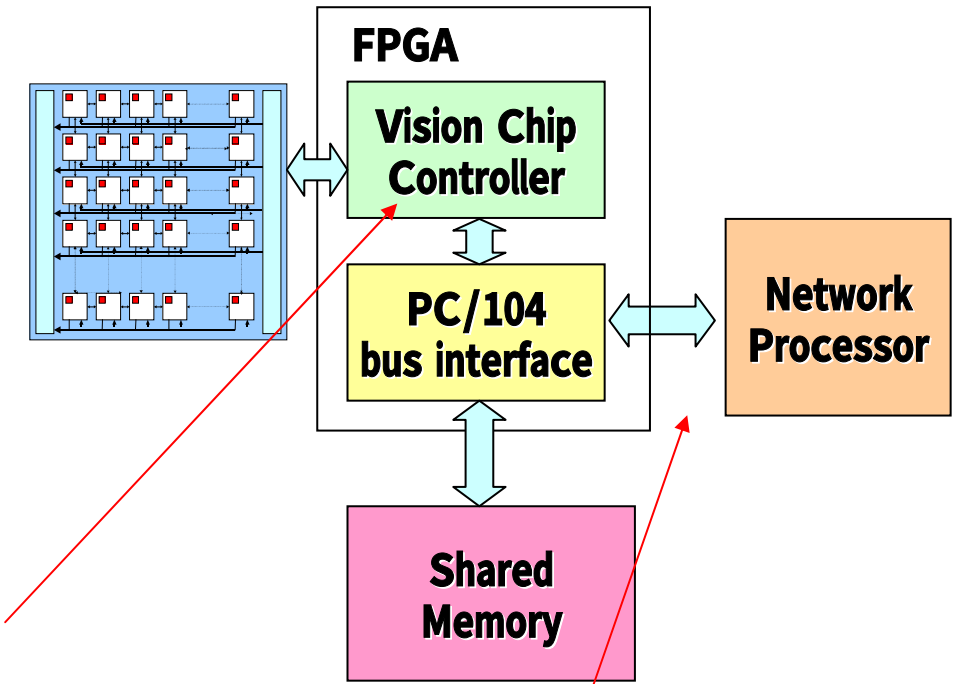
優先度継承プロトコル

- 各タスク J_i は，本来の優先度 P_i と，各時刻における動的な優先度 p_i を持つ
- あるタスク J_i はクリティカルセクションの入り口で，より低優先度のタスク J_k に待たされるとき，優先度を継承して $p_k = p_i$ とする
- J_k がクリティカルセクションから出るときには元に戻る
- （その他，クリティカルセクションが多重化されたときのルールがある）

優先度継承の動作



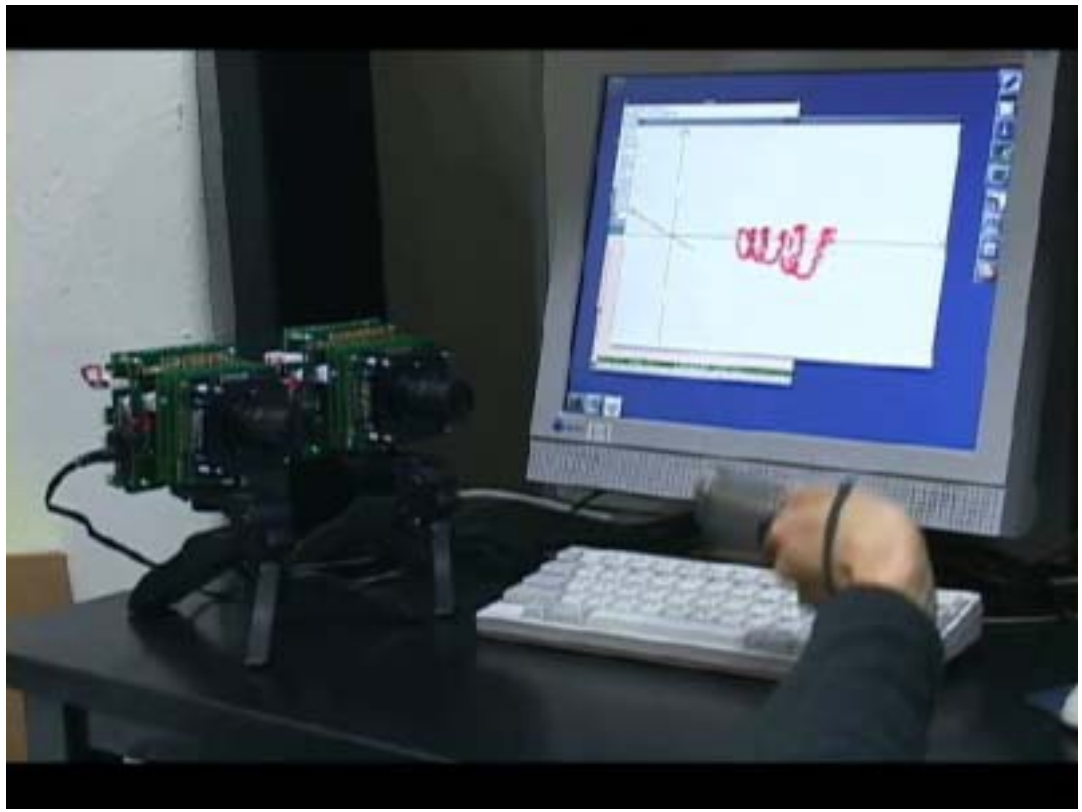
ビジョンチップにおけるリアルタイム処理



ソフトウェアA-D**変換等を含む高時間分解能のリアルタイム処理**:
~ 100 ns
(**専用アーキテクチャ**, OS**無し**)

- 1000fps **視覚情報取得**
- UDP**通信**
  ~ 1 ms
  (SH-4, μITRON**互換** OS)

動作デモ



[**齊藤**2005]

References

[Buttazzo1997] G. C. Buttazzo: Hard Real-Time Computing Systems – Predictable Scheduling Algorithms and Applications, Kluwer Academic, 1997.

[Liu1973] C. L. Liu and J. W. Layland: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, J. ACM, vol.20, no.1, pp.46-61, 1973.

[**齊藤**2005] **齊藤, 鏡, 小室, 石川：ネットワーク接続機能を実装した高速ビジョンチップシステム, 日本機械学会ロボティクス・メカトロニクス講演会**2005, 2A1-N-096, 2005.