

第II章 バッチ処理システム

第Ⅱ章 バッチ処理システム

学習目標

1. 対話型処理、リアルタイム処理（オンライントランザクション処理やリアルタイム制御処理）と比較しながら、バッチ処理の特徴、長所、短所を理解させる。
2. バッチ処理の形態の違いについて理解させる。
3. バッチ処理オペレーティングシステムの機能の概要と特徴を理解させる。
4. ジョブ制御言語を使って簡単なジョブを定義させる。
5. バッチ処理プログラムとオンライントランザクション処理プログラムの違い、ファイル構造との対応を理解させる。
6. データ構造とプログラムの制御構造を対応させてプログラムを設計させる。

全体概要

1980年代初期からのパソコンの普及およびそれ以降の定着、80年代半ばからのワークステーションの登場、さらにLANの普及とともにネットワークコンピューティングの時代が到来し、ますますダウンサイジングに拍車がかかっている。従来の汎用コンピュータは、90年代のコンピュータ環境がクライアントサーバモデルにとって代わられる状況下で相当の勢力の衰えを見せている。このような時代背景にあり、バッチ処理システムがかすんでしまっている。

しかしながら、メインフレームコンピュータ企業の売り上げが相対的には全盛期の2割から3割程度に落ちてきているとはいうものの、まだまだ収入の大部分は汎用コンピュータによっていることも否めない。

そもそもコンピュータプログラムは、人間の操作と深く係わりを持つ知的な処理と、アルゴリズム的に大量の演算を繰り返し行う処理とに分けられよう。前者はパソコンやワークステーションにとって代わられているが、後者はやはり相当な大型の計算能力を持つコンピュータでないと迅速な処理ができない。

クライアントサーバ時代にあっても、汎用コンピュータは大型の計算処理や大型の基幹データベースアクセス用のマシンとして今後も活躍の道は残っているであろう。

情報化時代というのは、大型処理と人間に近い部分での対話処理との共存によって成り立つこともあり、バッチ処理の重要性を忘れてはならないはずである。

再度基本に立ち戻って、バッチ処理の形態をキチンと理解しておく必要がある。

内容のあらまし

節 項	内 容
<p>1. バッチ処理</p> <p>(1) バッチ処理とその特徴</p> <p>(2) バッチ処理システムの例</p> <p>(3) バッチ処理の形態</p> <p>(4) 対話処理におけるバッチ処理</p>	<p>バッチ処理の定義、 バッチ処理の本質 他の処理方式との比較</p> <p>大学入試センターの試験の採点と集計の例 物品要求処理からの月間物品管理票作成処理</p> <p>オープンバッチ処理、 クローズドバッチ処理 カフェテリア方式</p> <p>センターバッチ処理、 リモートバッチ処理</p>
<p>2. バッチ処理のオペレーティングシステム</p> <p>(1) バッチ処理機能の実現</p>	<p>オペレーティングシステムの基本構成 オペレーティングシステムの狙い バッチ処理オペレーティングシステム 他のオペレーティングシステムとの比較</p>
<p>(2) ジョブ制御</p>	<p>ジョブ制御言語、 ジョブの構成 ワークステーションとパソコン</p>
<p>3. 主要用語</p>	

1. バッチ処理

(1) バッチ処理とその特徴

① バッチ処理の定義

バッチ処理は、一括処理ともいい、データ処理においてある一定期間にデータを収集・蓄積しておき、それらを一括して処理する方式をいう。この処理形態は、大量のデータ処理を行うような長時間コンピュータを使用したり、定期的にデータ処理を行ったり、時間の制約がないような場合に良く利用される。

この処理の狙いは、コンピュータ資源 [CPU、主メモリ、外部記憶装置など] を最大の効率で使用するにより、コンピュータの生産性を最も上げようとするところにある。

そのためには、各資源を安定的に使う形態であること、全ての資源がジョブによって効率良く利用されることが求められる。

これを実現するための最良な方法は、投入された全てのジョブをジョブキューに蓄え、その中から資源への要求事項を満足し、並行的に実行できるジョブを選択することである。これはつまるところジョブの多重度を上げることに他ならない。

② バッチ処理の本質

他の処理方式 (対話型、オンライントランザクション処理型など) との違いは、基本的にプログラムの処理中に人との介在が少ないことである。もともと商用コンピュータはバッチ処理によりできるだけ短時間に多くのジョブ処理を行えるよう設計された経緯がある。非常に高価であったコンピュータを、人手が介在するアイドル時間を極力減らしてコンピュータ資源 (特にCPUやディスク装置、磁気テープなど) の利用率を高めるために、ジョブの処理順序を計画的に決めたり、人手が必要な領域を自動化する改善が積み重ねられてきた。

- ・ 大量の計算処理

バッチ処理では、多量なデータ処理の連続的な実行において機能 (生産性) を発揮する。

- ・ 確定した実行手順

一度計算処理に入ると処理の終了まで人間のコントロールがきかないため、予め実行手順を確定しておく必要がある。

③ 他の (対話型、オンライントランザクション型) 処理方式との比較

バッチ処理での人手の介在は、計算ジョブの受付処理、ジョブをコンピュータに投入する作業、オペレータ操作卓でのOSからの問い合わせなどに対する応答、周辺装置などのセッティング、コンソールログのチェック、ジョブの結果と入力ジョブとの突合せなどである。コンソールに対する以外には対話は発生しない。また、バッチ処理の初期、中期くらいまでの時代には、コンピュータの立ち上げと終了、テープのマウント作業においても相当人手が介在したが、コンピュータの生産性の向上のため次第に自動化していった。とにかく単位時間当たりの処理量を最大限とし、コンピュータ資源利用を最大化するようなパフォーマンスを提供することに注力される。

対話型処理では、計算処理に必要なデータの入力や編集、コマンドの発行、途中結果の確認といった部分がコンピュータとの対話となる。この処理は、初期はバッチ処理における人手の空き時間をなくす、ソフトウェア開発の生産性が向上するなど、主としてTSSとしての効果を上げてきた。TSS端末、グラフィック端末、業務用専用端末の利用効果が評価をされた。

ワークステーションやパソコンが登場すると、対話型のコンピューティングの意義に大きな革命が起こった。利用者のプログラムを開発するための時間よりは、出来合いの商用パッケージを使ったコンピュータ利用である。このシステムでは、人間の思考過程のスピードにあったパフォーマンスが提供されればよい。

オンライントランザクション処理では、業務の核部分の処理は中央集中コンピュータで、要求

業務の入力と処理結果の表示は通信回線を介して遠隔地にある端末で行う。このシステムでは、要求が即時処理されることに最大の注力が払われる。

図表Ⅱ－１ 処理形態の比較

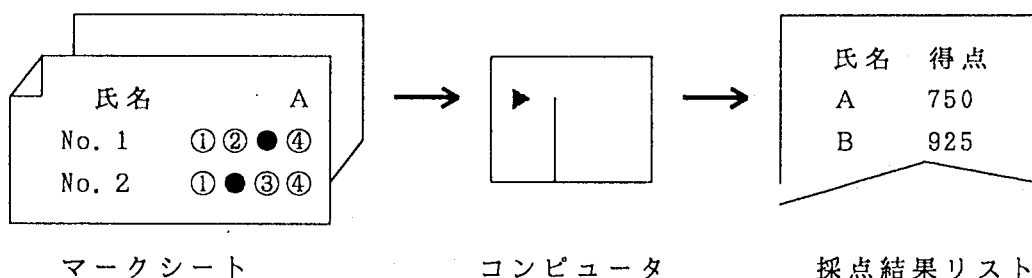
	処理の例	情報のやりとり	処理タイミング	データ量	即時
バッチ	蓄積された売上傳票ファイルを逐次コンピュータに読み込み、計算処理した後売上集計リストを出力する。	処理中には、人との情報のやりとりはない。	データを蓄積して処理の準備完了時	多量	なし
対話型	科学技術シミュレーションプログラムにおいて、計算条件をセットする。入力ミスを即時に訂正する。	処理中でも、人との情報のやりとりが頻繁	主に、データの発生時点	小量	あり
オンライン	現金自動支払機から出金額と口座番号が知らされ、現金引きだし処理とコンピュータへの記帳が行われる。	X	データの発生時点	小量	あり

(注) 第二種共通テキスト情報処理システム 中央情報教育研究所 を参考に作成

(2) バッチ処理システムの例

① 試験の採点と集計 (図表Ⅱ－2)

図表Ⅱ－2 試験 (大学入試、予備校の模試、国家資格試験など) の採点と集計の例



(注) 第二種共通テキスト情報処理システム 中央情報教育研究所 を参考に作成

② 受注製品開発に関連する経費要求処理から月次実績表を作成する処理 (図表Ⅱ－3)

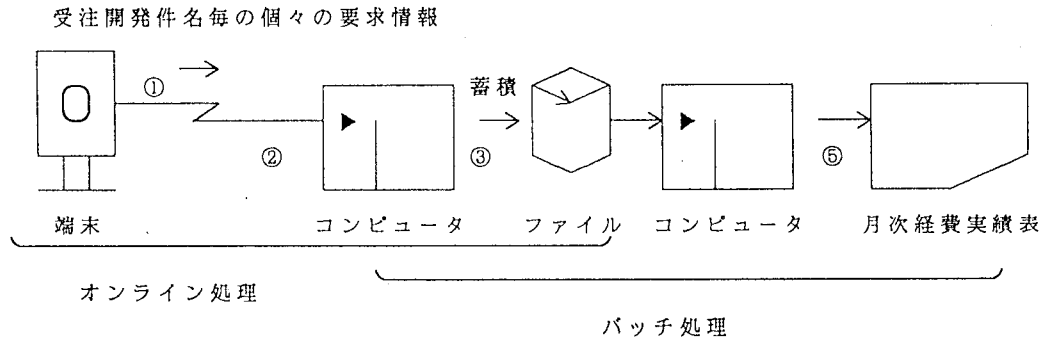
図表Ⅱ－3において下記のような順で処理を行う例を掲げる。

受注している製品開発件名ごとに、

- ・ 開発過程で調達あるいは必要とする費目と費用を端末に投入する。
- ・ 予算と調達、経費実績を表示し、調達、経費要求伝票を作成する。
- ・ ファイルに各種調達実施処理の記録をする。
- ・ ファイル内に蓄積された情報を読み込んでバッチ処理する。
- ・ 月次経費実績表を作成する。

予めオンライントランザクション処理でデータを蓄積してから一括処理する点が上記①と異なっている。

図表Ⅱ-3 月次調達、経費実績表作成処理



(注) 第二種共通テキスト情報処理システム 中央情報教育研究所 を参考に作成

(3) バッチ処理の形態

① オープンバッチ処理とクローズドバッチ処理およびカフェテリア方式

利用者自らがジョブの投入、コンソールの操作、結果の取り出しまでのマシン操作を行う方をオープンバッチ処理、ジョブの受付があり専任のオペレータに操作を依頼し結果だけを受け取る方をクローズドバッチ処理という(図表Ⅱ-4)。

カフェテリア方式では、バッチ登録を利用者自らがを行い、マシンの操作は専任のオペレータが行う。処理結果は利用者が受け取る(図表Ⅱ-4)。

② センタバッチ処理とリモートバッチ処理

ジョブを計算センターのマシンに直接投入する方をセンターバッチ処理、遠隔地からジョブを入力し通信回線を介してセンタマシンに投入することができる。遠隔地にはジョブ投入用カードリーダー、結果を出力するプリンタ装置を設置し、バッチ処理を実行する方式をリモート(遠隔)バッチ処理という。

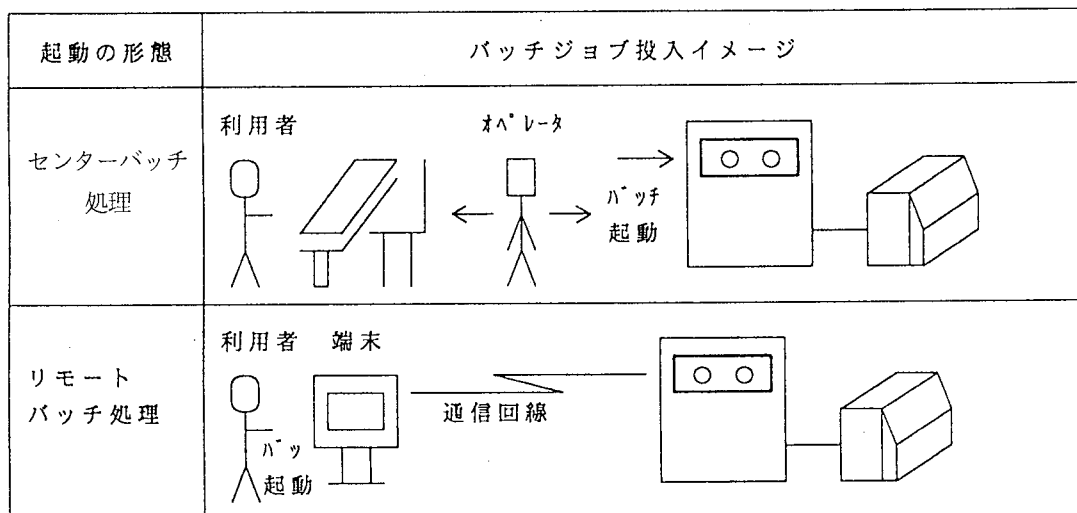
また、計算センターやリモートバッチステーションからだけでなく、TSS端末からもバッチジョブを投入することができる。

図表Ⅱ-4 各バッチ処理の比較

処理形態	バッチ投入者とオペレーション	備 考
オープンバッチ処理		ユーザ時間帯で借り切り、全てのオペレーションを行う。マシンの立ち上げ、電源断、ディスクの準備、MTのマウント/デマウント、ジョブの投入、コンソールの操作ログ情報一切を行う。
クローズドバッチ処理		ユーザは、作業手順書、プログラムカード、媒体を添付しオペレータに渡す。結果を受け取る。
カフェテリア方式		ユーザはジョブの出入りの部分を自分でを行い、コンピュータに直接触れる部分をオペレータが行う。

(注) 第二種共通テキスト情報処理システム 中央情報教育研究所 を参考に作成

図表Ⅱ-5 センターバッチ処理とリモートバッチ処理



(注) 第二種共通テキスト情報処理システム 中央情報教育研究所 を参考に作成

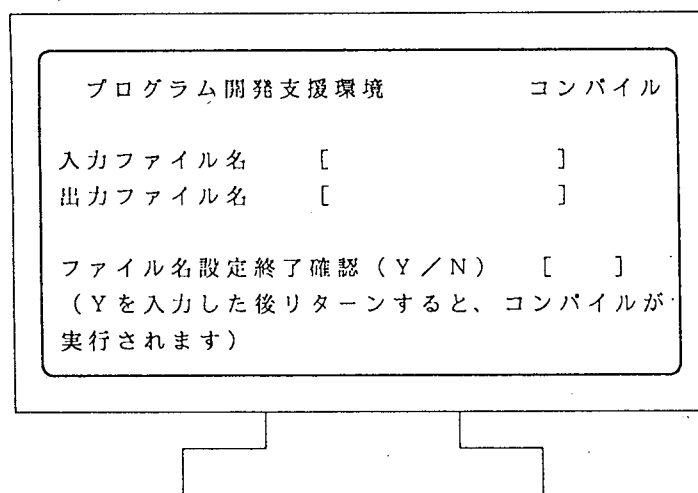
(4) 対話処理におけるバッチ処理

バッチ処理を行うに当たり、必要な情報を対話処理で入力することができ、利用者の手元でジョブ投入が可能となる。これを対話処理におけるバッチ処理という。

1970年代までは、例えばプログラムのソースコードやプログラムの実行時に必要となるデータさらには処理の順序情報を指定するジョブ制御言語などを一緒にパンチカードなどにまとめてコンピュータに投入していた。80年代にはいるとTSS端末の普及に伴いプログラムやデータの変更あるいは処理順序の変更などもそこで行うことができるようになった。

現在では、プログラム開発の支援環境も相当充実しており、入出力ファイル名、実行完了後の扱いなど端末主導で行うことができ、バッチ処理の準備作業に要する時間がかなり少なくて済むようになっている。

図表Ⅱ-6 バッチジョブの入力画面



出典： 第二種共通テキスト情報処理システム 中央情報教育研究所

(注) ここでの対話型バッチ処理は、従来の汎用コンピュータのTSS端末で行う処理方式を意味しているが、UNIXマシンやパソコンにおいてもバッチ処理という言葉が使われており、前者の拡張された機能と捉えてよい。

2. バッチ処理のオペレーティングシステム

(1) バッチ処理機能の実現

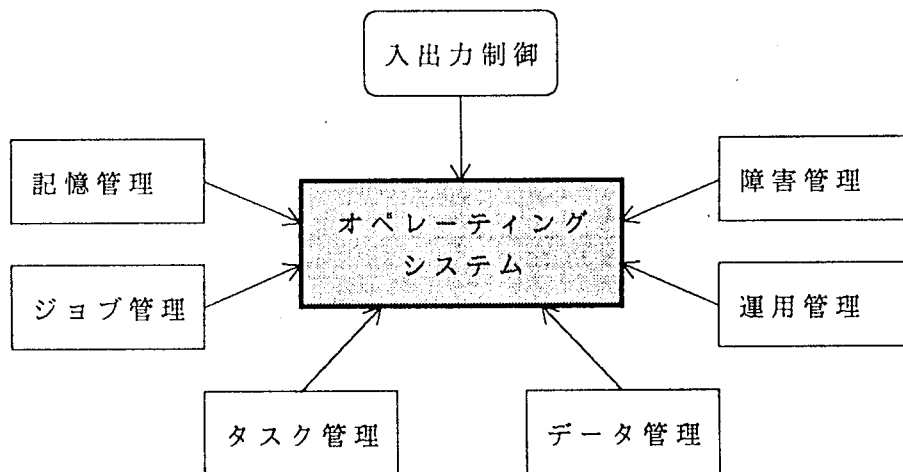
① オペレーティングシステムの基本構成

オペレーティングシステムは、次の役割を果たすプログラムの集合体である。

- ・ コンピュータの各種資源 [CPU、主記憶装置、補助記憶装置、データ、データベースなど] の効率的な確保・解放・稼動を可能とすること。
- ・ ジョブの実行順序の制御、多重処理の実現
- ・ 障害発生時の被害最小化機能

従来はOSの基本機能は、図表Ⅱ-7にあるように七つの要素に分けられ、勿論現在でもそれが基本であることには違いはない。しかしながら、単一コンピュータから分散型コンピュータへ、さらにネットワークコンピューティングへとシステムが複雑化しており、コンピュータ資源や制御すべき対象がどんどん増えている。

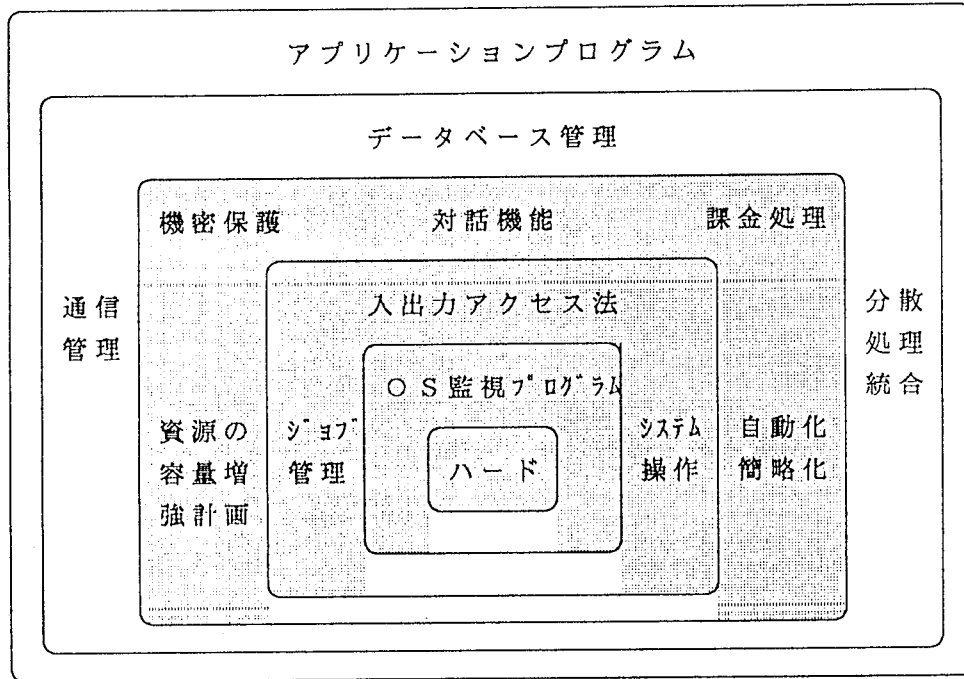
図表Ⅱ-7 OSの基本的な構成



出典：第二種共通テキスト情報処理システム 中央情報教育研究所

これからのオペレーティングシステムの対応範囲から考えると、コンピュータ内部での管理・制御だけでなく、むしろ自動化だとか、より人間に近い部分の制御、ネットワーキングにおける管理・制御あるいはセキュリティなどがクローズアップされてきている。これらは、従来の基本OS機能の範囲からは逸脱するという意見もあるだろうが、いまや狭義のOSでなく、広義のOSについて語らなくてはならないであろう。

図表Ⅱ-8 今後のOSの位置づけ



a. 制御プログラム部分

ア. 監視プログラム

コンピュータ資源を監視するとともに、プログラムの動作を監視制御する。

全てのコンピュータ資源の有効利用を実現し、高性能、高信頼の要求に応える。

・資源管理

CPU割当て、仮想空間と主メモリの管理、入出力管理、システム全体の効率を考慮した資源の配分管理

・プログラム管理

プログラムファイルから仮想空間への割当て・解除
プログラム間資源利用の排他制御、同期制御

・信頼性の確保

ハード、ソフト異常事態への対応能力

イ. 入出力アクセス法

ファイル、通信回線へのアクセスを管理する。

利用者には論理的なデータの入出力機能を提供し、内部的に物理的なデータの入出力に変換する。

SAM, PAM, DAM, VSAM, TAMなど。

ウ. システム操作機能

OSとオペレータとの交信を管理。オペレータは、システムの全体を監視し、動作状況に応じたシステムの変更やオペレータに求められる判断を行う。

システムとの交信には、システムやジョブに関する起動・停止、状態の問い合わせ、強制的な操作、トラブル通知に対する対応などを行う。

エ. ジョブ管理

ジョブの開始、実行、終了に関連する管理を行う。

ジョブキューからジョブを入力し、JCLを解析して使用する資源の使い方に誤りがないかをチェックする。

誤りがなければ、ジョブステップ毎に資源の割当て、解放を行うとともに、資源の利用履歴を保存する。

この情報は課金処理に使われる。終了後に処理結果がアウトプットキューに出力される。

b. 機密保護

端末を介した遠隔地からの不特定のユーザのコンピュータ利用は、必然的に機密データへのアクセスの脅威が発生する。企業のコンピュータセンターにとっては顧客の重要なデータ、社内の重要な機密データ、個人情報、また行政のコンピュータセンターにとっては住民データが保存されており、データの機密保護は社会レベルの課題である。

オペレータの人為的なミス、故意の操作、権利のない者のログイン、ハッカーのいたずら、コンピュータウイルスの蔓延など脅威はますます大きくなってきている。これらの不正利用に対して、コンピュータ側では

ア. 利用者の識別と正当性の確認（端末から高度な不正使用が可能）

パスワードやIDカードが代表例

イ. コンピュータ資源に対する不正アクセスの防止とアクセス履歴の保存

ファイルの命名上の規約、資源アクセスへの権利レベルの設定、データ情報の暗号化

ウ. 消去したデータに関する内容の消滅

エ. データアクセスに関する監査レポートの発行

などにより、機密保護を行っている。

c. 対話機能

メインフレームでは、バッチ処理の非効率性を解消するのにTSSの実現が極めて重要な出来事であった。これにより計算機との直接対話が可能となり、次第に対話を効率化させる方向で、ライン端末からスクリーン端末へ、プログラム開発支援機能の高度化、伝票入出力処理などが能率的に行えるようになっていった。

d. 課金

コンピュータの使用料金に関連する要素として、CPU、主メモリ、二次記憶装置、外部デバイス、入出力量、ターンアラウンドのプライオリティ、有料ソフトウェアの使用料が考えられる。

e. 自動化

コンピュータの生産性の向上、オペレータ介在時間の短縮のために、自動化、省力化が図られている。

例えば、磁気テープのオートマウント、出力結果に関するユーザに最も近い印刷場所の自動選択、ディスクの階層性（アクセス速度、データ量に応じた格納先の選定）などが行われる。

f. 資源容量計画

システムのチューニングやキャパシティの管理は大型のコンピュータには重要な作業の一つである。システムチューニングはシステムの動作状況をモニタリングしてシステム効率上のネックを見つけ、バランスのとれたシステムに構成を変更することである。

キャパシティ（資源容量管理）では、資源の使用状況、ジョブ、トランザクションの量、応答性などのデータをとり、現在および将来の業務量を想定してシステムにどのくらいの強化が必要であるかを予測する。

状況データの収集には、ハードウェアモニタやソフトウェアモニタを用いる。これにより、システムオーバヘッドやボトルネックを探す。

② スプーリング

バッチ処理で重要視される尺度にスループット（throughput）という概念があり、これは性能を表すもので単位時間当たりのジョブの処理件数に相当する。また、利用者がジョブを計算センターの受付に渡してからジョブの結果を受け取るまでの時間をターンアラウンドタイム（turn around time）という。これもまた性能の尺度の一つとなっている。

一般に、バッチジョブに関しては、資源の使用量にバラツキが多く、できるだけ多数の利

ユーザーが平均的に満足のいくようにしかけを行っている。

- ・ ターンアラウンドタイムの保証
- ・ スループット向上維持

このため、投入されたジョブをコンピュータの使用量に応じて分類している。これには、ジョブクラスという概念を導入して、ジョブクラス毎にジョブの待ち行列を作っている。ジョブクラスは、CPUの使用量、主メモリの使用量、特殊な装置の使用状況によって分類される。これにより、ターンアラウンドの保証、スループットの維持が行われシステムが安定的に運用されることになる。

スループットの向上の障害になりうるのは、ジョブを読むためのカード入出力装置、ジョブの結果を出力するための印刷装置のような比較的低速な機器からの制約である。これを少なくするために、カードリーダーからの入力ストリームが、高速なディスクに束ねて格納され、OSからのジョブの入力は実態的にはディスクから行うことにより高速化される。また、プリント出力結果についてもOSは直接プリンタに出すのではなく、ディスクにはきだしをしておいて、別途ディスクからプリンタへ出力されることにより全体の効率上がる。

このようにOSは低速の入出力機器からジョブを読んだり結果を出力したりするのではなく、一次記憶装置を使って、オフライン的にカードリーダーから高速のディスク装置へ、また同様にディスク装置からプリンタへ出すことによりシステム性能を上げることができる。

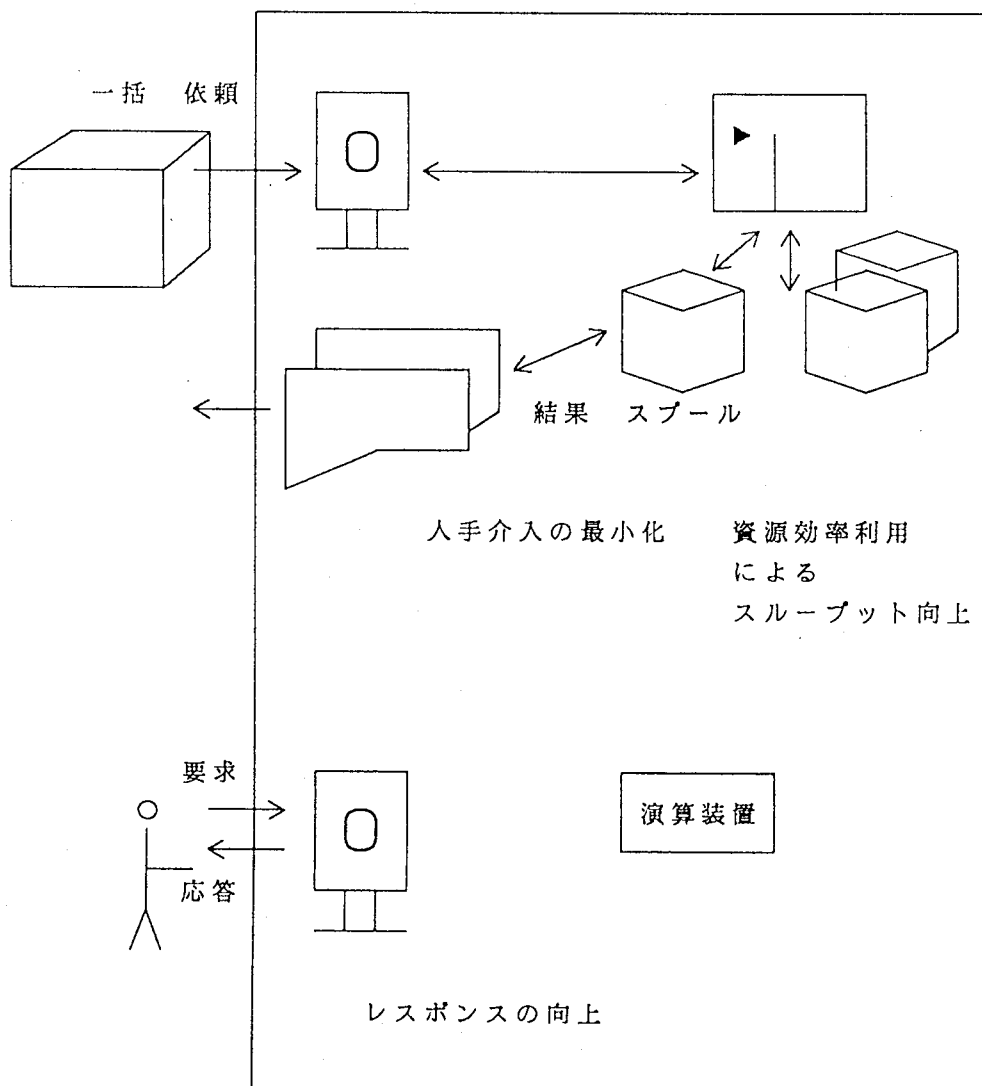
このメカニズムをスプーリング (SPOOL: simultaneous peripheral operations online) という。このスプーリングの存在については利用者のプログラムは知らなくてよい。プログラムはカードリーダーからデータを読んだり、プリンタへ出力していることをイメージしながら入出力をしていればよく、スプールの存在は意識にはない。カードリーダーからの読み込み要求があるとOSの内部でそれがディスクスプーラからの読み込みに変えられ、また、プリンタへの出力要求があれば、やはりOSの内部でディスク装置にはきだすように取り扱われている。

③ 他のオペレーティングシステムとの比較

1960年代にはバッチ処理の中で即時性の高い処理要求が出始め、オンラインリアルタイム処理のオペレーティングシステムが開発された。これをオンライントランザクション処理と呼んでいる。いまここで、バッチ、対話型、オンライントランザクションの処理の特徴をまとめておこう。

	バ ッ チ 処 理	対 話 型 処 理	トランザクション処理
システム資源の効率	資源の利用率高い	高 く な い	高 く な い
ス ル ー プ ッ ト	高 い	高 く な い	高 く な い
レ ス ポ ン ス	長 い	短 い	短 い

図表Ⅱ-9 バッチ処理との比較



(2) ジョブ制御

① ジョブ制御言語

ジョブの実行に関連する情報をオペレーティングシステムに指示するための言語

- ア. ジョブステップやデータの区切りを指示する。
- イ. ハードウェア資源の割当てを指示する。
- ウ. 実行するプログラムやライブラリを指示する。

② ジョブの構成

1個以上の順序づけをされたジョブステップ(実行するプログラム名で)を指示する形式をとる。各ジョブステップ毎に実行するプログラム名の指示、利用したいライブラリ名の指示、ファイルの確保・削除の指示、データ入力の指示などを行う。

///JOBNAME	翻訳、編集、実行するジョブ
///STEP1 EXEC PGM =	コンパイルを指示
///STEP2 EXEC PGM =	リンクエディットを指示

//

//SYSIN DD

データの入力装置を指示

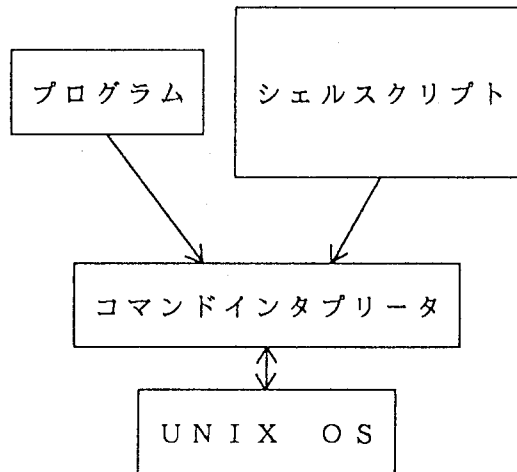
//SYSPRINT DD

出力先を指示

③ ワークステーションとパソコン

UNIX ワークステーションやMS-DOSなどのパソコンにおいてもバッチ処理機能がある。

図表II-10 UNIXのシェルスクリプト



3. 主要用語

オープンバッチ処理

クローズドバッチ処理

カフェテリア方式

センターバッチ処理

リモートバッチ処理

スプール

スルーブット

ターンアラウンドタイム

JCL