

専用オンラインシステムにおけるマルチタスクシステム 性能解析†

野中清孝^{††} 大野豊^{†††}

専用オンラインシステムは一般に複雑なマルチタスクで構成される。これらのシステムの性能は多くの要因によって支配されるため、解析モデルによって全体システムの性能を予測することは難しい問題である。

本論文では、専用オンラインシステムの性能を決める基本的要因を各タスクの処理特性、オペレーティングシステムの負荷、各タスクの実行優先度に限定した上でこれらをモデル化し、近似解析によってシステムの性能を表わす指標を導いた。マルチプログラミング制御下の各タスクの処理は、システム内の幅濶のために時間的に引き延ばされるが、本解析では、この引き延ばされる時間の分布をタスクの膨張処理時間分布と定義し、その平均と分散を入力トラフィックと上記の基本的要因の関数として導いた。そして、この結果を使い、全体システムをトランザクションが呼で各タスクがサーバであるような開待ち行列網とみなすことにより、システムの定量的諸量やシステムの限界能力等を求める近似式を導いた。さらに、最適スケジューリング方策やシステムのボトルネックについても導いた。また、ある例題システムを対象としてシミュレーションを行うことにより、本解析の有効性を確かめた。特に、システムの限界能力に関しては、極めてよい精度で計算できることを確認した。

1. はじめに

計算機システムの総合的な性能を表わす指標を、解析モデルによって導くことは、それを支配する要因の多さと複雑さのために難しいと指摘されていたが、Buzenが1973年に、サービス時間が指数分布であるような閉待ち行列網モデル (closed queuing network model) における平衡状態確率の効率のよい計算アルゴリズムを提示¹⁾して以来、この閉待ち行列網を不特定多数のユーザの計算処理を行うバッチ、TSSシステムに適用して、マルチプログラミング機能を持つ計算機システムの性能解析が、数多くなされてきた^{2),3)}。しかし、バンキングシステム等の専用オンラインシステムに対する性能解析には、上記の閉待ち行列網モデルはほとんど利用されない。これは、閉待ち行列網モデルでは専用オンラインシステムの処理方式および制御構造の特性を十分に反映できないことに起因している。

専用オンラインシステムは、一般に処理能力向上を目的として、マルチタスクで構成される。各タスクは、ある特定のオンラインジョブが機能的に分割された処理の単位であり、システムのマルチプログラミング機能によって、タスクごとにその実行の制御と管理が行われる。これらのタスクは、通常、システム開始

時に生成されて主記憶に常駐している場合が多い。システムに加わる負荷は端末からランダムに入力されるトランザクションであり、このトランザクションがシステム内で必要なタスクの処理を逐次、あるいは並行に受けることによりその処理が完遂される。

システム設計のある段階で、タスク構成やタスクスケジューリングにおける各タスクの実行優先度が決定されるが、これらの要因はいずれも専用オンラインシステムの性能を支配する重要な要因である。したがって、これらの決定の時には、最適決定のための有効な情報を得るために、コストと時間に問題のあるシミュレーションに先だって、待ち行列モデルによる性能解析が一般に行われる。システム設計時の性能解析では、現在設計中のシステムが入力トラフィックに対して、はたしてどの程度まで耐えうるのか、そして、その限界能力がどのような要因に支配されているのかを見極めることが重要である。また、あるタスク構成のもとで、どのような実行優先度を各タスクに設定すれば、処理能力が向上するのか、さらに、ターンアラウンドタイム等のシステムの定量的諸量と入力トラフィックの関係を予測することが重視される。

筆者等は、上述の専用オンラインシステムの総合的な性能指標や性能情報を予測するための1つの解析モデルを提案する。本論文では、初めに専用オンラインシステムの待ち行列モデルとしての構造について検討する。そして、いくつかの限定された条件のもとではあるが、システムの性能を支配する要因を各タスクの処理特性、オペレーティングシステム (以下 OS と

† Performance Analysis for a Multi-task System of a Dedicated Online System by KIYOTAKA NONAKA (Department of Information Engineering, Takuma Radio College) and YUTAKA OHNO (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 鹿間電波工業高等専門学校情報工学科
††† 京都大学工学部情報工学科

略)の負荷,各タスクの実行優先度に限定した上で,これらをモデル化し,近似解法によって入力トラフィックとシステムの定量的諸量の関係,システムの限界能力を上記の要因の関数として導いている.さらに,ボトルネックや最適スケジューリング方策等の有用な性能情報についても導いている.

2. オンラインシステムの構造と解析モデル

この章では,専用オンラインシステムの特徴を待ち行列モデルとしての観点から考察すると共に,この論文の性能解析における前提を明確にする.

専用オンラインシステムでは,端末から入力されたトランザクションはシステムの通信制御プログラムによって受信され,その後,システム内の必要なタスクの処理を逐次,あるいは並行に受けることによりその処理が完遂される.トランザクションがタスクを渡り歩く経路は,タスクをノードとするネットワークで表わされ,この経路はトランザクションの種類ごとに異なる.また,トランザクションはシステムにランダムに到着し,各タスクは一時には,唯一つのトランザクションに対する処理しかできないので,各タスクの前にはトランザクションの待ち行列ができる.トランザクションの到着率が小さい時には,トランザクションはシステム内を定期的に流れるが,到着率を大きくしていくと,ある到着率以上の負荷に対してはシステムの定常性が失われ,いくつかのタスクに対するトランザクションの待ち行列が無限に成長してしまう.この論文では,システムに定常性が保たれるトランザクションの到着率の最大値をスループットと呼ぶ.また,トランザクションの到着率を大きくしていく時,最初に定常性を失う待ち行列に対応するタスクをボトルネックタスクと呼ぶ.トランザクションの流れという観点からみると,上述したように専用オンラインシステムは,トランザクションが呼で,各タスクがサーバであるような開待ち行列網モデルとみることができる.

しかし,マルチプログラミングの制御の観点からみると,タスクをサーバとはみなせない.タスクは単にトランザクションに対する処理の内容を規定しているものである.タスクの処理の内容は,CPU処理とI/O処理の逐次的な系列で表わされるが,このトランザクションの処理に必要なCPU処理,I/O処理を,ハードウェア資源の管理を行うOSに対し要求する実体がタスクであり,実際のサーバはハードウェア資源と言える.したがって,専用オンラインシステムに対する

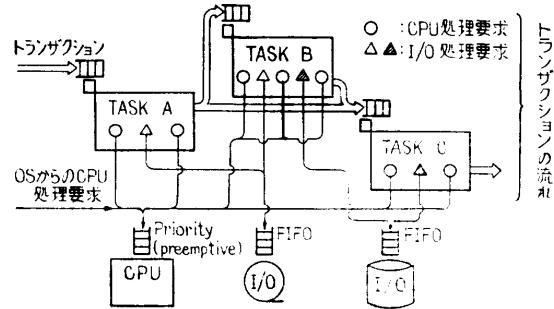


図1 専用オンラインシステムの待ち行列の構造

Fig. 1 Structure of queuing network for an online system as example.

全体的待ち行列モデルは,図1に示すように,トランザクションの流れからみるとタスクを仮のサーバとする開待ち行列網であり,制御的にはハードウェア資源をサーバとする待ち行列が存在する複雑な多重待ち行列とみなすことができる.

専用オンラインシステムでは,各タスクの処理特性が明確であるので,性能向上を目的として一般に優先度方式が採用され,割込が発生するたびにタスクの切換えが preemptive に行われる.また,I/O装置に対する入出力は,通常,先着順のスケジュールが行われるので,この論文の解析においても,これらの制御形態を前提とする.すなわち,CPU処理に対しては preemptive な優先度制御を,I/O処理に対しては先着順の制御を仮定する.さらに,専用オンラインシステムの性能解析では,通信制御等のOSの負荷も無視できない.これらはタスクのCPU処理要求に対し割込的に優先されて処理されるので,解析モデルにおいても最も高い優先度を持つCPU処理要求がタスク以外からランダムに発生していることを考慮にいれる必要がある.ただし,各タスクのプログラムは,主記憶に常駐しているものとする.すなわち,プログラムのSWAP-IN/OUTによるオーバーヘッドがないことを仮定する.以上のことを基本的前提として,3章で本論文での専用オンラインシステムの性能解析を示す.

3. マルチタスクシステムの解析

3.1 マルチタスクシステムの定式化

n 個のタスクから構成される専用オンラインシステムを以下のように特徴づける.

n 個のタスクの優先度の設定の方法は全部で $n!$ があるが,この可能なスケジューリング方策の全体を θ で表わし,このうちの一つのスケジューリング方策を

$\theta \in \Theta$ とする。また、タスクの集合を $T^\theta = \{T_1^\theta, T_2^\theta, \dots, T_n^\theta\}$ で表わす。 T_i^θ はスケジューリング方針 θ のもとで、 i 番目に高い優先度を持つタスクとする。タスク T_i^θ のトランザクションに対する処理は CPU 処理と I/O 処理がこの順序で逐次的に $(N_i - 1)$ 回交互に行われ、最後は CPU 処理で終るものとする。そして、タスク T_i^θ の j 番目の CPU 処理時間 (確率変数 P_{ij} で表わす) と k 番目の I/O 処理時間 (確率変数 D_{ik} で表わす) の分布関数を各々 $F_{ij}(t)$ ($j=1, 2, \dots, N_i$), $H_{ik}(t)$ ($k=1, 2, \dots, N_i - 1$) とする。 $F_{ij}(t)$ は平均が s_{ij} , 分散が σ_{ij}^2 の一般分布, $H_{ik}(t)$ は解析可能性を考慮して、平均が $1/\tau_{ik}$ の指数分布と仮定する。また、確率変数 P_{ij} ($j=1, 2, \dots, N_i$), D_{ik} ($k=1, 2, \dots, N_i - 1$) はすべて互いに独立と仮定する。

$$F_{ij}(t) = \text{Prob}(P_{ij} \leq t). \quad (1)$$

$$H_{ik}(t) = \text{Prob}(D_{ik} \leq t) = 1 - e^{-\tau_{ik}t}. \quad (2)$$

システムに加わる負荷を $M^\theta(\lambda) = (g_1^\theta(\lambda), g_2^\theta(\lambda), \dots, g_n^\theta(\lambda))$ で表わす。 $g_i^\theta(\lambda)$ はタスク T_i^θ に対するトランザクションの到着率であり、システムへの端末からのトランザクションの到着率 λ の関数として表わされるものとする。 OS のオーバヘッドを $V(g_0(\lambda), s_0)$ で表わす。 $g_0(\lambda)$ は割込処理要求の発生率であり、これもまた、システムへのトランザクションの到着率 λ の関数として表わされるものとする。また、 s_0 は一回の割込処理要求に対する CPU 処理時間の平均である。

3.2 近似の方針

タスク T_i^θ において1つのトランザクションに対する処理を開始してから、そのタスクにおける処理を完全に終了するまでの時間を確率変数 X_i で表わす。 X_i はタスク T_i^θ での実質の処理時間 (N_i 回の CPU 処理時間の合計と $(N_i - 1)$ 回の I/O 処理時間の総和) と、マルチプログラミング環境下における高優先度の CPU 処理要求との競合や同一 I/O 装置に対するアクセスの競合等による遅れとの和である。タスク T_i^θ における実質の処理時間の分布は、トランザクションの到着率 λ とは独立である一定の分布を持つが、ハードウェア資源の競合に対する遅れは、 λ が大きくなるに従って、特に優先度の低いタスクでは急激に大きくなるものと予想される。この意味で X_i の分布関数 $R_i^\theta(t; \lambda)$ ($= \text{Prob}(X_i \leq t)$) を、システムに加わる負荷 $M^\theta(\lambda)$ のもとでのタスク T_i^θ の膨張処理時間分布と呼ぶ。

我々の近似の大きな方針は、まず初めに各タスクの膨張処理時間分布 $R_i^\theta(t; \lambda)$ の特性値である平均 $e_i^\theta(\lambda)$ と分散 $v_i^\theta(\lambda)$ を何等かの解析モデルで λ の関数と

して近似する。2章で述べたように、専用オンラインシステムはトランザクションの流れからみると、タスクを仮のサーバとする開待ち行列網モデルとみなせた。そこで、先に求めた各タスクの膨張処理時間分布の特性値を、この開待ち行列網モデルとみなした時の各タスクのサービス時間分布の特性値に対応させて使うことにより、トランザクションに関する定量的諸量をトランザクションの到着率の関数として解析しようとするものである。

3.3 タスクの膨張処理時間の平均と分散

CPU 処理, I/O 処理を実際に行うのは、CPU, I/O 装置等のハードウェア資源である。この解析では、CPU の数は1台とする。すなわち、この1台で OS およびタスクからの CPU 処理要求すべてを実行するものとする。これに対し、I/O 装置は各タスクからの I/O 処理要求を実行する専用の I/O 装置が存在するものとする。すなわち、各タスクからの I/O 処理要求に対しては、アクセスの競合はないものとする。アクセスの競合がある場合については、後述する。

I/O 装置に対しアクセスの競合がないことを仮定すると、確率変数 X_i の分布、すなわち、タスクの T_i^θ の膨張処理時間分布 $R_i^\theta(t; \lambda)$ に影響を与えるのはタスク T_i^θ より優先して実行される OS のオーバヘッド V とタスク T_j^θ ($j=1, 2, \dots, i-1$) の CPU 処理のみである。これらの高優先度の CPU 処理要求の平均発生率 λ_p と1件の要求に対する CPU 処理時間の平均 $1/\mu_p$ は、次式で与えられる。

$$\lambda_p = g_0(\lambda) + \sum_{j=1}^{i-1} g_j^\theta(\lambda) \cdot N_j. \quad (3)$$

$$1/\mu_p = (g_0(\lambda) \cdot s_0 + \sum_{j=1}^{i-1} (g_j^\theta(\lambda) \cdot \sum_{k=1}^{N_j} s_{jk})) / \lambda_p. \quad (4)$$

膨張処理時間分布 $R_i^\theta(t; \lambda)$ を求めるために、タスク T_i^θ において1件のトランザクションの処理が開始されてから、その処理が完全に終了するまでの間に、それより優先されて割込的に実行される CPU 処理要求が平均 λ_p のポアソン分布で発生し、その1件の CPU 処理時間の分布関数 $F_p(t)$ が平均 $1/\mu_p$ の指数分布、

$$F_p(t) = 1 - e^{-\mu_p t} \quad (5)$$

であるようなモデルを考える。図2にそのモデルの図式を示す。タスク T_i^θ の処理は CPU 処理と I/O 処理の逐次的な処理であるが、CPU 処理中に高優先度の CPU 処理要求が発生すると、タスク T_i^θ にお

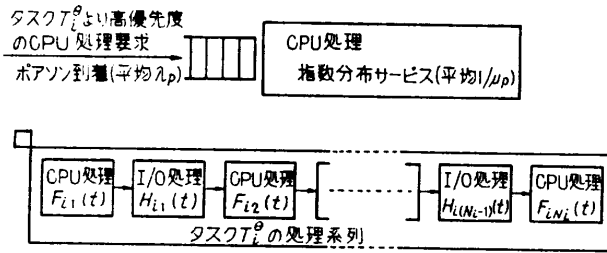


図2 膨張処理時間分布解析モデル図式
Fig. 2 Model for analyzing $R_{ij}^*(t; \lambda)$.

る CPU 処理は中断されて、高優先度の CPU 処理要求が先に処理されるものとする。しかし、高優先度の CPU 処理とタスク T_i^j の I/O 処理は、並行に実行できる。ここで、確率変数 $P_{ij}' (j=1, 2, \dots, N_i)$ と $D_{ik}' (k=1, 2, \dots, N_i-1)$ を、

P_{ij}' ; T_i^j において j 番目の CPU 処理を開始してから、その処理を完了するまでの時間、
 D_{ik}' ; T_i^j において k 番目の I/O 処理を開始してから、 $(k+1)$ 番目の CPU 処理を開始するまでの時間、

と定義すれば、確率変数 X_i はこれらの確率変数の和、

$$X_i = \sum_{j=1}^{N_i} P_{ij}' + \sum_{k=1}^{N_i-1} D_{ik}' \quad (6)$$

となる。したがって、まず確率変数 P_{ij}' , D_{ik}' の各々の分布について考える。

[1] 確率変数 P_{ij}' の分布

確率変数 P_{ij}' の分布関数を $RP_{ij}(t) (= \text{Prob}(P_{ij}' \leq t))$ とすると、 $RP_{ij}(t)$ は次式で表わされる。

$$RP_{ij}(t) = \int_0^t \sum_{n=0}^{\infty} P_n(x) \cdot B_n(t-x) dF_{ij}(x). \quad (7)$$

ここで、 $F_{ij}(x)$ はタスク T_i^j の j 番目の CPU 処理時間 (P_{ij}) が x 以内である確率 (式(1)) であり、 $P_n(x)$ は P_{ij} が x の時、その CPU 処理が高優先度の

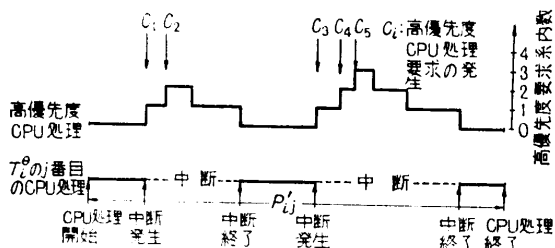


図3 タスク T_i^j の j 番目の CPU 処理
Fig. 3 The j -th CPU processing for the task T_i^j .
CPU 処理要求の発生により n 回中断される確率である。また $B_n(t)$ は 1 回の中断時間の分布関数を $B(t)$ とする時、この分布 $B(t)$ に従う n 個の確率変数の

和の分布関数である。図3はタスク T_i^j における j 番目の CPU 処理の開始から終了までの一例を示したタイムチャートである。この図において、1つの中断時間の分布は、高優先度の CPU 処理要求の発生とその処理時間の分布を、各々平均 λ_p のポアソン分布、平均 $1/\mu_p$ の指数分布 (式(5)) と仮定しているので、待ち行列モデル $M/M/1$ におけるサービス継続時間の分布に対応していることに注意されたい。この分布関数 $B(t)$ のラプラス・ステルチェス変換 (以下 $L-S$ 変換と略) $B^*(s) (= \int_0^{\infty} e^{-st} dB(t))$ は、次式の関数方程式を満たすことが知られている⁴⁾。

$$B^*(s) = \mu_p / (\mu_p + \lambda_p + s - \lambda_p \cdot B^*(s)). \quad (8)$$

$B_n(t)$ は n 個の中断時間の和の分布であるが、これらは独立と考えてよいので次式が成り立つ。

$$B_n(t) = \int_0^t B_{n-1}(t-x) dB(x) \quad (n \geq 2) \quad (9)$$

$B_n(t)$ の $L-S$ 変換 $B_n^*(s) (= \int_0^{\infty} e^{-st} dB_n(t))$ は、式(9)の両辺の $L-S$ 変換を行うと、たたみ込みの $L-S$ 変換は各 $L-S$ 変換の積により、次式となる。

$$B_n^*(s) = B_{n-1}^*(s) \cdot B^*(s) = B_{n-2}^*(s) \cdot B^*(s)^2 = \dots = B^*(s)^n. \quad (10)$$

また、中断回数の分布関数 $P_n(x)$ は、以下に示すくポアソン分布となる。(証明付録1)

$$P_n(x) = (\lambda_p x)^n e^{-\lambda_p x} / n!. \quad (11)$$

さて、式(7)の説明であるが、簡単に行うために分布関数 $F_{ij}(x)$ の密度関数 $f_{ij}(x)$ の存在を仮定すれば、タスク T_i^j の j 番目の CPU 処理時間が区間 $(x, x+dx)$ に入る確率は $f_{ij}(x)dx$ であり、そのとき n 回の中断が起る確率は $P_n(x)$ である。このとき、率変数 P_{ij}' はタスク T_i^j の j 番目の CPU 処理時間 x と n 回の中断時間の和となり、これらは互いに独立と考えてよいので、 n 個の中断時間の和の分布関数を $B_n(x)$ とすれば、 P_{ij}' の分布関数 $RP_{ij}(t)$ は次式で与えられる。

$$RP_{ij}(t) = \int_0^t \sum_{n=0}^{\infty} B_n(t-x) \cdot P_n(x) \cdot f_{ij}(x) dx. \quad (7')$$

この関数は一般的には、ステルチェス積分を用いて式(7)で表わされる。

P_{ij}' の分布関数 $RP_{ij}(t)$ の $L-S$ 変換を $RP_{ij}^*(s) (= \int_0^{\infty} e^{-st} dRP_{ij}(t))$ とすると、式(7)において、たたみ込みの $L-S$ 変換は各 $L-S$ 変換の積となり、整理すると以下のようになる。

$$RP_{ij}^*(s) = \sum_{n=0}^{\infty} \left\{ \int_0^{\infty} e^{-sx} dB_n(x) \right\}$$

$$\begin{aligned} & \cdot \int_0^\infty e^{-s \cdot x} P_n(x) dF_{ij}(x) \Big\} \\ &= \sum_{n=0}^\infty \left\{ B^*(s)^n \right. \\ & \quad \cdot \int_0^\infty e^{-(s+\lambda_p)x} \frac{(\lambda_p x)^n}{n!} dF_{ij}(x) \Big\} \\ &= \int_0^\infty e^{-(s+\lambda_p-\lambda_p B^*(s))x} dF_{ij}(x) \\ &= F_{ij}^*(s+\lambda_p-\lambda_p B^*(s)). \end{aligned} \tag{12}$$

ただし, $F_{ij}^*(s) = \int_0^\infty e^{-s \cdot t} dF_{ij}(t)$. (13)

また, P_{ij}' の平均 $E(P_{ij}')$ と分散 $V(P_{ij}')$ は, 以下に示す関係式から求めることができる.

$$E(P_{ij}') = [-dR P_{ij}^*(s)/ds]_{s=0}. \tag{14}$$

$$V(P_{ij}') = [d^2 R P_{ij}^*(s)/ds^2 - (dR P_{ij}^*(s)/ds)^2]_{s=0}, \tag{15}$$

これより, P_{ij}' の平均と分散は次式となる.

$$E(P_{ij}') = \frac{\mu_p}{\mu_{ij}(\mu_p - \lambda_p)}, \tag{16}$$

$$V(P_{ij}') = \frac{\mu_p^2 \sigma_{ij}^2}{(\mu_p - \lambda_p)^2} + \frac{2\lambda_p \mu_p}{\mu_{ij}(\mu_p - \lambda_p)^3}. \tag{17}$$

[2] 確率変数 D_{ik}' の分布

タスク T_i^0 において k 番目の I/O 処理を開始してから $(k+1)$ 番目の CPU 処理を開始するまでの時間 (図4参照) を表す確率変数 D_{ik}' の分布関数を $RD_{ik}(t) (= \text{Prob}(D_{ik}' \leq t))$ とする. I/O 処理の開始時点では, システムに存在する高優先度の CPU 処理要求の数 (以下, 高優先度の系内数と言う) は 0 である. タスク T_i^0 において k 番目の I/O 処理時間の分布関数は, $H_{ik}(t)$ としたが (式(2)), この I/O 処理時間が区間 $(x, x+dx)$ の間に入る確率は, $H_{ik}(x)$

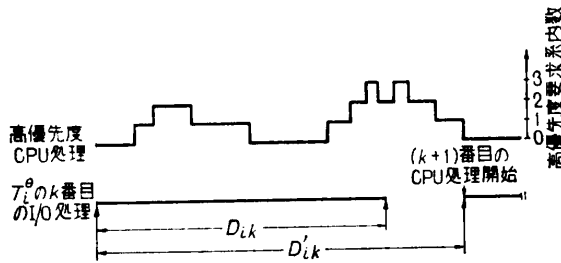


図4 タスク T_i^0 の k 番目の I/O 処理

Fig. 4 The k -th I/O processing for the task T_i^0 .

の密度関数を $h_{ik}(x)$ とすると, $h_{ik}(x)dx$ であり, この I/O 処理が終了した時, 高優先度の系内数が n である確率を $q_n(x)$ とする. この時, D_{ik}' は I/O 処理時間 x と高優先度の系内数が n の状態から始めて,

最初に系内数が 0 になるまでの時間 (確率変数 Z で表わす) との和である. これらは独立と考えてよいので, 確率変数 Z の分布関数を $A_n(t) (= \text{Prob}(Z \leq t))$ とすると, D_{ik}' の分布関数は次式で与えられる.

$$RD_{ik}(t) = \int_0^t \sum_{n=0}^\infty q_n(x) \cdot A_n(t-x) dH_{ik}(x). \tag{18}$$

ここで, $RD_{ik}(t)$, $A_n(t)$ の $L-S$ 変換を, それぞれ $RD_{ik}^*(s) (= \int_0^\infty e^{-s \cdot t} dRD_{ik}(t))$, $A_n^*(s) (= \int_0^\infty e^{-s \cdot t} dA_n(t))$ とし, $q_n(x)$ のラプラス変換を $q_n^*(s) (= \int_0^\infty e^{-s \cdot x} q_n(x) dx)$ とおくと, 式(18)から $RD_{ik}^*(s)$ は次式となる.

$$\begin{aligned} RD_{ik}^*(s) &= \sum_{n=0}^\infty \int_0^\infty e^{-s \cdot x} dA_n(x) \\ & \quad \cdot \int_0^\infty e^{-s \cdot x} q_n(x) dH_{ik}(x) \\ &= \sum_{n=0}^\infty \tau_{ik} A_n^*(s) q_n^*(s + \tau_{ik}). \end{aligned} \tag{19}$$

ただし, 式(2)より

$$dH_{ik}(x)/dx = \tau_{ik} e^{-\tau_{ik} x}. \tag{20}$$

$q_n(t)$ は高優先度の系内数が 0 である状態を起点として, t 時間後に高優先度の系内数が n になる確率であるが, これは待ち行列モデル $M/M/1$ での有限時間後の状態確率に対応している. この状態確率のラプラス変換は既に知られており⁵⁾, 本解析の場合には, 到着率, サービス率をそれぞれ λ_p, μ_p の $M/M/1$ と考えればよいので, $q_n^*(s)$ は以下の式で表わされる.

$$q_n^*(s) = \beta(s)^n (\beta(s) - \alpha(s)) / (\gamma(s)(1 - \alpha(s))). \tag{21}$$

ただし,

$$\gamma(s) = \sqrt{(s + \lambda_p + \mu_p)^2 - 4\lambda_p \mu_p}. \tag{22}$$

$$\alpha(s) = (s + \lambda_p + \mu_p + \gamma(s)) / (2\mu_p). \tag{23}$$

$$\beta(s) = (s + \lambda_p + \mu_p - \gamma(s)) / (2\mu_p). \tag{24}$$

確率変数 Z の分布関数 $A_n(t)$ は, 待ち行列モデル $M/M/1$ における系内数が n である任意の時点からのサービス継続時間の分布関数に対応しており, これはまた, $M/M/1$ における系内数が 0 である任意の時点からのサービス継続時間の n 個の和の分布に等しいことに注意されたい. n 個のサービス時間は各々独立と考えてよいので, 分布関数 $A_n(t)$ の $L-S$ 変換 $A_n^*(s)$ は, 高優先度系のサービス継続時間の分布関数の $L-S$ 変換 $B^*(s)$ を用いて (式(8)), 次式で表わされる.

$$A_n^*(s) = B^*(s)^n. \tag{25}$$

ここで, 式(21)と式(25)を式(19)に代入して, 整理すると $RD_{ik}^*(s)$ は次式となる.

$$\begin{aligned}
 RD_{ik}^*(s) &= \sum_{n=0}^{\infty} \frac{\tau_{ik} B^*(s)^n \beta(s+\tau_{ik})^n (\beta(s+\tau_{ik}) - \alpha(s+\tau_{ik}))}{\gamma(s+\tau_{ik}) (1 - \alpha(s+\tau_{ik}))} \\
 &= \tau_{ik} / (\mu_p (1 - B^*(s) \beta(s+\tau_{ik})) (\alpha(s+\tau_{ik}) - 1)). \quad (26)
 \end{aligned}$$

ただし、 $\alpha(s)$, $\beta(s)$, $\gamma(s)$ は各々式 (23), (24), (22) であり、 $B^*(s)$ は式 (8) で与えられる。

確率変数 D_{ik}' の平均 $E(D_{ik}')$ と分散 $V(D_{ik}')$ は、式 (26) より次のように求め得る。(計算過程略)

$$\begin{aligned}
 E(D_{ik}') &= [-dRD_{ik}^*(s)/ds]_{s=0} \\
 &= \frac{\gamma(\tau_{ik}) + \mu_p - \lambda_p - \tau_{ik}}{2\tau_{ik}(\mu_p - \lambda_p)}. \quad (27)
 \end{aligned}$$

$$\begin{aligned}
 V(D_{ik}') &= [d^2RD_{ik}^*(s)/ds^2 - (dRD_{ik}^*(s)/ds)^2]_{s=0} \\
 &= \left\{ \frac{\gamma(\tau_{ik}) + \mu_p - \lambda_p - \tau_{ik}}{2\tau_{ik}(\mu_p - \lambda_p)} \right\}^2 + \frac{\tau(\tau_{ik}) - \mu_p - \lambda_p - \tau_{ik}}{\tau_{ik}\tau(\tau_{ik})(\mu_p - \lambda_p)} \\
 &\quad + \frac{\mu_p(\gamma(\tau_{ik}) - \mu_p + \lambda_p - \tau_{ik})}{\tau_{ik}(\mu_p - \lambda_p)^3}. \quad (28)
 \end{aligned}$$

ただし、

$$\gamma(\tau_{ik}) = \sqrt{(\mu_p + \lambda_p + \tau_{ik})^2 - 4\lambda_p\mu_p}. \quad (29)$$

[3] 確率変数 X_i の分布

確率変数 X_i は式 (6) で示したように、確率変数 P_{ij} ($j=1, 2, \dots, N_i$), D_{ik}' ($k=1, 2, \dots, N_i-1$) の和である。これらはすべて各々独立なので (付録 2 で説明)、確率変数 X_i , すなわち、タスク T_i^{θ} の膨張処理時間の平均 $e_i^{\theta}(\lambda)$ と分散 $v_i^{\theta}(\lambda)$ は式 (16), (17), (27), (28) から次式となる。

$$\begin{aligned}
 e_i^{\theta}(\lambda) &= \sum_{j=1}^{N_i} \frac{\mu_p}{\mu_{ij}(\mu_p - \lambda_p)} \\
 &\quad + \sum_{k=1}^{N_i-1} \frac{\gamma(\tau_{ik}) + \mu_p - \lambda_p - \tau_{ik}}{2\tau_{ik}(\mu_p - \lambda_p)}. \quad (30)
 \end{aligned}$$

$$\begin{aligned}
 v_i^{\theta}(\lambda) &= \sum_{j=1}^{N_i} \left\{ \frac{\mu_p^2 \sigma_{ij}^2}{(\mu_p - \lambda_p)^2} + \frac{2\lambda_p \mu_p}{\mu_{ij}(\mu_p - \lambda_p)^3} \right\} \\
 &\quad + \sum_{k=1}^{N_i-1} \left[\frac{\{\gamma(\tau_{ik}) + \mu_p - \lambda_p - \tau_{ik}\}^2}{2\tau_{ik}(\mu_p - \lambda_p)} \right. \\
 &\quad \left. + \frac{\gamma(\tau_{ik}) - \mu_p - \lambda_p - \tau_{ik}}{\tau_{ik}\tau(\tau_{ik})(\mu_p - \lambda_p)} \right. \\
 &\quad \left. + \frac{\mu_p(\gamma(\tau_{ik}) - \mu_p + \lambda_p - \tau_{ik})}{\tau_{ik}(\mu_p - \lambda_p)^3} \right] \quad (31)
 \end{aligned}$$

ただし、

$$\gamma(\tau_{ik}) = \sqrt{(\mu_p + \lambda_p + \tau_{ik})^2 - 4\lambda_p\mu_p}. \quad (29)'$$

上記の式における λ_p, μ_p はタスク T_i^{θ} より優先して実行される CPU 処理要求の発生率と、その平均処理

時間の逆数であるが、これは式 (3), (4) で示したようにシステムへのトランザクションの到着率 λ の関数として表わされる。

また、独立な確率変数の和の分布の $L-S$ 変換は、それぞれの $L-S$ 変換の積で表わされるので、タスク T_i^{θ} の膨張処理時間分布 $R_i^{\theta}(t; \lambda)$ の $L-S$ 変換 $R_i^{\theta*}(s) \left(= \int_0^{\infty} e^{-st} dR_i^{\theta}(t; \lambda) \right)$ は次式となり、この式に式 (12), (26) を代入すればよい。

$$R_i^{\theta*}(s) = \prod_{j=1}^{N_i} RP_{ij}^*(s) \cdot \prod_{k=1}^{N_i-1} RD_{ik}^*(s). \quad (32)$$

[4] I/O 装置へのアクセスの競合

I/O 装置に対するアクセスの競合の形態は、システムごとに種々の形態があり、その待ち行列モデルとしての構造も複雑であるために一概に論じえない。特別な場合として、もし競合の起る I/O 系が部分モデルとして、待ち行列モデル $M/M/1$ としてとらえることができるならば、アクセスの競合による待ち合せと I/O 処理時間の和の分布が、指数の形* となるので、これまでと同様の解析で整った解を簡単に導くことができるが、ここでは省略する。一般的には、アクセスの競合が起る I/O 系を部分的に何等かの解析モデル (例えば、有限呼源の $M/M/1$ モデル等) で競合による待ち合せを含めた I/O 処理時間の平均 ($1/\tau_{ik}'$) を求め、これまでの議論におけるタスク T_i^{θ} での k 番目の I/O 処理時間分布 $H_{ik}(t)$ を (式 (2)), 平均 $1/\tau_{ik}'$ の指数分布と仮定するならば、これまでの議論はアクセスの競合が起る場合にも適応できる。

3.4 性能解析

専用オンラインシステムはトランザクションの流れからみると、開待ち行列網モデルとみなせた。ここでは、この開待ち行列網モデルの各サービスステージ、すなわち、各タスクのサービス時間分布の特性値を先に導いた各タスク T_i^{θ} の膨張処理時間分布 $R_i^{\theta}(t; \lambda)$ の特性値に対応させて使うことにより、システムの総合的な性能を表わす指標およびシステムの定量的諸量を解析する。開待ち行列網モデルは、呼の到着がポアソン到着で各サービスステージのサービス時間が指数分布である場合には、各サービスステージの出力がまたポアソンプロセスとなるので整った解を導けるが、一般分布の場合には、マルコフ性が失われ解を導くことができない。そこで近似という実際の見地から、以下の 2 つの仮定をおき、各サービスステージを各々単一待ち行列モデル $M/G/1$ とみなして解析する。

(仮定 a) 各タスク T_i^{θ} のトランザクションの到

* $1 - (1 - \rho)e^{-\rho t}$. ただし、 ρ は I/O 装置に対する利用率。

着分布を平均 $g_i^0(\lambda)$ の指数分布, その処理時間分布を平均 $e_i^0(\lambda)$, 分散 $v_i^0(\lambda)$ の一般分布とする.

(仮定 b) 各タスクへのトランザクションの到着は, それぞれ相互に独立であり, また, 各タスクへのトランザクションの到着とその処理時間も, それぞれ相互に独立であるとする.

[1] スループットとボトルネックタスク

上記の仮定により, 各タスクは待ち行列モデル $M/G/1$ に支配されると考えると, タスク T_i^0 の平衡条件は, $M/G/1$ の平衡条件から,

$$g_i^0(\lambda) \cdot e_i^0(\lambda) < 1. \quad (33)$$

となる. また, 全体システムの平衡条件は, 各タスクの平衡条件の論理積, すなわち,

$$\bigwedge_{i=1}^n (g_i^0(\lambda) \cdot e_i^0(\lambda) < 1). \quad (34)$$

ただし,

$$\bigwedge_{i=1}^n a_i \stackrel{\text{def}}{=} a_1 \wedge a_2 \wedge \dots \wedge a_n.$$

となり, この条件を満たす最大の λ がスケジューリング方策 θ のもとでのスループット $TH(\theta)$ となる.

$$TH(\theta) = \max(\lambda \mid \bigwedge_{i=1}^n (g_i^0(\lambda) \cdot e_i^0(\lambda) < 1)). \quad (35)$$

この $TH(\theta)$ は意味的にはスケジューリング方策 θ のもとでのシステムの限界能力を示す. すなわち, システムが耐えうる最大のトランザクションの到着率である. また, ボトルネックタスクは λ を大きくしていくとき, 各タスク T_i^0 の平衡条件 (式(33)) を最初にくずすタスクである. 言い換えれば, λ が $TH(\theta)$ を越える場合には, このボトルネックタスクに対するトランザクションの待ち行列が増大し続けることを意味する.

[2] 最大スループット

n 個のタスクからなるマルチタスクシステムにおけるスケジューリング方策は, 全部で $n!$ があるが, これを $\theta = \{\theta_1, \theta_2, \dots, \theta_{n!}\}$ で表わす. 各スケジューリング方策に対して, 式(35)からスループットが定まるが, これらの最大値をシステムの最大スループットと定義し, TH_{\max} で表わす.

$$TH_{\max} = \max(TH(\theta_1), TH(\theta_2), \dots, TH(\theta_{n!})). \quad (36)$$

ここで, 最大スループットを実現するためのスケジューリング方策の集合を最適スケジューリング方策 θ_{op} と定めると, θ_{op} は次式のように表わすことができる.

$$\theta_{op} = \{\forall \theta \in \theta \mid TH(\theta) = TH_{\max}\}. \quad (37)$$

[3] システムの定量的諸量

各タスク T_i^0 におけるトランザクションの平均待

ち時間 ω_i^0 , 平均待ち行列長 q_i^0 等の平衡状態における統計的諸量は, 待ち行列 $M/G/1$ モデルの諸結果を利用して, 以下のように近似できる.

$$\omega_i^0(\lambda) = \frac{g_i^0(\lambda) \cdot e_i^0(\lambda)^2}{2(1 - g_i^0(\lambda) \cdot e_i^0(\lambda))} \left(1 + \frac{v_i^0(\lambda)}{e_i^0(\lambda)^2} \right). \quad (38)$$

また, タスク T_i^0 の系内数が 0 でない確率, すなわち, タスク T_i^0 の占有率 $\rho_i^0(\lambda)$ は次式で近似できる.

$$\rho_i^0(\lambda) = g_i^0(\lambda) \cdot e_i^0(\lambda). \quad (39)$$

4. シミュレーションとの比較

図5に示すマルチタスクシステムを例題として取りあげ, 本論文での解析的方法とシミュレーションとの比較を示す. この例題システムは5個のタスクから構成され, 各タスクの処理特性は図中に示す通りである. 例えば, タスクBはCPU処理, I/O処理, CPU処理の順に逐次的に実行され, その処理時間分布はこの順序で平均2msの位相2のアーラン分布, 平均20msの指数分布, 平均10msの単位分布であることを示す. また, 各タスクで1件のトランザクションの処理が実行された時, 他のタスクへ渡されるトランザクションの比率をタスク間のアークに示す. ただし, この例題ではOSの負荷は考慮していない.

この例では, タスク数が5個なのでスケジューリング方策は全部で $5!$ 通りがあるが, このすべてに対し解析的方法で求めたスループットを, スループットの小さい順に並べて図6に示す. また, 同図にいくつかのスケジューリング方策に対し, シミュレーションで求めたスループットをプロットした. 表1は図6でプロットした点に関するスケジューリング方策, スループット, ボトルネックタスクを解析計算, シミュレーションそれぞれについて示したものである. また, 図7はある1つのスケジューリング方策のもとで, トランザクションの到着率 λ と各タスクの占有率, 平均待ち行列長との関係を示したものである.

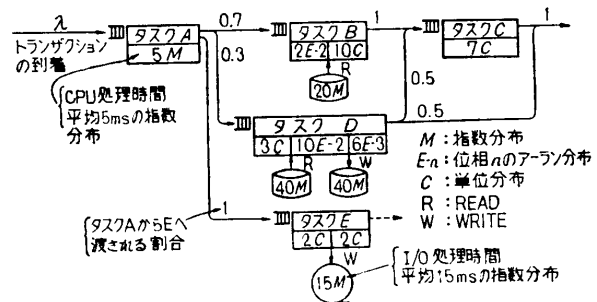


図5 評価のための例題システム

Fig. 5 Example system for evaluation.

表1 スケジューリング方策とスループット, ボトルネックタスク

Table 1 Relation between priority policy and throughput, bottleneck task.

	タスク優先度 高←→低	解析モデル		シミュレーション	
		スループット	BNT*	スループット**	BNT
a	DBEAC	33.61	D	33.0 ~ 33.5	D
b	DEABC	31.09	B	31.0 ~ 31.5	B
c	ADECB	26.70	B	26.0 ~ 26.5	B
d	DBCAE	26.17	E	25.75 ~ 26.25	E
e	EACBD	23.77	D	23.5 ~ 24.0	D

* ボトルネックタスク

** 下限は定常状態を確認 上限は飽和状態を確認

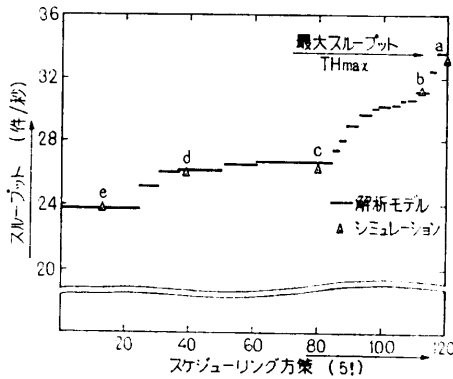


図6 スケジューリング方策とスループット

Fig. 6 Relation between priority policy and throughput.

スループットに関しては、図6、表1から判るように解析計算とシミュレーションの結果が極めて一致しており、その誤差は±3%以内であった。また、各スケジューリング方策に対するボトルネックタスクも一致した。さらに、図7に示すように各タスクの占有率、平均待ち行列長も比較的良好に一致している。タスクの占有率は、いわゆる設備の利用率 (utilization) と同様な性格を持つ性能指標であるが、この占有率に対するシミュレーションと解析計算の相対誤差を調べると、優先度が低くCPUバウンドであるタスク程大きな誤差(最大で約15%)が表われた。しかし、トラフィックの増大に伴って、すなわちタスクの占有率が1に近づくにつれ、この相対誤差は収束する傾向が確認された。このために、この占有率をもとに計算するスループットに関しては、非常によい精度で結果が導かれたものとする。これらの誤差は一般には、解析におけるいくつかの仮定による誤差の相乗によって表われるものであるが、どの仮定による誤差が結果としての指標にどの程度影響をあたえているかは、評価

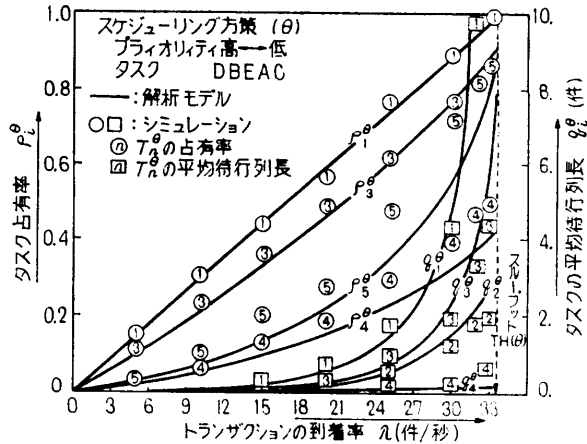


図7 タスクの占有率と平均待ち行列長

Fig. 7 Utilization and mean queue length of task.

対象システムの実特性によっても変わるものであり、一概に論ずることは難しい。この誤差の限界、収束性については今後の課題である。

なお、シミュレーションモデルはGPSS(約150ブロック)で作成した。また、解析計算においても、手計算で行うには多くの時間を費やすことが予想されたので、任意のマルチタスクシステムの評価に適応できるようなプログラム(FORTRAN, 約300ステップ)を作成して、解析した。

5. むすび

専用オンラインシステムの性能は、極めて多くの要因によって支配されるが、本論文では、これらの要因をタスクの処理特性、オペレーティングシステムの負荷、各タスクの実行優先度に限定した上で、解析モデルによってシステムの性能を表わす指標を導いた。マルチプログラミング制御下の各タスクの処理は、システム内の輻湊のために時間的に引き延ばされるが、本解析では、この引き延ばされる時間の分布をタスクの膨張処理時間分布と定義し、その平均と分散を入力トラフィックと上記の基本的要因の関数として導いた。そして、この結果を使い、各タスクをトランザクションの待ち行列に対するサーバとみなすことにより、システムの定量的諸量やシステムの限界能力等を求める近似式を導いた。さらに、システムのボトルネックや最適スケジューリング方策についても導いた。また、ある例題システムを対象としてシミュレーションを行うことにより、本解析の有効性を確めた。ただし、この近似解析の誤差の範囲、収束性については何もふれ

ていない。この問題については、実際のシステムへの適応も含めて別の機会に報告したい。

最後に、本研究について貴重なご意見、ご指導をいただいた日本電気中央研究所の原田紀夫博士に深く感謝します。

参 考 文 献

- 1) Buzen, J. P.: Computational Algorithms for Closed Queuing Networks with Exponential Servers, Communications of the ACM, Vol. 16, No. 9, pp. 527-531 (1973).
- 2) Lipsky, L. and Church, J. D.: Applications of a Queuing Network Model for a Computer System, ACM Computing Surveys, Vol. 19, No. 3, pp. 205-221 (1977).
- 3) 石黒: Gate のある M/M/m 待行列ネットワークモデルの研究, 情報処理, Vol. 19, No. 7, pp. 614-620 (1978).
- 4) 西田: 待ち行列の理論と応用, p. 132, 朝倉書店 (1971).
- 5) 本間: 待ち行列理論, p. 31, 理工学社 (1966).
- 6) Jakson, J. R.: Jobshop like queuing systems, Management Science, Vol. 10, No. 1, p. 31 (1963).
- 7) 原田: 計算機システムの数学的基礎理論, 信学会誌, Vol. 59-D, No. 12, pp. 843-850 (1976).
- 8) Gordon, W. J. and Newell, G. F.: Closed queuing systems with exponential servers, Oper. Res. 15. 2, pp. 254-265 (1967).
- 9) 本山, 中箴, 大町: 循環待行列モデルによるオンラインシステムの性能評価, 情報処理学会第19回全国大会論文集, pp. 983-984 (1977).
- 10) Denning, P. J. and Buzen, J. P.: The Operational Analysis of Queuing Network Models, ACM Computing Surveys, Vol. 10, No. 3, pp. 225-262 (1978).
- 11) Muntz, R. R.: Queuing Networks: A Critique of the State of the Art and Directions for the Future, ACM Computing Surveys, Vol. 10, No. 3, pp. 353-360 (1978).

付 録 1

中断回数分布 $P_n(x)$

タスクの CPU 処理時間が x のとき、1 回の中断が起る確率 $P_1(x)$ を考える。今、確率 P を CPU 処理開始後 y 時間後に初めて中断が発生し、その中断が残り CPU 処理が再開されて残りの $(x-y)$ 時間には何も中断が起らない確率とする。高優先度の CPU 処理要求の発生は平均 λ_p のポアソン到着を仮定しているので、区間 $[y, y+dy]$ に初めて中断が起る確率は $\lambda_p e^{-\lambda_p y} dy$ であり、また、残りの $(x-y)$ 時間に中断

が起らない確率は $e^{-\lambda_p(x-y)}$ である。したがって、その同時確率 p は、

$$p = \lambda_p e^{-\lambda_p y} dy \cdot e^{-\lambda_p(x-y)} = \lambda_p e^{-\lambda_p x} dy. \quad (i)$$

これを y について積分すると、すなわち $P_1(x)$ は、

$$P_1(x) = \int_0^x \lambda_p e^{-\lambda_p x} dy = \lambda_p x e^{-\lambda_p x}. \quad (ii)$$

となる。同様に、 $P_n(x)$ は y 時間後に 1 回中断し、残りの $(x-y)$ 時間に $(n-1)$ 回中断するものと考えればよいので次式が成り立つ。

$$P_n(x) = \int_0^x \lambda_p e^{-\lambda_p y} \cdot P_{n-1}(x-y) dy \quad (n \geq 2). \quad (iii)$$

この式から $P_n(x)$ のラプラス変換 $P_n^*(s)$ を求めると、右辺はたたみ込みのラプラス変換なので、

$$\begin{aligned} P_n^*(s) &= \int_0^\infty e^{-sx} \lambda_p e^{-\lambda_p x} dx \cdot P_{n-1}^*(s) \\ &= \frac{\lambda_p}{s + \lambda_p} \cdot P_{n-1}^*(s) = \dots = \left(\frac{\lambda_p}{s + \lambda_p} \right)^{n-1} P_1^*(s). \end{aligned} \quad (iv)$$

また、 $P_1(x)$ のラプラス変換は、式(ii)から、 $P_1^*(s) = \lambda_p / (s + \lambda_p)^2$ となり、これを式(iv)に代入して、ラプラス逆変換によって $P_n(x)$ を求めると、

$$P_n(x) = (\lambda_p x)^n e^{-\lambda_p x} / n!. \quad (v)$$

付 録 2

確率変数 P_{ij}' , D_{ik}' の独立性
確率変数 S_{ij} , U_{ik} を、

S_{ij} : タスク T_i^j の j 番目の CPU 処理中に高優先度 CPU 処理要求発生のために、中断させられる総時間 ($j=1, 2, \dots, N_i$).

U_{ik} : タスク T_i^j の k 番目の I/O 終了後、次の CPU 処理 ($(k+1)$ 番目の CPU 処理) が始まるまでの時間 ($k=1, 2, \dots, N_i-1$).

とすると、 P_{ij}' , D_{ik}' は、

$$P_{ij}' = P_{ij} + S_{ij}. \quad (vi)$$

$$D_{ik}' = D_{ik} + U_{ik}. \quad (vii)$$

ここで、すべての j, k に対して、 P_{ij} と D_{ik} , P_{ij} と U_{ik} , D_{ik} と S_{ij} , および S_{ij} と U_{ik} は、互いに独立である。また、 $j \neq l$ であるようなすべての j, l に対し、 P_{ij} と P_{il} , P_{ij} と S_{il} , S_{ij} と S_{il} , さらに、 $k \neq m$ であるようなすべての k, m に対し、 D_{ik} と D_{im} , D_{ik} と U_{im} , U_{ik} と U_{im} もまた、互いに独立である。(ただし、 $l=1, 2, \dots, N_i$, $m=1, 2, \dots, N_i-1$) したがって、確率変数 $P_{ij}' (j=1, 2, \dots, N_i)$, $D_{ik}' (k=1, 2, \dots, N_i-1)$ は、すべて互いに独立である。

(昭和 55 年 2 月 25 日受付)

(昭和 55 年 4 月 22 日採録)