(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4279452号
(P4279452)
(45)発行日　平成21年6月17日(2009.6.17)　(24)登録日　平成21年3月19日(2009.3.19)

(51)Int.Cl.　F I
G06F 12/00 (2006.01)　G06F 12/00 501A
G06F 12/00 520J

請求項の数 9　(全 25 頁)

(21)出願番号	特願2000-513227 (P2000-513227)
(86)(22)出願日	平成10年9月18日 (1998.9.18)
(65)公表番号	特表2001-517836 (P2001-517836A)
(43)公表日	平成13年10月9日 (2001.10.9)
(86)国際出願番号	PCT/US1998/019453
(87)国際公開番号	WO1999/015994
(87)国際公開日	平成11年4月1日 (1999.4.1)
審査請求日	平成17年9月14日 (2005.9.14)
(31)優先権主張番号	08/933,681
(32)優先日	平成9年9月19日 (1997.9.19)
(33)優先権主張国	米国(US)
(31)優先権主張番号	09/010,301
(32)優先日	平成10年1月21日 (1998.1.21)
(33)優先権主張国	米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100089705
弁理士　社本　一夫
(74)代理人　100071124
弁理士　今井　庄亮
(74)代理人　100076691
弁理士　増井　忠弐
(74)代理人　100075236
弁理士　栗田　忠彦
(74)代理人　100075270
弁理士　小林　泰

最終頁に続く

(54)【発明の名称】１つの記憶媒体の名称空間を別の記憶媒体の名称空間に移植する場合に既定のアクションを実行するシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
第１デバイスの名称空間の少なくとも一部分を、ファイル・システムを介してアクセスされる第２デバイスの名称空間に移植するための方法であって、前記第１デバイスの名称空間と前記第２デバイスの名称空間とには、ファイルまたはディレクトリのうちの一方が関係し、該ファイルまたはディレクトリの前記一方はマウント・ポイント属性を含み、前記方法が、
前記第２デバイスの名称空間内のある位置にあるファイルまたはディレクトリの前記一方にマウント・ポイントを設定するステップであって、前記位置は、前記第１デバイスの名称空間の少なくとも一部分を前記第２デバイスの名称空間に移植する位置であり、前記マウント・ポイントが、前記ファイルまたはディレクトリの前記一方に含まれる前記マウント・ポイント属性により定められ、該マウント・ポイント属性がタグと値とを含み、前記タグが、当該マウント・ポイントを処理するドライバであるマウント・ポイント・ドライバを特定する、ステップと、
Ｉ／Ｏ要求を、前記マウント・ポイントに関係する前記ファイル・システムによって開始するステップであって、前記Ｉ／Ｏ要求が、前記マウント・ポイントが関連した少なくとも１つの名称コンポーネントによって特徴付けられた経路名を含む、ステップと、
を含み、
前記Ｉ／Ｏ要求を処理する前記ファイル・システムが、前記経路名の解明において前記名称コンポーネントを検出したとき、該検出された名称コンポーネントに対応するファイ

ルまたはディレクトリの前記一方の前記マウント・ポイント属性に含まれる前記タグが特定する前記マウント・ポイント・ドライバが、前記Ｉ／Ｏ要求の完了前に、前記Ｉ／Ｏ要求を処理する役割を引き受け、

前記マウント・ポイント・ドライバが、前記検出された名称コンポーネントに対応するファイルまたはディレクトリの前記一方の前記マウント・ポイント属性に含まれる前記値に基づいて、前記第１デバイスの名称空間において前記経路名により指定された情報にアクセスするためのアクションを開始すること、

を特徴とする方法。

【請求項２】

請求項１記載の方法において、前記アクションが、前記Ｉ／Ｏ要求とは異なる、方法。

【請求項３】

請求項１記載の方法において、ユーザへの表示のため情報をユーザ・インターフェースに送るステップを含む、方法。

【請求項４】

請求項１記載の方法において、前記第１デバイスおよび前記第２デバイスの統合名称空間をユーザに表示するステップを含む、方法。

【請求項５】

請求項１記載の方法において、前記Ｉ／Ｏ要求を完了するステップを含む、方法。

【請求項６】

請求項１記載の方法において、前記Ｉ／Ｏ要求は、前記第１デバイス上の情報にアクセスする、方法。

【請求項７】

請求項１記載の方法において、前記第１デバイス上でアクセスした情報をキャッシュして、前記第１デバイスにアクセスせずに、前記キャッシュした情報を検索可能とするステップを含む、方法。

【請求項８】

請求項１記載の方法において、前記キャッシュした情報を格納して、前記第２デバイスから前記キャッシュした情報を検索可能とする、方法。

【請求項９】

請求項１ないし８のいずれか１項記載の方法をコンピュータに実行させるコンピュータ実行可能命令を含むコンピュータ読み取り可能記録媒体。

【発明の詳細な説明】

【０００１】

（発明の背景）

（発明の分野）

本発明は、１つの記憶媒体の名称空間（namespace）を別の記憶媒体の名称空間に移植する（graft）システムおよび方法に関する。更に、特定すれば、本発明は、１つの記憶媒体の名称空間を別の記憶媒体の名称空間に移植する際に、ユーザまたはプログラムの介入なしに、アクションの実行を可能とする。

（従来の技術的現状）

機能性コンピュータ・システムは、通常、３つの基本的な構成要素から成る。第１の構成要素はコンピュータ・ハードウエアであり、第２の構成要素はユーザまたはアプリケーション・プログラムであり、第３の構成要素はオペレーティング・システムである。一般に、コンピュータ・ハードウエアは、中央演算装置（ＣＰＵ）のようなデバイス、ＲＡＭおよびＲＯＭのようなシステム・メモリ、磁気または光ディスク・ストレージのような大容量記憶装置、キーボードまたはその他の入力デバイス、ならびにディスプレイまたはその他の出力デバイスを含む。コンピュータ・システムのユーザは、通常、ユーザ・プログラムまたはアプリケーション・プログラムと双方向処理を行う。このようなプログラムは、周知のワード・プロセッシング・アプリケーション、スプレッドシート・アプリケーション、データベース・アプリケーション等を含む。コンピュータのオペレーティング・シス

テムは、ユーザにアプリケーション・プログラムの実行を開始させるとうような、多くの機能を実行する。加えて、最近のオペレーティング・システムは、アプリケーション・プログラムとコンピュータ・システムのハードウエアとの間のインターフェースも提供する。したがって、一旦はアプリケーション・プログラムが直接コンピュータ・システムのハードウエアにアクセスすることが一般的にはなったが、最近のオペレーティング・システムは、標準化され一貫性のあるインターフェースを提供し、ユーザ・アプリケーションが、標準化にしたがってコンピュータのハードウエアとインターフェースしたり、あるいはこれにアクセスすることが可能となっている。

【０００２】

ユーザ・アプリケーションのようなプロセスと特定の種類のハードウエア・デバイスとの間に一貫性のあるインターフェースを備えるためには、実際のハードウエアとプロセスとの間に、いくつかのソフトウエア・レイヤがあるとよい。例えば、あるプロセスはオペレーティング・システムにコールすることができる。一方、オペレーティング・システムは、ハードウエア・ドライバ・レイヤが提供するサービスを利用することができる。そして、ハードウエア・ドライバ・レイヤは、直接ハードウエアとインターフェースする。このような階層的手法の大きな利点は、残りのレイヤに大きなインパクトを及ぼすことなく、レイヤの追加や置換が可能なことにある。例えば、システムに新たなハードウエア・デバイスを追加する場合、新たなデバイスを追加すれば、オペレーティング・システムはこのハードウエアにアクセスすることができる。これらの変更は全て、既存のアプリケーション・プロセスに対するインパクトを最小に抑えて、またはインパクトを生ずることなく行うことができる。

【０００３】

多くのオペレーティング・システムは、種々の大容量記憶装置とインターフェースするＩ／Ｏサブシステムを備えている。このようなＩ／Ｏサブシステムは、Ｉ／Ｏ要求を、当該Ｉ／Ｏ要求を満たすために実行しなければならないアクションに変換するために必要な多くの詳細を扱う。これらのアクションの多くは、ファイルからの情報の検索というような上位概念から、磁気ディスク上の特定の場所への読み取りヘッドの位置付けや、当該特定の場所からの情報の検索というような下位アクションへの変換要求のために、必要となるものである。

【０００４】

最近のコンピュータ・システムでは、通常データは階層的な大容量記憶装置上に格納し、ツリー構造に似せた階層にディレクトリおよびファイルを編成している。いずれのファイルまたはディレクトリの場所も通常、経路名（path name）で指定する。経路名は、典型的に、ルート即ち開始ディレクトリから始まり、所望のファイルまたはディレクトリに達するまでの、後続の各サブディレクトリの名称を示す。例えば、“file.dat”と呼んでいるファイルは、ディレクトリ“temp”に格納することができ、これは、ルート・ディレクトリ“root＼.”のサブディレクトリである。file.datにアクセスするには、Ｉ／Ｏ要求は、適切なファイルを特定しアクセスするために、Ｉ／Ｏサブシステムを特定の場所に導く経路名を明示的または暗示的に含めばよい。file.datの経路名は、したがって、＼root＼temp＼file.datとなる。

【０００５】

Ｉ／Ｏサブシステムが、関連する経路名を有するＩ／Ｏ要求を受け取った場合、経路名を解明し、特定の記憶装置上の特定の場所への変換を可能とし、Ｉ／Ｏ要求を満たせるようにしなければならない。経路名は、直接Ｉ／Ｏサブシステムに供給することができ、あるいは既知の情報に基づいて作成してもよい。あるファイル・システム内におけるファイル名の解明は、典型的に、多段手順である。通常、ファイル・システムが特定に成功するために必要な、名称付きコンポーネントの全てをデコードする段階から開始する。次に、この手順は、経路名において連続する名称コンポーネントを、通常左から右に識別する繰り返しプロセスに進む。この手順は、これら名称解明の各々の成功または失敗で終了する。このように、上述の例では、経路名を連続するコンポーネントに分解し、解明プロセスに

よって、順番に、ルート・ディレクトリ、tempサブディレクトリ、そしてファイルfile.datを特定していく。

【０００６】

安価で強力なコンピュータの普及により、多くのユーザおよび組織は、ローカル・エリア・ネットワーク（ＬＡＮ）、ワイド・エリア・ネットワーク（ＷＡＮ）、またはその他のネットワーク構築技術を利用して、相互接続したコンピュータのネットワークを組み立て、個々のユーザにデータおよび情報を共有させている。多くの場合、これらのコンピュータ・ネットワークは、多数の記憶装置を有し、多数のユーザによってアクセス可能となっている。その結果、個々のユーザは多くの異なる記憶装置へのアクセスを有する場合もある。記憶装置の数、および各記憶装置上のディレクトリおよびサブディレクトリの階層が増大するに連れて、ユーザは彼らのデータを見つけ出し管理するのが一層困難となる。

【０００７】

ボリュームとは、ファイル・システムによってフォーマットし、ファイル・システム経路またはＩ／Ｏサブシステム内部のデバイス名によってアクセス可能な記憶単位である。ユーザに多数のボリュームが利用可能であり、その各々がディレクトリおよびサブディレクトリの階層を有する場合、ユーザは個々のボリュームのディレクトリおよびサブディレクトリ間の相互関係を概念的に把握することが困難となるであろう。これは、当然特定の階層構造にしたがって編成するような大量のデータは、単一の記憶ボリューム上には全ての関連データの格納および編成ができなくなる程大きくなるという特異な問題を招く。このような状況では、通常、一部のデータを１つのボリュームに格納し、別のデータを他のボリュームに格納することが必要となる。このように論理的に関連するデータを多数のボリュームに分割すると、一部のユーザには、一層の困惑および困難が感じられるであろう。

【０００８】

このような欠点（limitation）を多少とも克服するために、記憶装置上のデータの物理的な編成とは独立した、記憶装置の論理図を提示する機構が開発されている。これらの機構は、１つの記憶装置の名称空間（namespace）を他の記憶装置の名称空間に移植し、単体の統一した論理図をユーザに提示しようというものである。

【０００９】

ここで図１を参照し、このような移植プロセスの一例を示す。図１には、２つのボリューム２０および２２を示す。ボリューム２０および２２は、各々、独立したディレクトリ構造を有する。ボリューム２０のディレクトリ構造を全体的に２４で示し、ボリューム２２のディレクトリ構造を全体的に２６で示す。ここで用いる場合、ディレクトリ構造および名称空間は、相互交換可能に用い、特定のボリューム上にあるディレクトリおよびファイルの名称または構造を特定するものとする。ボリューム２０およびボリューム２２は各々個別のディレクトリ構造を有するが、この例では、論理ディレクトリ構造２８で示すように、単一の統合化したディレクトリ構造をユーザに提示することが望ましい。論理ディレクトリ構造２８は、ディレクトリ構造２６をDir4 ３０においてディレクトリ構造２４に移植することが望ましいことを示す。このような移植を行うと、論理ディレクトリ構造２８が得られる。

【００１０】

１つの記憶ボリュームの名称空間を別の記憶ボリュームの名称空間に移植するには、多くの機構が使用可能であるが、「マウント・ポイント」（mount point）と呼ばれる、１つの共通した技術がある。マウント・ポイントとは、別の名称空間またはディレクトリ構造をマウントまたは移植する名称空間位置のことである。したがって、図１では、Dir4 ３０をマウント・ポイントと定義する。図１では、これは破線の矢印３２で示す。概念的には、マウント・ポイント３０を横断するときに、ファイル・システムはボリューム２０の代わりにボリューム２２にアクセスする。例えば、ユーザに論理ディレクトリ構造２８を提示する場合、ユーザは、Dir8におけるfile.datにアクセスすることを望み、そのファイルへの経路は、C:＼Dir2＼Dir4＼Dir8＼file.datとして与える。ファイル・システムがこの経路名を解明する場合、ファイル・システムは、ボリューム２０にアクセスすることに

よって、Dir4に対する経路名を解明する。一旦Dir4にアクセスしたなら、しかしながら、ファイル・システムはDir4をマウント・ポイントとして認識し、ボリューム２２にアクセスすることによって、解明プロセスを続ける。本質的に、マウント・ポイントは従来より一種の特殊ディレクトリとして作用し、他のボリュームの名称空間内のある位置にアクセスを方向転換させることができる。

【００１１】

マウント・ポイントは、基盤の物理的記憶構造とは無関係の論理編成を作成することができるが、いくつかの問題も存在する場合がある。例えば、図１のボリューム２２がリムーバブル記憶装置である場合、論理ディレクトリ構造２８を介してこのデバイスにアクセス可能となる前に、リムーバブル媒体を検索しマウントしなければならない。典型的に、これらの工程は、ユーザに論理ディレクトリ構造２８を提示する前に行われる。このような検索およびマウント・プロセスは、例えば、ユーザによってオペレーティング・システムを介して、またはアクセスする前に当該デバイスを検索しマウントしなければならないことを知っているプログラムによって起動することが考えられる。ユーザまたはアプリケーション・プロセスにリムーバブル・ボリュームを検索しマウントする負担を強いると、いくつかの望ましくない結果が生ずる。例えば、どのようにして媒体を検索しマウントするのかわかるように、ユーザを訓練しなければならない。アプリケーション・プログラムに負担を強いると、リムーバブル媒体を検索しマウントしようとするアプリケーション・プログラム毎に、このような機能性を組み込まなければならない。この結果、大量の冗長性および手間の繰り返しを生じさせることになる。

【００１２】

したがって、ユーザに論理ディレクトリ構造を提示し、ユーザまたはアプリケーション・プログラムがリムーバブル記憶媒体にアクセスしようとする前に実行しなければならない、リムーバブル記憶媒体のマウントおよび検索のようなコマンドを不要とすることができるシステムを有することができれば望ましいであろう。また、これらの機能を、手間の繰り返しを生じさせずに行えれば望ましいであろう。更に、ユーザがどのようにリムーバブル媒体を検索しマウントするか知る必要なく、この結果を得ることが望ましい。

（発明の概要）

従来の技術的現状における前述の問題は、本発明が克服することに成功した。本発明は、１つの記憶媒体の名称空間の全部または一部を同じまたは別の記憶媒体の名称空間に移植する際に、任意のアクションを実行するシステムおよび方法に関するものである。本発明は、広範囲のシステムを実現し、広範囲の結果を得るために用いることができる。例えば、マウント・ポイントを横断するとき、オペレーティング・システムは、Ｉ／Ｏ要求を完了する前に、自動的に適切な媒体を検索し装填することも可能である。本発明は非常にロバスト性が高いので、マウント・ポイントを横断するときに、あらゆる任意のアクションでも実行可能であり、通常ではＩ／Ｏシステムやマウント・ポイントを横断するときに、Ｉ／Ｏ要求には関連のないアクションでも実行可能である。

【００１３】

本発明は、アクティブ・マウント・ポイント・ドライバ（active mount point driver）を提供する。このドライバは、オペレーティング・システムのＩ／Ｏサブシステムの一部としてもよい。経路またはファイル名を含むＩ／Ｏ要求を受け取った場合、従来の方法でこの経路およびファイル経路を解明する。解明プロセスの間にアクティブ・マウント・ポイントに出くわした場合、制御をアクティブ・マウント・ポイント・ドライバに引き渡す。すると、アクティブ・マウント・ポイント・ドライバは、Ｉ／Ｏ要求の完了を進めるために必要なあらゆるアクションを実行することができる。一旦アクティブ・マウント・ポイント・ドライバが、Ｉ／Ｏ要求が完了する前に行う必要がある任意の所望のアクションを完了または開始したなら、Ｉ／Ｏサブシステムの別のコンポーネントに制御を再度引き渡し、アクティブ・マウント・ポイント・ドライバのアクションによってＩ／Ｏ要求が未だ完了していない場合、このＩ／Ｏ要求を完了させることができる。

【００１４】

アクティブ・マウント・ポイント・ドライバは、そのアクションを実行する際、広範囲におよぶシステム資源、プロセス、ドライバ等を利用することができる。一旦制御をアクティブ・マウント・ポイント・ドライバに引き渡したなら、アクティブ・マウント・ポイント・ドライバは、システムの資源、プロセス、ドライバ、サブシステム等の全てにアクセスすることができる。システムが他のシステムとネットワークを組んでいる即ち接続している場合、アクティブ・マウント・ポイント・ドライバは、これらのシステム、ならびに当該システムに関連するあらゆるコンポーネント、サブシステム、プロセス、ドライバ等にアクセスすることができる。本発明が定義する構造は、ロバスト性が高いので、アクティブ・マウント・ポイント・ドライバは、アクティブ・マウント・ポイントに出くわしたときには、望ましいアクションであればいずれでも実行することができる。

【００１５】

アクティブ・マウント・ポイントの機能（capability）の一例として、全てのＣＤをＣＤジュークボックスに統合するディレクトリ階層を作成し、ユーザに表示することができる。特定のＣＤにアクセスしたいユーザがいれば、単にディレクトリ構造内の適切な場所を選択するだけでよい。Ｉ／ＯシステムがＣＤへの経路名を解明し始めると、１つ以上のアクティブ・マウント・ポイントに出くわす可能性がある。これらのアクティブ・マウント・ポイントは、とりわけ、制御をアクティブ・マウント・ポイント・ドライバに引き渡すことができ、次いでアクティブ・マウント・ポイント・ドライバは、ＣＤジュークボックスにコマンドを発行し、適切なＣＤを選択し装填させることができる。次いで、Ｉ／Ｏ要求を完了するために、制御をＩ／Ｏサブシステムに引き渡すことができる。ＣＤが音楽を収容している場合、アクティブ・マウント・ポイント・ドライバは、一旦ＣＤがマウントされたならば、マルチメディア・サブシステムにコマンドを発行し、当該ＣＤを演奏し始めることも可能である。アクティブ・マウント・ポイントに出くわした場合には、その他のあらゆるアクションも利用可能である。

【００１６】

今述べたシステムを実現するためには、多くの機構が使用可能である。重要な特徴の中には、アクティブ・マウント・ポイントを識別可能なこと、およびアクティブ・マウント・ポイントに出くわしたときに制御をアクティブ・マウント・ポイント・ドライバに引き渡すことができることである。この移行を行う機構は、効率的であり、しかも一旦アクティブ・マウント・ポイント・ドライバが実行すべきアクションを完了したならば、必要に応じて、アクティブ・マウント・ポイント・ドライバから制御を戻せるようでなければならない。加えて、制御をアクティブ・マウント・ポイント・ドライバに引き渡す場合、アクティブ・マウント・ポイント・ドライバは、実行すべきアクションを特定即ち選択できなければならない。以下で更に詳しく説明するが、どのアクションを完了すべきか決定即ち選択した場合、アクティブ・マウント・ポイント・ドライバは、他の資源から他の情報を収集することができ、あるいは制御を他のドライバまたはプロセスに引き渡してこれらの決定を行うことができる。本発明の一実施形態は、マウント・ポイントと共に格納してある追加情報を利用して、これらの機能の一部を遂行する。

【００１７】

多くのシステムでは、ディレクトリまたはファイルは、異なるプロパティ（property）または属性の集合体と見なすことができる。ファイルの共通な属性には、名称属性や、システム、読み取り専用、隠れ（hidden）等のような種々のフラグ属性、およびデータを格納するためのデータ属性がある。ディレクトリの共通属性には、ディレクトリの名称、可能性として何らかのセキュリティ情報、当該ディレクタが収容するディレクトリまたはファイルを識別するポインタまたはその他の機構等がある。本発明の一実施形態では、追加の属性をディレクトリまたはファイルに付加することによって、アクティブ・マウント・ポイントを作成する。Ｉ／Ｏ要求中にディレクトリまたはファイルに出くわした場合、Ｉ／Ｏサブシステムはこの追加の属性を識別し、ディレクトリまたはファイルをアクティブ・マウント・ポイントとして認識することができる。次いで、制御をアクティブ・マウント・ポイント・ドライバに渡し、次の処理に進むことができる。アクティブ・マウント・ポ

イント・ドライバは、種々のソースから情報を検索することができ、その中には、ディレクトリまたはファイルの１つ以上の属性が含まれる。この情報に基づいて、アクティブ・マウント・ポイント・ドライバは、実行すべきアクションを決定することができる。一旦アクティブ・マウント・ポイントがその作業を完了したなら、次いで制御をＩ／Ｏサブシステムに戻し、次の処理に進むことができる。

【００１８】

アクティブ・マウント・ポイント属性は、ファイルおよびディレクトリ双方に追加することができる。属性は、好ましくは、性質上添加的とし、個々のファイルおよびディレクトリが、アクティブ・マウント・ポイント属性のステータスに応じて、アクティブ・マウント・ポイントであるか、またはそうでないというようにするとよい。好ましくは、アクティブ・マウント・ポイント属性は、タグおよびデータ値双方を有する。タグは、アクティブ・マウント・ポイントの「オーナー」であるアクティブ・マウント・ポイント・ドライバを識別する際に用いる。通常、アクティブ・マウント・ポイントのオーナーは、当該アクティブ・マウント・ポイントに関与するＩ／Ｏ要求の全部または一部を処理する役割を担う。この構造によって、単一のシステム内に多数のアクティブ・マウント・ポイント・ドライバを位置付け、各々が異なる目的を達成するように構成することが可能となる。アクティブ・マウント・ポイント属性のデータ値は、オーナーによってアクティブ・マウント・ポイント属性に格納する。したがって、オーナーは、アクティブ・マウント・ポイント属性の値を用いて、当該アクティブ・マウント・ポイントに関与する個々のＩ／Ｏ要求を完了する際に必要なまたは役に立つ、あらゆるデータを格納することができる。このようなデータ値は、どのようなアクションを実行すべきか判断する際に必要な全ての情報を含むことができ、あるいはどのようなアクションを実行すべきかについて判断する際に必要な情報を突き止めることをアクティブ・マウント・ポイント・ドライバに可能にするポインタまたはその他の情報を含むこともできる。また、アクティブ・マウント・ポイントの値は、どのようなアクションを実行すべきかについて判断する際に用いるべき、その他のソフトウエア・エンティティを識別することも可能である。要するに、アクティブ・マウント・ポイント属性の値は、オーナーが制御するので、オーナーはアクティブ・マウント・ポイントにあらゆる望ましい情報を格納することができる。

【００１９】

本発明の一実施形態は、複数の階層状ドライバを有するＩ／Ｏシステムを利用する。アクティブ・マウント・ポイント・ドライバは、Ｉ／Ｏシステムのレイヤの１つを形成する。特定のドライバがあるアクティブ・マウント・ポイント属性を識別すると、このドライバはアクティブ・マウント・ポイントのタグおよび値を抽出する。次いで、アクティブ・マウント・ポイントのタグおよび値と共に、Ｉ／Ｏ要求を階層状ドライバの別のものに受け渡し、あるドライバがそれ自体をアクティブ・マウント・ポイントのオーナーであることを確認するまで続ける。次いで、オーナーは制御を取得し、Ｉ／Ｏ要求の処理を再開する。アクティブ・マウント・ポイントのオーナーは、Ｉ／Ｏ要求を完全に処理することができ、あるいは他のドライバ、資源、情報、プロセス等を利用してＩ／Ｏ要求を完全に処理することもできる。状況によっては、オーナーは他のコンピュータまたはシステムも用いて、Ｉ／Ｏ要求を完全に処理することも可能である。

【００２０】

各アクティブ・マウント・ポイントはタグおよび値双方を有するので、アクティブ・マウント・ポイント機構は、非常に柔軟な構造を備えており、あらゆる数のドライバがあらゆる数の機能を遂行するためにも使用可能である。自動的にリムーバブル媒体を装填しマウントするシステムの一例について既に示した。別の例として、アクティブ・マウント・ポイントは、通常ではＩ／Ｏシステムとは関連のないアクションを実行することも可能である。例えば、アクティブ・マウント・ポイントは、安全な物理的設備に格納してあるファイルの名称空間を、大きなディレクトリ階層に移植するセキュア・ファイル・システム（secure file system）を実現する際にも使用可能である。個人がセキュア・ファイルの１つにアクセスする場合、制御をアクティブ・マウント・ポイント・ドライバに引き渡す。

すると、アクティブ・マウント・ポイント・ドライバは、特殊な妥当性判断手順を起動し、ファイルにアクセスしようとしている個人が、それを行う許可を有するか否かについて判定を行う。例えば、アクティブ・マウント・ポイント・ドライバは、ファイルにアクセスしようとしている個人に対して、別個の挑戦を開始することができる。また、アクティブ・マウント・ポイント・ドライバは、Ｅ－メールまたはその他のメッセージをセキュリティ部署に送り、ある個人がファイルにアクセスしようとしていたことを通知する。ファイルがある時間中またはある日にだけアクセスが許可される場合、アクティブ・マウント・ポイント・ドライバは、日時をチェックし、そのファイルに対するアクセスを許可すべきか否かについて判定を行うことができる。この例でわかるように、アクティブ・マウント・ポイント・ドライバは、直接入手可能な情報に基づいて、あるいはアクセスまたは入手した情報のいずれかに基づいて、あらゆる数のアクションでも実行することができる。

【００２１】

最後の例として、本発明を利用すると、従来では記憶装置の名称空間内には移植できなかったものを、記憶装置の名称空間に移植することが可能となる。例えば、マウント・ポイントは、１つの大容量記憶装置のファイル・システム名称空間を、他の大容量記憶装置のファイル・システム名称空間に移植するために従来用いられてきた。しかしながら、本発明では、インターネットまたはイントラネットのウェブ・サイトのような、ファイル・システム名称空間でないものでも、大容量記憶装置の名称空間に移植することが可能となる。インターネットは、ユーザ・インターフェースでは、アクティブ・マウント・ポイントからユーザ・インターフェースに情報を供給することによって、特殊なアイコンで表わすことも可能である。ユーザがそのアイコンを開こうとした場合、アクティブ・マウント・ポイント・ドライバは、インターネットまたはその他のデータ・プロバイダへのアクセスを行い、ユーザ・インターフェースを介してデータ・プロバイダからユーザに入手可能なコンテンツに関するフィードバックを提供する。先の例に示したように、本発明は非常にロバスト性が高い機構を備えているので、１つのデバイスの名称空間を別のデバイスの名称空間に移植する際、あらゆる任意のアクションを実行することが可能である。

【００２２】

本発明の付加的な利点は、以下に続く説明に明記してあり、部分的にはその記載から自明であり、あるいは本発明の実施によって習得することができる。本発明の目的および利点は、添付した請求の範囲に特定して指摘した手段および組み合わせによって実現し、獲得することができる。本発明のこれらおよびその他の目的ならびに特徴は、以下の説明および添付した請求の範囲から一層明白となり、あるいは以下に明記する本発明の実施によって習得することができよう。

【００２３】

本発明の先に引用したおよびその他の利点ならびに目的を得る形態について、添付図面に示すその具体的な実施形態を参照しながら、先に簡単に説明した本発明の更に特定的な説明を行う。これらの図面は本発明の典型的な実施形態のみを図示するのであり、したがってその範囲を限定するとは見なさないことを理解の上で、添付図面の使用によって、本発明を更に具体的かつ詳細に記載し説明することとする。

（好適な実施形態の詳細な説明）

以下に、本発明のシステムおよび方法を実現するために用いる実施例の構造または処理のいずれかを例示するために、図面を用いて本発明について説明する。このように図面を用いて本発明を提示することは、その範囲の限定として解釈すべきことではない。本発明は、階層状データ格納のための方法およびシステム双方を念頭に入れている。本発明の実施形態は、種々のコンピュータ・ハードウエアを備えた特殊目的コンピュータまたは汎用コンピュータから成るものとすることができ、以下で更に詳細に説明する。

【００２４】

また、本発明の範囲に該当する実施形態は、実行可能な命令またはデータ・フィールドが記憶されているコンピュータ読み取り可能媒体も含む。このようなコンピュータ読み取り可能媒体は、汎用コンピュータまたは特殊目的コンピュータによってアクセス可能で、か

つ入手可能なあらゆる媒体とすることができる。一例として、限定ではなく、このようなコンピュータ読み取り可能媒体は、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、ＣＤ－ＲＯＭまたはその他の光ディスク・ストレージ、磁気ディスク・ストレージまたはその他の磁気記憶装置、あるいは所望の事項可能な命令またはデータ・フィールドを格納するために用いることができ、汎用コンピュータまたは特殊目的コンピュータによってアクセス可能なその他のあらゆる媒体を含むことができる。前述の組み合わせも、コンピュータ読み取り可能媒体の範囲に含まれて当然である。実行可能な命令とは、例えば、汎用コンピュータ、特殊目的コンピュータ、または特殊目的処理装置に、ある種の機能または機能群を実行させる、命令やデータを含む。

【００２５】

図２および以下の論述は、本発明を実現可能な、適当な計算機環境の端的で全体的な説明を行うことを意図したものである。本発明は、パーソナル・コンピュータが実行するプログラム・モジュールのような、コンピュータ実行可能命令に全体的に関連して説明するが、これが必要条件ということではない。一般的に、プログラム・モジュールは、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造等を含み、特定のタスクを実行したり、あるいは特定の抽象的なデータ型を実装する。さらに、本発明は、ハンド・ヘルド型デバイス、マルチ・プロセッサ・システム、マイクロプロセッサを用いたまたはプログラム可能な民生用電子機器、ネットワークＰＣ、ミニコンピュータ、メインフレーム・コンピュータ等を含む、その他のコンピュータ・システム構成とでも実施可能であることを、当業者は認めよう。また、本発明は、分散型計算機環境においても実施可能であり、その場合、タスクは、通信ネットワークを通じてリンクしてあるリモート処理デバイスによって実行する。分散型計算機環境では、プログラム・モジュールは、ローカルおよびリモート記憶装置双方に位置することができる。

【００２６】

図２を参照すると、本発明を実現するシステム例は、従来のコンピュータ３４の形態の汎用計算機を含み、演算装置３５、システム・メモリ３６、およびシステム・メモリ３６ないし演算装置３５を含む種々のシステム・コンポーネントを結合するシステム・バス３７を含む。システム・バス３７は、種々のバス・アーキテクチャのいずれかを用いたメモリ・バスまたはメモリ・コントローラ、周辺バス、およびローカル・バスを含む、数種類のバス構造のいずれでもよい。システム・メモリは、リード・オンリ・メモリ（ＲＯＭ）３８、およびランダム・アクセス・メモリ（ＲＡＭ）３９を含む。起動中等においてコンピュータ３４内のエレメント間の情報転送に供する基本ルーチンを収容する基本入出力システム（ＢＩＯＳ）４０は、ＲＯＭ３８に格納しておくことができる。また、コンピュータ３４は、図示しない磁気ハード・ディスクの読み取りおよび書き込みを行う磁気ハード・ディスク・ドライブ４１、リムーバブル磁気ディスク４３の読み取りおよび書き込みを行う磁気ディスク・ドライブ４２、ならびにＣＤ－ＲＯＭまたはその他の光媒体のようなリムーバブル磁気ディスク４５の読み取りおよび書き込みを行う光ディスク・ドライブ４４も含むことができる。磁気ハード・ディスク・ドライブ４１、磁気ディスク・ドライブ４２、および光ディスク・ドライブ４４は、ハード・ディスク・ドライブ・インターフェース４６、磁気ディスク・ドライブ・インターフェース４７、および光ディスク・ドライブ・インターフェース４８によって、それぞれシステム・バス３７に接続してある。これらのドライブおよびそれに関連するコンピュータ読み取り可能媒体は、コンピュータ読み取り可能命令、データ構造、プログラム・モジュール、およびコンピュータ３４用のその他のデータの不揮発性格納を可能にする。ここに記載する環境の一例は、磁気ハード・ディスク４１、リムーバブル磁気ディスク４３およびリムーバブル光ディスク４５を採用するが、磁気カセット、フラッシュ・メモリ・カード、ディジタル・ビデオ・ディスク、ベルヌーイ・カートリッジ、ランダム・アクセス・メモリ（ＲＡＭ）、リード・オンリ・メモリ（ＲＯＭ）のような、コンピュータによるアクセスが可能なデータを格納することができる、他の種類のコンピュータ読み取り可能媒体も、動作環境の一例において使用可能であることは、当業者には認められよう。

【００２７】
多数のプログラム・モジュールを、ハード・ディスク、磁気ディスク４３、光ディスク４５、ＲＯＭ３８またはＲＡＭ３９上に格納することができ、オペレーティング・システム４９、１つ以上のアプリケーション・プログラム５０、その他のプログラム・モジュール５１、およびプログラム・データ５２を含む。ユーザは、キーボード５３およびポインティング・デバイス５４のような入力デバイスによって、コマンドおよび情報をコンピュータ３４に入力することができる。他の入力デバイス（図示せず）として、マイクロフォン、ジョイスティック、ゲーム・パッド、衛星パラボラアンテナ（satellite dish）、スキャナ等を含むことができる。これらおよびその他の入力デバイスは、多くの場合、システム・バス３７に結合してあるシリアル・ポート・インターフェース５５を介して、演算装置３５に接続しているが、パラレル・ポート、ゲーム・ポートまたはユニバーサル・シリアル・バス（ＵＳＢ：universal serial bus）のようなその他のインターフェースによって接続することも可能である。モニタ５６または別の種類の表示装置も、ビデオ・アダプタ５７のようなインターフェースを介して、システム・バス３７に接続する。モニタに加えて、パーソナル・コンピュータは、典型的に、スピーカやプリンタのような、その他の周辺出力デバイス（図示せず）を含む。

【００２８】
コンピュータ３４は、リモート・コンピュータ５８のような１つ以上のリモート・コンピュータへの論理接続を用いて、ネットワーク環境で動作することも可能である。リモート・コンピュータ５８は、パーソナル・コンピュータ、サーバ、ルータ、ネットワークＰＣ、ピア・デバイス、またはその他の一般的なネットワーク・ノードとすることができ、典型的に、コンピュータ３４に関して先に記載したエレメントの多くまたは全てを含むが、メモリ記憶装置５９だけを図２に示す。図２に示す論理接続は、ローカル・エリア・ネットワーク（ＬＡＮ）６０およびワイド・エリア・ネットワーク（ＷＡＮ）６１を含み、ここでは、例示のためにこれらを提示するが、限定ではない。このようなネットワーク環境は、事務所の企業規模のコンピュータ・ネットワーク、イントラネットおよびインターネットでは一般的である。

【００２９】
ＬＡＮネットワーク環境で用いる場合、コンピュータ３４は、ネットワーク・インターフェースまたはアダプタ６２を介して、ローカル・ネットワーク６０に接続する。ＷＡＮネットワーク環境で用いる場合、コンピュータ３４は典型的にモデム６３またはインターネットのようなワイド・エリア・ネットワーク６１を通じて通信を確立するその他の手段を含む。モデム６３は、内蔵型でも外付け型でもよく、シリアル・ポート・インターフェース５３を介してシステム・バス３７に接続する。ネットワーク環境では、コンピュータ３４に関連して図示したプログラム・モジュールまたはその一部を、リモート・メモリ記憶装置５９に格納することも可能である。図示のネットワーク接続は一例であり、コンピュータ間に通信リンクを確立するその他の手段も使用可能であることは認められよう。

【００３０】
ここで図３を参照して、クライアント・プロセスと、Ｉ／Ｏ要求を処理するための複数のドライバ手段を用いるＩ／Ｏシステムを有するオペレーティング・システムとの間の双方向処理の簡略化した図を示す。この図は、例えば、Microsoft Windows NTオペレーティング・システムを表わす。図３の図は、Ｉ／Ｏ要求を処理する複数のドライバ手段を用いる、あらゆるオペレーティング・システムも代表することができる。Ｉ／ＯシステムにおいてＩ／Ｏ要求を処理するために階層状ドライバ手段を用いることにより、多くの利点が得られる。このようなアーキテクチャから得られる利点の１つは、特定の目的のために特注した追加のドライバ手段を挿入可能なことである。以下で更に詳しく提示するが、本発明の一実施形態は、このような階層状アーキテクチャを利用することもでき、Ｉ／Ｏシステム内のレイヤの１つとして、アクティブ・マウント・ポイント・ドライバ手段を含むことができる。以下の図３に関する論述は、このような実施形態に対して説明することを意図するものである。

【００３１】
図３において、クライアント・プロセス６６は、オペレーティング・システム・サービス６８を利用して、Ｉ／Ｏ要求を実行する。これは、典型的に、クライアント・プロセス６６が、オペレーティング・システムが提供するアプリケーション・プログラム・インターフェース（ＡＰＩ）機能にコールすることによって行われる。適切なＡＰＩ機能をコールすると、その結果として、オペレーティング・システム・サービス６８にコールすることになる。このようなコールを矢印７０で示す。
【００３２】
図３において、クライアント・プロセス６６は、「ユーザ」モードで動作するものとして示してあり、オペレーティング・システム表面は「カーネル」モードで動作するものとして示している。最近のオペレーティング・システムは、典型的に、種々のアプリケーション・プログラムおよび直感的ユーザ・インターフェースに対してロバストな環境を備えている。このようなオペレーティング・システムは、通常、当該オペレーティング・システムの精巧性レベル、およびオペレーティング・システムが実現するセキュリティ機能に応じて、異なる動作レベル即ち「モード」を有する。通常のアプリケーション・プログラムは、典型的に、最も低い優先度で走り、適所にセキュリティ・デバイスの完全な補体（complement）を有し、他のアプリケーションまたはオペレーティング・システムの他の層との干渉を禁止している。ハードウエアおよびオペレーティング・システムが提供するサービスには、インターフェースまたは機構を介してのみアクセスされ、これらの制御によって、ユーザ・モードにおけるユーザ・アプリケーションまたはその他のプロセスがシステムを「クラッシュ」する可能性を抑えるようにしている。この優先度が最も低いモードを、典型的に、ユーザ・モードと呼び、殆どのコンピュータ・ユーザになじみの深いモードである。ドライバとそれらの関連するハードウエアとの緊密な統合のため、そして多くのドライバが実行するタスクの時間重要性（time critical nature）のために、ドライバは典型的にオペレーティング・システム・モードで走る。これは、かなり高めの優先度およびかなり低めのセキュリティ保護を有する。このモードを通常「カーネル」モードと呼ぶ。ドライバおよびその他のオペレーティング・システム・サービスをカーネル・モードに置くことによって、オペレーティング・システムを実行する優先度を高め、ユーザ・モードでは不可能な多くの機能を実行することが可能となる。
【００３３】
クライアント・プロセス６６がオペレーティング・システム・サービス６８をコールしＩ／Ｏ要求を実行する場合、Ｉ／Ｏ要求を処理する第１ドライバ手段にＩ／Ｏ要求を渡す。図３では、ファイル・システム・ドライバ７２およびデバイス・ドライバ７４が、Ｉ／Ｏ要求を処理するドライバ手段の例を表わす。Ｉ／Ｏ要求を第１ドライバ手段に渡す動作は、図３では、例えば矢印７６で示している。次に、ファイル・システム・ドライバ７２は、Ｉ／Ｏ要求を取得し、全体的にＩ／Ｏ要求の部分的処理を実行し、その後このＩ／Ｏ要求を次のドライバに渡す。
【００３４】
一例として、クライアント・プロセス６６がディスク上の特定のファイルを開き、このファイルから情報を検索しようとしていると仮定する。Ｉ／Ｏ要求は、クライアント・プロセス６６からオペレーティング・システム・サービス６８に渡り、更にファイル・システム・ドライバ７２に移る。ファイル・システム・ドライバ７２は、次に、Ｉ／Ｏ要求をファイル名からディスク上の特定の場所に変換する。変換プロセスは、その特定の場所におけるディスクの読み出しまたは書き込みを行うべきデータ・ブロックの数も含むことができる。次に、この情報を、例えば、デバイス・ドライバ７４のような次のドライバに渡すことができる。デバイス・ドライバ７４が要求する情報を渡すプロセスは、図３では矢印７８および８０で示す。デバイス・ドライバ７４は、読み取りまたは書き込み対象のデータ・ブロックの場所および数を取り込み、それらを適切な制御信号に変換し、ハードウエア・デバイス８２から所望の情報を検索するか、あるいは所望の情報を格納する。次に、検索したデータは、デバイス・ドライバ７４からファイル・システム・ドライバ７２に渡

し、最終的に、戻り矢印８４で示すように、クライアント・プロセス６６に戻すことができる。ステータス情報も同様に戻すことができる。

【００３５】

図３では、Ｉ／Ｏ要求は、ファイル・システム・ドライバ７２とデバイス・ドライバ７４との間で直接受け渡すことはない。代わりに、Ｉ／Ｏマネージャ８６を介してドライバ間でＩ／Ｏ要求を受け渡す。しかしながら、Ｉ／Ｏマネージャは全ての実施態様において必ずしも必要な訳ではない。Ｉ／Ｏ要求を直接１つのドライバから別のドライバに渡すような実施形態も存在するであろう。

【００３６】

次に図４を参照し、本発明の一実施形態の上位図を示す。この実施形態は、本発明のいくつかの基本概念を示し、限定ではなく一例として与えるものである。即ち、この実施形態は、アクティブ・マウント・ポイントに関連するいくつかの基本概念を示す。マウント・ポイントとは、別の名称空間またはディレクトリ構造をマウントまたは移植すべき名称空間位置のことである。マウント・ポイントは、移植先の名称空間への接合部として作用する。アクティブ・マウント・ポイントによって、これを横断するときに、アクションを移植先の名称空間に取り込むことが可能となる。本発明では、マウント・ポイントは、１つのデバイスの名称空間から他のデバイスの名称空間の全部または一部への接合部を与えるために用いることができる。加えて、マウント・ポイントは、単一のデバイスにおいて、名称空間の一部分から名称空間の別の部分への接合部を備えるためにも使用可能である。したがって、この例は１つのデバイスの名称空間から別のデバイスへの移植を示すが、同じ原理は、あるデバイスの名称空間の一部分を、このデバイスの名称空間の別の部分に移植するときにも適用できる。

【００３７】

図４に示す実施形態では、ＣＤジュークボックス８８に一連のＣＤを装填してある。アクティブ・マウント・ポイント技術を用いるＩ／Ｏサブシステムは、個々のＣＤの名称空間を、ハードウエア・デバイス９０のような別のハードウエア・デバイスの名称空間全体に移植してある。移植プロセスの結果得られた論理名称空間の部分的な表示を、図４において９２として示す。この例では、ディレクトリ“my jukebox”は４つのサブディレクトリを有し、各々ＣＤジュークボックス８８内の１枚のＣＤを表わす。図４では、これらのＣＤを、“Hit of the Day”、“Encyclopedia”、“Current Events”、および“Baseball”として示す。論理名称空間は、図４では９４として示す、プログラム１およびプログラム２というような種々のユーザ・モード・プログラムに利用可能とすることができる。

【００３８】

ユーザ・モード・プログラム９４は、Ｉ／Ｏサブシステムからの Ｉ／Ｏ要求を行うあらゆるプログラムとすることができる。例えば、ユーザ・モード・プログラムの１つは、Microsoft Windows Explorerプログラムのような、ユーザに論理名称空間を表示するプログラムとすればよい。このようなプログラムは、ユーザが論理名称空間を詳しく調べ、ディレクトリまたはファイル上で種々の機能を実行することを可能にする。このようなアクションの１１つに、プログラム実行の開始がある。この例では、恐らくユーザはＣＤジュークボックス８８内にあるＣＤの１枚にアクセスしたいのであろう。ユーザは、このようなアクションを実行するには、ユーザに表示してある対応する名称を活性化すればよい。すると、名称を活性化するプロセスは、適切なデバイスの経路名を用いて、Ｉ／Ｏサブシステムへの Ｉ／Ｏ要求を開始する。

【００３９】

この例の目的のために、“Hit of Day”ＣＤは音楽ＣＤであり、“Encyclopedia”ＣＤ、“Current Events”ＣＤ、および“Baseball”ＣＤはデータＣＤであると仮定する。“Current Events”ＣＤの場合、更に、このＣＤはオンライン・データ・サービスにアクセスし、“Current Events”ＣＤ上の情報を更新することができると仮定する。第１の例として、ユーザは“Hit of the Day”ＣＤを活性化したと仮定する。Ｉ／Ｏ要求がＩ／Ｏサブシステムに渡されると、Ｉ／Ｏサブシステムは、このＩ／Ｏ要求に付随する経路名の解

明を始める。Ｉ／Ｏシステムは、例えば、ハードウエア・デバイス９０から、解明プロセスの間情報を検索する。“Hit of the Day”の名称をハードウエア・デバイス９０上でチェックしたなら、Ｉ／Ｏサブシステムは、アクティブ・マウント・ポイントを横断したことを認識する。次に、ハードウエア・デバイス９０上に格納してあるアクティブ・マウント・ポイント内の情報を抽出し、制御をアクティブ・マウント・ポイント・デバイス９６に渡すことができる。アクティブ・マウント・ポイント・デバイス９６は、ユーザが“Hit of the Day”ＣＤにアクセスしようとしていたことを認識し、適切なアクションを実行する。この場合、アクティブ・マウント・ポイント・ドライバ９６は、メディア・サービス・ドライバ９８を用いて、“Hit of the Day”ＣＤをＣＤジュークボックス８８に装填することができる。一旦“Hit of the Day”ＣＤをＣＤジュークボックス８８に装填したなら、アクティブ・マウント・ポイント・ドライバ９６は、自動的に“Hit of the Day”ＣＤを演奏する工程に進むことができる。このような工程は、メディア・サービス・ドライバ９８および／または別個のオーディオ・レンダリング・ドライバ（audio rendering driver）および関連するハードウエアの使用を必要とすることがある。これを図４において、例えば、他のドライバ１００および他のデバイス／システム１０２で示す。これらのドライバによるＣＤジュークボックス８８への直接アクセスを破線の矢印１０４で示す。アクティブ・マウント・ポイント・ドライバ９６がＣＤを装填するために用いる機構は、媒体を実装する手段の一例である。メディア・サービス・ドライバ９８が当該プロセスの一部である場合、メディア・サービス・ドライバ９８はこのような手段の一部をなす場合もあり得る。

【００４０】

他の例として、ユーザが“Current Events”ＣＤにアクセスしたと仮定する。以前と同様に、Ｉ／Ｏサブシステムがこのに関連するアクティブ・マウント・ポイントを横断したときに、制御がアクティブ・マウント・ポイント・ドライバ９６に移る。アクティブ・マウント・ポイント・ドライバ９６は、メディア・サービス・ドライバ９８を利用して適切なＣＤを装填する。しかしながら、この場合、アクティブ・マウント・ポイント・ドライバ９６は、オンライン・データ・サービスにアクセスし、ユーザに更新を与える工程も実行することもあり得る。

【００４１】

前述の例では、アクティブ・マウント・ポイント・ドライバ９６は、Ｉ／Ｏサブシステムが提供する情報に基づいて実行すべきアクションを決定することができた。他の状況では、アクティブ・マウント・ポイント・ドライバ９６は、他の情報、ドライバ、プロセス、システム等を頼りに、情報を提供するか判断を行う機能を備えなければならない場合もある。図４では、例えば、アクティブ・マウント・ポイント・ドライバ９６は、環境変数１０６から情報を受ける場合もある。環境変数１０６は、日時、ＣＤジュークボックス８８に現在装填中のＣＤ、またはその他の何らかの種類の情報というような、システム上で利用可能な情報を表わす。このような環境変数は、Ｉ／Ｏサブシステムによって、マウント・ポイント・ドライバ９６へ渡される情報の一部として供給することも可能である。アクティブ・マウント・ポイント・ドライバ９６が、実行すべきアクションについて決定する際、その補助として他のドライバ、プロセス、システム等を必要とする場合、制御を渡すことができ、あるいは適切なデバイスまたはコンポーネントから情報を要求することができる。

【００４２】

図４では、アクティブ・マウント・ポイント・ドライバ９６は、他のドライバ１００によって、他のデバイス１０２にアクセスすることも可能である。加えて、アクティブ・マウント・ポイント・ドライバ９６は、専用マウント・ポイント・プログラム１０８のような関連するプロセスに制御を引き渡したり、あるいはこれから指示または情報を受けることも可能である。プログラム１０８は専用マウント・ポイント・プログラムとして示しているが、アクティブ・マウント・ポイント・ドライバ９６は、アクティブ・マウント・ポイントの処理のみを専用とするのではないプログラムまたはプロセスも利用可能である。ア

クティブ・マウント・ポイントに出くわしたときに実行するアクションまたは複数のアクションに関する決定に供するこれらの実体のいずれにおいても、そのロジックは、アクションを選択する手段の一例である。

【００４３】

図４では、アクティブ・マウント・ポイント・ドライバ９６、メディア・サービス・ドライバ９８、およびその他のドライバ１００は、全てＩ／Ｏ要求を処理するドライバ手段の例である。図４では、実施態様によっては、アクティブ・マウント・ポイント・ドライバ９６は、Ｉ／Ｏサブシステムの一部としたり、異なるサブシステムの一部としたり、あるいはそれ自体のソフトウエア・サブシステムとする場合もあり得る。

【００４４】

図４の実施形態に示すように、アクティブ・マウント・ポイントに出くわした場合、１つのドライバからアクティブ・マウント・ポイント・ドライバに制御を移管する。したがって、本発明の範囲内に該当する実施形態は、受け取ったＩ／Ｏ要求を処理する制御を第１ドライバ手段から第２ドライバ手段に移管する手段を備えることができる。次に図５を参照し、このような手段を実現するために用いる一機構の例の基本概念を提示する。図５は、例えば、Ｉ／Ｏ処理を実行する複数のドライバ手段を利用するＩ／Ｏシステムを示す上位概念図を示す。クライアント・プロセス１１０は、Ｉ／Ｏ要求を行い、矢印１１４で示すように、最終的にオペレーティング・システム・サービス１１２に転送する。図５に示すＩ／Ｏシステムは、Ｉ／Ｏ処理を実行する複数のドライバ手段を備えている。限定ではなく一例として、図５では、このようなドライバ手段は、レイヤ１ドライバ１１６、レイヤ２ドライバ１１８、およびレイヤＮドライバ１２０で示す。

【００４５】

Ｉ／Ｏ要求はドライバ間で受け渡しされるので、本発明の範囲に該当する実施形態は、１つのドライバ手段から他のドライバ手段にＩ／Ｏ要求を受け渡す手段を備えることができる。一例として、図５では、このような手段は矢印１２２および１２４で示し、１つのドライバから他のドライバに直接受け渡すＩ／Ｏ要求を示す。このような手段は、Ｉ／Ｏ要求の１つのドライバから他のドライバへの移管を処理するＩ／Ｏマネージャを備える場合もある。このようなＩ／Ｏマネージャは図５のＩ／Ｏマネージャ１２６とすることができる。他の組み合わせも用いることができる。本質的に、１つのドライバ手段から他のドライバ手段にＩ／Ｏ要求を移管することを可能にする機構であれば、そのいずれもが、Ｉ／Ｏ要求を１つのドライバから他のドライバに受け渡す手段として使用するのに適しているであろう。

【００４６】

図５では、Ｉ／Ｏマネージャ１２６は、クライアント・プロセス１１０から受け取ったＩ／Ｏ要求を、レイヤ１ドライバ１１６に送出する。このようなＩ／Ｏ要求は、適切な情報を適切なドライバに転送するＩ／Ｏマネージャまたはその他のいずれかの機構がコールする関数またはサービスの形態を取ることができる。Microsoft Windows NTでは、例えば、メッセージ・ドリブン機構を用いて、Ｉ／Ｏシステムの種々のドライバ間で通信を行う。このシステムでは、Ｉ／Ｏ要求が発生すると、Ｉ／Ｏマネージャは Ｉ／Ｏ要求パケット（ＩＲＰ）を作成し、ＩＲＰを適切なドライバに送る。Ｉ／Ｏ要求を処理し他のドライバに送出するに連れて、情報がＩＲＰに追加され、このＩＲＰを次のドライバに渡すことができる。加えて、新たなＩＲＰを作成し次のドライバに送ることも可能である。ある状況では、ＩＲＰは、次のドライバに渡す前に、変更または「変形」（transmogrify）することも可能である。Microsoft Windows NTでは、Ｉ／Ｏマネージャは、ドライバ間でＩＲＰを転送する役割を担う。他のシステムでは、他の機構を用いる場合もある。このような実施態様の詳細は、設計選択事項と考えられ、本発明には重要ではない。

【００４７】

ここで図５を参照すると、Ｉ／Ｏ要求は、矢印１２２で示すように、種々のドライバを介して送出され、各ドライバは、いずれかの要求された処理を実行し、次いでＩ／Ｏ要求を次のドライバに送出する。図５は各ドライバが順番にＩ／Ｏ要求を受け取ることを示すが

、実施形態によっては、あるドライバを飛ばし、Ｉ／Ｏ要求を処理するのに必要なドライバのみが実際にＩ／Ｏ要求を処理するようにした方が望ましい場合もあることを注記しておく。

【００４８】

本発明の一実施形態では、複数のドライバを用いてＩ／Ｏ処理を実行する場合、アクティブ・マウント・ポイント属性を有するファイルまたはディレクトリに出くわしたときに、Ｉ／Ｏ処理を処理する通常のシーケンスを中断する機構が存在する。この場合、制御を他のドライバに渡し、アクティブ・マウント・ポイントに関連するＩ／Ｏ要求を処理すべきか否かについて判断を下す。したがって、本発明の範囲に該当する実施形態は、Ｉ／Ｏ要求の処理を中断する手段を備えることができる。図５では、このような手段は、例えば、レイヤＮドライバ１２０に組み込むことができる。本発明のこの実施形態では、アクティブ・マウント・ポイント属性を有するファイルまたはディレクトリに出くわした場合、通常の処理のシーケンスを中断する。

【００４９】

アクティブ・マウント・ポイント属性を認識した場合、Ｉ／Ｏ要求を処理する通常のシーケンスを保留とし、Ｉ／Ｏ要求の処理を完了する工程に移る。これらの工程は、Ｉ／Ｏ要求を処理する制御を異なるドライバに移管し、アクティブ・マウント・ポイントに対するＩ／Ｏ要求の処理に、当該ドライバが参加できるようにすることを伴う。したがって、本発明の範囲内に該当する実施形態は、１つのドライバ手段から他のドライバ手段に、受け取ったＩ／Ｏ要求を処理する制御を移管する手段を含むことができる。Ｉ／Ｏ要求の処理がアクティブ・マウント・ポイントを識別した場合に、Ｉ／Ｏ要求を処理するドライバから他のドライバに制御を移管するあらゆる機構が、利用可能である。図５では、このような機構は、例えば、矢印１２８によって示しており、これは、Ｉ／Ｏ要求の処理中にアクティブ・マウント・ポイントに出くわした場合に、レイヤＮドライバ１２０からレイヤ１ドライバ１１６にＩ／Ｏ要求を処理する制御を移管することを示す。以下で更に詳細に説明するが、１つのドライバから他のドライバに制御を移管する機構は、ある情報も転送し、制御を引き受けたドライバが、アクティブ・マウント・ポイントに関与するＩ／Ｏ処理を適正に処理できるようにするとよい。したがって、本発明の範囲内に該当する実施形態は、１つのドライバから他のドライバに情報を受け渡す手段も備えるとよい。

【００５０】

図５に示した実施形態では、レイヤ１ドライバ１１６は、アクティブ・マウント・ポイントに関与するＩ／Ｏ要求を処理する役割を担うアクティブ・マウント・ポイント・ドライバを表わす。一旦レイヤ１ドライバ１１６が制御を引き受けると、レイヤ１ドライバ１１６は、アクティブ・マウント・ポイントに関与するＩ／Ｏ要求を更に処理するために、あらゆる適切なアクションを実行することができる。実行可能なアクションのいくつかの例は、既に図４に関連付けて論じ更にそれ以外でも論じた。図５における実施形態は、簡略化した実施形態を表わし、レイヤ１ドライバ１１６は、他のエンティティを利用して、Ｉ／Ｏ要求を処理する間に完了すべきあるアクションを実行する。図５では、これらのエンティティは、例えば、他のエンティティ・ブロック１３０によって表わしてある。矢印１３２は、レイヤ１ドライバ１１６と他のエンティティ１３０との間で受け渡す制御および情報を表わす。他のエンティティ１３０は、例えば、他のドライバ、他のシステム、種々のデータ・ソースおよびデータ・パイプ、ユーザ・モードまたはカーネル・モード・プロセス、システム・サービス等を表わすことができる。また、レイヤ１ドライバ１１６は、矢印１３４で示すように、以前のドライバのいずれでも利用可能である。矢印１３４は、例えば、Ｉ／Ｏサブシステムが Ｉ／Ｏ要求の処理を再開する前に、開始または完了しなければならないいずれかのアクティブ・マウント・ポイント・アクションをレイヤ１ドライバ１１６が一旦実行した場合に、通常のＩ／Ｏ処理に制御を戻すことを表わす。Ｉ／Ｏ要求が完了した場合、矢印１３８で示すように、結果をクライアント・プロセスに戻すことができる。

【００５１】

レイヤＮドライバ１２０は、記憶装置１３６上でアクティブ・マウント・ポイントに出くわしたときを認識することを注記しておく。アクティブ・マウント・ポイント属性の詳細およびこれらがどのようにしてファイルまたはディレクトリの一部となるかについては、以下で更に詳細に示す。アクティブ・マウント・ポイントは、Ｉ／Ｏ要求のＩ／Ｏ処理におけるいずれの時点でも出くわし得ることを認めるのは重要である。しかしながら、通常このようなマウント・ポイントは、前述のような名称解明プロセスにおいて出くわす。更に、ディレクトリ階層において多数のマウント・ポイントを用いて多数の名称空間を単一の論理名称空間に移植する場合、アクティブ・マウント・ポイントは、以前のマウント・ポイントを介してアクセスしたデバイス上で出くわす場合もある。言い換えると、名称解明プロセスは、経路名の解明を進める際に、１つの記憶装置から、アクティブ・マウント・ポイントに出くわした第２の記憶装置に導くことができる。

【００５２】

最近のオペレーティング・システムでは、ファイルおよびディレクトリは、単に「属性」の集合体と考えることができる。属性とは、その最も抽象的なレベルにおいては、単にデータ格納位置のことである。異なる属性を用いて、ファイルまたはディレクトリの異なるプロパティを識別したり、あるいはファイルまたはディレクトリを扱わなければならないオペレーティング・システムおよびその他のプロセスが当該ファイルまたはディレクトリに関するある情報を知ることができるような、異なる種類または量の情報を保持するために用いる。例えば、ファイルは、プロセスに当該ファイルを識別させる名称属性、および当該ファイルに格納してあるデータを含むデータ属性を収容することができる。ファイルは、ファイルにアクセスできる人およびその方法を示すセキュリティ属性、タイム・スタンプ属性、ファイルが格納されているディレクトリを識別する属性のような、あらゆる数の他の属性でも有することができる。ディレクトリは、同様の種類の属性を収容することができるが、ディレクトリは、典型的には、ユーザが大量のデータを格納できるデータ属性を含まない。本発明のある種の実施形態では、アクティブ・マウント・ポイント属性をファイルまたはディレクトリに追加することによって、アクティブ・マウント・ポイントを識別する。このような属性はファイルまたはディレクトリのいずれかに追加することができるが、マウント・ポイントは、典型的に、他のデバイスの名称空間にリンクする特殊な種類のディレクトリとして考えられる。しかしかがら、完全を期すため、本発明は、アクティブ・マウント・ポイント属性をファイルまたはディレクトリのいずれかに追加することに言及する。

【００５３】

これより図６を参照し、本発明と共に用いて好適なファイルまたはディレクトリのいずれかに対する属性の図表を示す。これらの属性は、具体的にはMicrosoft Windows NTのために開発したＮＴＦＳファイル・システムが用いる属性の変更リストを表わす。ＮＴＦＳファイル・システムについては、Microsoft Press（マイクロソフト・プレス社）発行のHelen Custer（ヘレン・カスター）によるInside the Windows NT File System（Windows NTファイルシステムの内側）に詳細に記載されており、この言及によりその内容は本願にも含まれるものとする。図６では、ファイルまたはディレクトリを構成する属性は、２つの基本的なグループに分割することができる。第１基本グループは、全体的に１４０で示してあり、ファイルおよびディレクトリ双方に共通の属性を表わす。１４２で示す第２基本グループは、あるファイル（左側に示す）またはディレクトリ（右側に示す）に特定の属性を含む。図６は、限定ではなく一例として与えたものであって、システムが使用する属性のいずれの集合体でも、アクティブ・マウント・ポイント属性に格納されている情報が適切なドライバによって識別し検索できるのであれば、本発明との使用に適していると考えられる。

【００５４】

属性１４０は、標準情報属性１４４、属性リスト１４６、名称属性１４８、セキュリティ記述属性１５０、アクティブ・マウント・ポイント属性１５２、およびその他のシステム属性１５４から成る。標準情報属性１４４は、読み取り専用、読み取り／書き込み、隠れ

等、ファイルに対する標準的な「ＭＳ－ＤＯＳ」属性、ファイルまたはディレクトリのタイム・スタンプ、および当該ファイルを指し示すディレクトリ数を表わす。属性リスト１４６は、ファイルまたはディレクトリが、マスタ・ファイル・テーブルに記録されている１つの記憶レコード以上を占める場合に、ＮＴＦＳがファイルまたはディレクトリを構成する追加の属性の場所を識別するために用いる属性である。マスタ・ファイル・テーブルは、ファイルまたはディレクトリの常駐属性全てを格納してある場所である。名称属性１４８は、ファイルまたはディレクトリの名称である。ファイルまたはディレクトリは、ＮＴＦＳ内に多数の名称属性を有することができ、例えば、長い名称、短いＭＳ－ＤＯＳ名称等がある。セキュリティ記述子属性１５０は、Windows NTが、ファイルまたはディレクトリを所有する者およびそれにアクセスできる者を指定するために用いるデータ構造を含む。これらの属性については、Inside the Windows NT File Systemに更に詳しく記載されている。

【００５５】

アクティブ・マウント・ポイント属性１５２は、本発明によって追加した新たな属性である。アクティブ・マウント属性１５２は、特定のファイルまたはディレクトリを、特定のドライバによる特殊な処理に必要なアクティブ・マウント・ポイントとして識別する。このアクティブ・マウント・ポイント属性は、２つの目的を達成可能にする十分な情報を含むことが好ましい。第１の目的は、当該アクティブ・マウント・ポイントを処理すべき特定のドライバ（アクティブ・マウント・ポイントのオーナー）を識別できなければならないことである。加えて、最大の柔軟性を得るためには、アクティブ・マウント・ポイントのオーナーは、後にオーナーが用いてアクティブ・マウント・ポイントを正しく処理することができるように、アクティブ・マウント・ポイントに関連するデータを格納できることが好ましい。アクティブ・マウント・ポイントに関するこれ以上の情報を以下に与える。他のシステム属性１５４は、ファイルまたはディレクトリと共に格納する他のいずれのシステム属性を代表する。

【００５６】

グループ１４２のファイル属性は、データ属性１５６および１５８、ならびにその他の属性１６０から成る。ファイルは、典型的に、１つ以上のデータ属性を有する。データ属性１５６および１５８は、ユーザ制御データを格納することができる場所を表わす。殆どのシステムにおけるファイルは、単一のデータ属性を備えている。しかしながら、ＮＴＦＳでは、多数のデータ属性が許される。ＮＴＦＳでは、ファイルは１つの無名の属性を有し、残りのデータ属性は全て名称のあるデータ属性である。データ属性に関するこれ以上の情報は、Inside the Windows NT File Systemにおいて見出すことができる。他の属性１６０は、ユーザまたはユーザ・プロセスが作成し格納する、その他のユーザ属性を代表する。これらの属性は、ユーザが望むいずれの関数にも使用可能である。

【００５７】

ディレクトリ属性は、例えば、インデックス・ルート属性１６２、インデックス割り当て属性１６４、ビットマップ属性１６６、およびその他の属性１６８から成るものとすることができる。これらの属性に関するこれ以上の情報は、先に言及し本願にも含まれるものとした、Inside the Windows NT File Systemにおいて見出すことができるが、本質的に、インデックス・ルート属性１６２は、ディレクトリが収容するファイルに対するインデックスを収容し、インデックス割り当て属性１６４はデータ・ブロックまたは「クラスタ」マッピングに関する情報を収容し、ビットマップ属性８２は、どのクラスタが使用中でありどのクラスタが空いているかを追跡する。その他の属性１６８によって示すように、その他の属性もディレクトリの一部として定義し、格納することが可能である。

【００５８】

これまでの論述は、特定の形式のファイルまたはディレクトリに関していくらか詳細な部分にまで入り込んだが、これは例示に過ぎず、本発明の範囲を限定するものとして解釈すべきではない。本発明は、ファイルまたはディレクトリの既存の属性にアクティブ・マウント・ポイントを追加した、いずれの形式のファイルまたはディレクトリとでも機能する

ものである。あるいは、既存の属性を利用し、アクティブ・マウント・ポイント属性情報を格納し、したがってファイルまたはディレクトリ内の既存の属性数を増大させることなく、アクティブ・マウント・ポイント属性を含ませる方法を等価的に得ることも可能な場合もある。

【００５９】

アクティブ・マウント・ポイント属性１５２は、マウント・ポイントを識別可能にする情報を格納し、マウント・ポイントは当該マウント・ポイントのオーナーが処理することが好ましい。したがって、これらの目標の達成を可能にするいずれの情報の組み合わせでも、アクティブ・マウント・ポイント属性１５２に用いることができる。一実施形態では、アクティブ・マウント・ポイント属性として、リパーズ・ポイント属性（reparse point attribute）を用いる。リパーズ・ポイント属性、およびこれらをどのように用いてＩ／Ｏ要求の処理を中断し制御を他のドライバに移管して処理するかに関するこれ以上の情報は、FILE SYSTEM PRIMITIVE ALLOWING REPROCESSING of I/O REQUESTS BY MULTIPLE DRIVERS IN A LAYERED DRIVER I/O SYSTEM（階層状ドライバＩ／Ｏシステムにおいて多数のドライバによるＩ／Ｏ要求の再処理を可能にするファイル・システム・プリミティブ）と題する米国特許出願第０８／２３９，５９３号（以後「リパーズ・ポイント出願」と呼ぶ）において見出すことができる。その内容は、この言及により本願にも含まれるものとする。次に図７を参照し、アクティブ・マウント・ポイント属性として用いて好適な属性の基本構造を示す。図７に示すアクティブ・マウント・ポイント属性は、タグ１７０および値１７２から成る。

【００６０】

アクティブ・マウント・ポイントは特定のドライバによって処理するので、本発明の範囲内に該当する実施形態は、特定のドライバをアクティブ・マウント・ポイントのオーナーとして識別する手段を備える。アクティブ・マウント・ポイントに関与するＩ／Ｏ要求の少なくとも一部を処理するドライバとして、特定のドライバを識別するいずれの機構でも、このような手段に使用可能である。アクティブ・マウント・ポイントが図７に示す構造を有する場合、このような手段は、例えば、タグ値１７０を備えるとよい。図７に示すアクティブ・マウント・ポイントでは、タグ１７０は、当該アクティブ・マウント・ポイントのオーナーのＩＤを含むデータ・ワードである。タグは、アクティブ・マウント・ポイントのオーナーを識別可能にするように割り当てなければならない。このようにタグを割り当てる機構であれば、いずれでも利用可能である。例えば、どのシステムにドライバをインストールするかには拘らず、同じタグを常に同じオーナーのドライバと関連付けるように、タグを割り当てることができる。例えば、タグ値のブロックを種々のドライバ製造業者に割り当てる、中央リポジトリまたはクリアリング・ハウスがあるとよい。この場合、ドライバ製造業者は、タグを特定のドライバに割り当てることができる。タグ値を多くとも単一のドライバに関連付けさせるような機構であれば、他のあらゆるものも使用可能である。あるいは、動的にローカル・タグ値を割り当て、ドライバのインストール中にタグ値をシステムによって割り当てるようにすることも可能な場合がある。アクティブ・マウント・ポイントのオーナーを識別可能とするあらゆる機構を、タグ値を割り当てるために使用することができる。

【００６１】

図７に示すアクティブ・マウント・ポイントは、オーナー制御の値フィールド１７２も含む。オーナー制御の値１７２は、アクティブ・マウント・ポイントのオーナーが、アクティブ・マウント・ポイントを適正に処理するために必要な任意の種類の情報を配置し得る場所を表わす。例えば、オーナーは、アクティブ・マウント・ポイントに出くわしたときに実行すべきアクションに関する決定を可能にする情報を挿入することができる。あるいは、オーナーに、情報を取得すべきその他の場所を識別させる情報、またはどんな他のドライバ、システム、サブシステム等が、アクティブ・マウント・ポイントを処理するために実行すべきアクションに関する判断を行うべきかを識別させる情報を、値１７２に格納することも可能である。

【００６２】
図７には示さないが、値１７２は、データ長インジケータによって処理することも可能である。この記憶フォーマットでは、データ・フィールドの長さを格納しておき、値を完成するにはどの位のデータを読み取らなければならないのかについて確認する。あるいは、実施形態によっては、固定長、またはポインタまたはリンクと共に連鎖する情報のブロックを利用する値を格納する方が一層効率的な場合もある。本質的に、値のフィールドを完成するにはどの位のデータを読み取らなければならないのかを特定する機構であれば、いずれでも利用可能である。また、オーナー・ドライバが格納する必要があり得るデータがどれくらいであるかについても、考慮すべきである。このような考慮は、どのようにデータ・フィールドを格納するか、およびデータ・フィールドの最大可能長に影響を及ぼす。
【００６３】
これより図８を参照し、図７のタグ１７０の一実施形態の更に詳細な図を提示する。図８の実施形態では、タグ１７０は３２ビット・ワードである。図８に示すように、種々の形式の情報をワードにエンコードする。基本的に、図８のタグは、３つの異なるエリアに分割する。ビット０～１５はタグ値１７４に割り当てる。これらのビットは、ドライバを作成する会社が、一意のタグ値を割り当て、彼らのドライバをアクティブ・マウント・ポイントのオーナーとして識別するために利用可能である。既に説明したように、このような値は、中央の機関によって割り当てることができ、あるいは種々の手順およびプロトコルを用いて、システムによって動的に割り当てることも可能である。ドライバがアクティブ・マウント・ポイントのオーナーをそれ自体であると識別できる限り、タグ値１７４を割り当てる機構としていずれでも利用可能である。ビット１６～２８は、予約フィールド１７６から成る。これらのビットは、今後の使用のために確保しておく。図８では１７８として示すビット２９は、名称代理フラグ（name surrogate flag）である。このビットをセットすると、ファイルまたはディレクトリは、システム内の別の名称付きエンティティを表わすことになる。これによって、ファイルまたはディレクトリが、他の名称付きエンティティの代理として作用することが可能となる。
【００６４】
図８では１８０として示すビット３０は、高レイテンシ・ビットである。１にセットすると、このタグを有するファイルまたはディレクトリは、データの最初のバイトを検索するのに長いレイテンシを有するものと考えられる。既に説明したように、本発明は、あらゆるデバイスの名称空間も、他のあらゆるデバイスの名称空間に移植するために使用可能である。状況によっては、ユーザまたはプログラムが移植部分にアクセスしたときに、要求を行った時点から最初のデータ・バイトが戻ってくるまでに高いレイテンシが生ずる場合がある。このような状況に出くわす可能性があるのは、例えば、アクティブ・マウント・ポイントがある記憶場所から媒体を検索し、その媒体をデバイスにマウントし、データを読み取ることができるように媒体を位置決めしなければならない場合である。これは、例えば、テープ・サイロ（tape silo）を他のデバイスの名称空間に移植した場合であろう。同様の状況は、インターネットまたはその他のネットワークのデータ・プロバイダを、デバイスの名称空間に移植する場合に、生ずる可能性がある。インターネットからデータを検索するためには、デュアル・アップ・ライン上で接続を確立するために、アクティブ・マウント・ポイントが必要となる場合がある。これは、レイテンシ・ビットを必ずセットすることになる程のレイテンシを招く可能性がある。
【００６５】
図８では１８２で示すビット３１は、アクティブ・マウント・ポイントのオーナーがMicrosoftドライバであることを示すために予約してある。セットすると、オーナー・ドライバがMicrosoftドライバであることを、他のドライバに確認させることができる。このフラグを含ませることによって、オペレーティング・システムは、内部ドライバに関連するタグを素早くソートすることができ、更に他のものが、それに属さないドライブを素早くソートすることができる。図８に示す構造は、先に言及し本願にも含まれるものとした、リパーズ・ポイント特許出願に記載されているリパーズ・ポイント属性との使用にも好適

な場合もある。

【００６６】

次に、図９を参照し、本発明の一実施形態の更に詳細な図を提示する。この実施形態では、クライアント・プロセス１８４は、矢印１８８で示すように、システム・サービス１８６にコールすることによって、Ｉ／Ｏ要求を開始する。Ｉ／Ｏマネージャ１９０は、Ｉ／Ｏ要求１９２をＩ／Ｏサブシステムのドライバに送出する。図９では、Ｉ／Ｏ要求１９２は、アクティブ・マウント・ポイント・ドライバ１９４、ファイル・システム・ドライバ１９６、およびデバイス・ドライバ１９８間で受け渡されるものであることがわかる。しかしながら、前述のように、実施形態によっては、Ｉ／Ｏ要求１９２は、当該Ｉ／Ｏ要求に直接関与するドライバにのみ導出する場合もある。Ｉ／Ｏ要求１９２は、ドライバ間で個々のＩ／Ｏ要求を受け渡すいずれかの機構とすればよい。このような機構は、直接関数コール、メッセージおよびその他の機構で構成することができる。Ｉ／ＯシステムがWindows NTの一部である場合、Ｉ／Ｏ要求１９２は、既に説明したようにＩＲＰ内に組み込むことができる。

【００６７】

Ｉ／Ｏ要求の処理中にアクティブ・マウント・ポイントに出くわした場合、デバイス・ドライバＡは、デバイスＡ２００上のディレクトリまたはファイルが、関連するアクティブ・マウント・ポイント属性を有することを認識する。アクティブ・マウント・ポイント属性の検出時に、デバイスＡは適切なアクティブ・マウント情報を抽出し、この情報を次の階層状ドライバに戻す。このプロセスは、図９において、アクティブ・マウント・ポイント情報２０２によって示す。アクティブ・マウント・ポイント２０２は、例えば、タグと、恐らくデバイス・ドライバ１９８が出くわすアクティブ・マウント・ポイントの値とから成る。

【００６８】

ファイル・システム・ドライバ１９６がアクティブ・マウント・ポイント情報２０２を受け取ると、それをアクティブ・マウント・ポイントとして認識する。ファイル・システム・ドライバ１９６がそのように構成してある場合、ファイル・システム・ドライバ１９６はタグをチェックし、アクティブ・マウント・ポイントを処理する役割を担うのでないことを確認する。この例では、しかしながら、アクティブ・マウント・ポインタは、アクティブ・マウント・ポイント・ドライバ１９４によって処理する。したがって、ファイル・システム・ドライバ１９６はアクティブ・マウント・ポイント情報２０２を次に高い層のドライバに渡す。他のドライバがアクティブ・マウント・ポイント・ドライバ１９４とファイル・システム・ドライバ１９６との間にある場合、各ドライバはアクティブ・マウント・ポイント情報２０２を受け取り、タグを検査して、アクティブ・マウント・ポイントを処理する役割を引き受けるのか否か確認する機会を有する。

【００６９】

しかしながら、最終的には、アクティブ・マウント・ポイント・ドライバ１９４がアクティブ・マウント・ポイント情報２０２を受け取る。次いで、これは、それ自体をアクティブ・マウント・ポイントのオーナーとして認識し、適切な処置を講じて、Ｉ／Ｏ要求の完了を促進するために必要なあらゆるアクションを遂行する。例えば、アクティブ・マウント・ポイントがリムーバブル媒体デバイスの名称空間を移植している場合、アクティブ・マウント・ポイント・ドライバ１９４は、読み取りまたは書き込みのために適切な媒体を突き止め、選択し、マウントし、配置するために新たな要求を発生しなければならない可能性がある。次いで、アクティブ・マウント・ポイント・ドライバ１９４は、Ｉ／Ｏ要求を発行し、クライアント・プロセス１８４が開始したＩ／Ｏ要求１９２を完了する。Ｉ／Ｏ処理のために媒体をマウントし媒体を位置付けるためにアクティブ・マウント・ポイント１９４を必要とする場合、実施形態は媒体をマウントする手段を備えればよい。このような手段は、アクティブ・マウント・ポインタ・ドライバ１９４が適切なデバイス上に適切な媒体をマウントする際に使用するいずれの機構でもよい。ＣＤジュークボックス、テープ・サイロ等は全て、これらの機能を達成するために活性化することが必要なインター

フェースを有する。典型的に、アクティブ・マウント・ポイント・ドライバ１９４は、別のドライバまたはシステムを介して、これらの機能にアクセスする。

【００７０】

アクティブ・マウント・ポイント・ドライバ１９４は、アクティブ・マウント・ポイントに出くわした場合に実行する必要があるアクションを決定するので、アクティブ・マウント・ポイント・ドライバ１９４は、アクションを選択する手段を有することができる。このような手段は、アクティブ・マウント・ポイント・ドライバ１９４に、クライアント・プロセス１８４からのＩ／Ｏ要求の処理に対して、更に開始または完了すべきアクションまたは複数のアクションを決定させる機構であればいずれでもよい。前述のように、実行するとよいアクションを決定するためには、アクティブ・マウント・ポイント・ドライバは、種々のドライバ、情報ソース、プロセス、システム、サブシステム等を利用することができる。これら種々の項目を図９に示す。例えば、情報は、環境変数２０４のような、種々の環境変数から得ることができる。環境変数２０４は、アクティブ・マウント・ポイント・ドライバ１９４が動作する環境に関する情報のあらゆるソースを代表する。このような環境変数は、日時、システムのハードウエア構成、あるいは動作環境またはシステムに関するその他のあらゆる情報等の種々のシステム・レベルの情報を含むことができる。環境変数２０４から情報を取得することに加えて、情報は、アクティブ・マウント・ポイント・プロセス２１０に関連するデバイスＢ２０６のようなその他のソース、またはその他のデバイス、システム、ドライバ２１２から得ることも可能である。

【００７１】

情報を提供することに加えて、他のデバイス、システム、ドライバ、関連するプロセス等は、決定プロセスにも参加することができる。したがって、アクションを選択する手段は、その他のデバイス、システム、ドライバ、関連するプロセス等を含むこともできる。このような状況では、アクティブ・マウント・ポイント・ドライバ１９４は、適切なエンティティに要求を行うか、あるいは制御を引き渡し、当該エンティティが発生したあらゆる応答または結果を受け取る。したがって、アクションを選択する手段は、アクティブ・マウント・ドライバ１９４内だけでなく、本発明が動作するシステムまたはネットワーク全体にわたるその他のエンティティ内にも常駐するとよい。関連するアクティブ・マウント・ポイント・プロセス２１０との通信を、要求２１４、矢印２１６、矢印２１８、および応答２２０によって表わす。デバイス・ドライバ２０８を介したデバイス２０６との通信を、要求２２２および応答２２４で示す。他のデバイス、システムおよびドライバとの通信は、要求２２６および応答２２８によって示す。アクティブ・マウント・ポイント・ドライバ１９４は、他のエンティティを用いて所望の機能を遂行する場合に、１つのソフトウエア・コンポーネントが他のソフトウエア・コンポーネントにコールまたはアクセスする際に利用可能なあらゆる種類の機構を利用できることは当業者には認められよう。このようなコールまたはアクセスは、ローカルであったり、あるいは先に説明したようにネットワークや他の通信会社を通じてのリモートであることもあり得る。

【００７２】

図９は、本発明の追加の一面も示す。この一面は、アクティブ・マウント・ポイント・ドライバ１９４が、リモート・ディスクのようなリモート・データ・ソース、他の記憶媒体、あるいはインターネットまたはイントラネットのようなデータ・パイプからのデータにアクセスするという場面を検討することによって、説明することができる。このような状況では、ユーザがリモート位置からの情報にアクセスしたい場合、アクティブ・マウント・ポイントを利用して、リモート・デバイスの名称空間をローカル・デバイスの名称空間に移植することができる。次いで、ユーザは、リモート・データにアクセスするために特別なアクションを実行することなく、透過的にデータにアクセスすることができる。このような状況では、リモート・データのローカル・キャッシュを作成し、リモート・データへのその後のアクセスを高速化することが望ましい場合もある。例えば、恐らく、アクティブ・マウント・ポイント・ドライバ１９４は、ユーザがアクセスした種々のリモート・データを検索したり、当該アクティブ・マウント・ポイント・ドライバは、ユーザが近い

将来アクセスすると予測する場合がある。この情報は、次にローカル記憶装置上にキャッシュし、次のアクセス時にデータへのアクセスを迅速化することができる。このプロセスは、図９では、デバイスＡ２００に書き込むキャッシュ・データ２３０によって示す。データをキャッシュした後、アクティブ・マウント・ポイントに出くわした場合、アクティブ・マウント・ポイント・ドライバ１９４が、リモート位置からデータを検索するために必要なレイテンシを発生する代わりに、デバイスＡ２００から情報を検索することが可能となる。この特徴を組み込んだ実施形態は、第１デバイスからアクセスしたデータをキャッシュする手段を備え、第１デバイスにそれ以上アクセスすることなく、情報を検索できるようにするとよい。

【００７３】

要約すると、本発明は、１つのデバイスの名称空間を他のデバイスの名称空間に移植する場合に、任意のアクションを実行するロバスト性の高い機構を提供する。本発明は、アクティブ・マウント・ポイントの概念を用い、アクティブ・マウント・ポイントを横断したときに、システムにあらゆる所望のアクションを実行することを可能にする。このようなアクションは、データの読み取りまたは書き込み、媒体のマウント等のような従来からＩ／Ｏ要求に関連するアクションだけでなく、Ｅ－メールの送信、プロセスの実行開始、オーディオＣＤのマウントおよび演奏、あるいはシステム内においてソフトウエアによって開始または完了することができるその他のあらゆる種類のアクションのような、通常ではＩ／Ｏ要求に関連のないアクションも含むことができる。この機構は非常にロバスト性が高いので、当初の設計者が予想しないような場面への拡張を可能とし、しかも現在の実施態様に不当なインパクトを与えることもない。

【００７４】

本発明は、その精神または本質的な特徴から逸脱することなく、他の特定的な形態においても具体化することができる。記載した実施形態は、いかなる観点においても、限定ではなく例示として見なすべきである。したがって、本発明の範囲は、前述の説明ではなく、添付した請求の範囲によって示すものとする。特許請求の範囲の均等の意味および意味に該当する全ての変更は、その範囲に包含するものとする。

【図面の簡単な説明】

【図１】　１つのデバイスの名称空間の別のデバイスの名称空間への移植を示す図である。

【図２】　本発明に適した動作環境を備えたシステムの一例である。

【図３】　複数の階層状ドライバを有するＩ／Ｏシステムの一例である。

【図４】　本発明の一実施形態の上位図である。

【図５】　特殊な属性に出くわした場合に、制御を特定のドライバに渡すことを示す図である。

【図６】　本発明と共に用いて好適な属性のリストである。

【図７】　アクティブ・マウント・ポイント属性の一実施形態の図である。

【図８】　アクティブ・マウント・ポイント属性のタグの一実施形態を示す図である。

【図９】　本発明の別の実施形態の上位図である。

【図１】

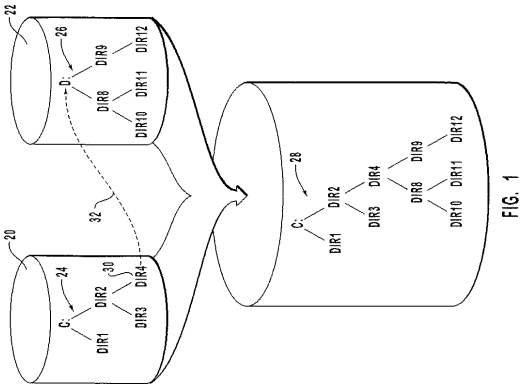
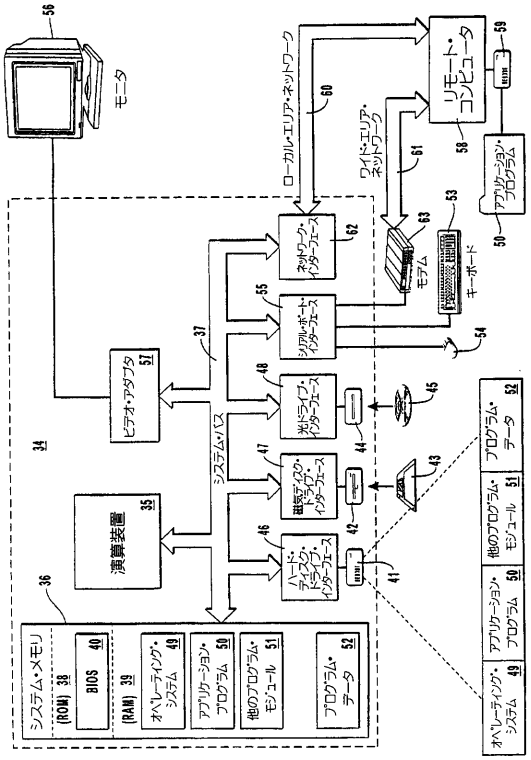
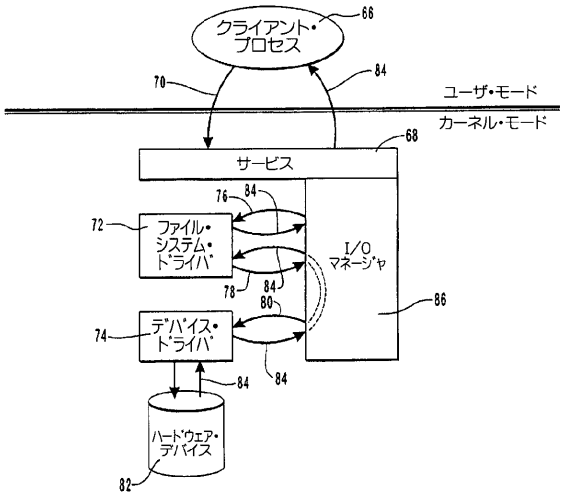


FIG. 1

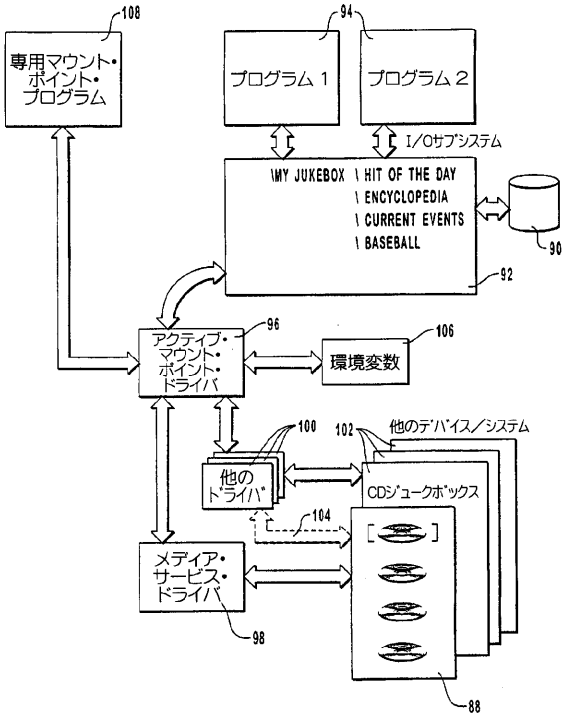
【図２】



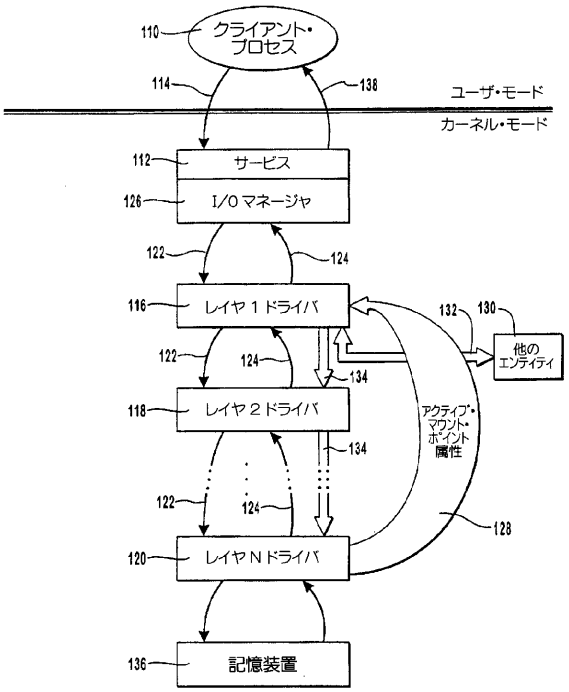
【図３】



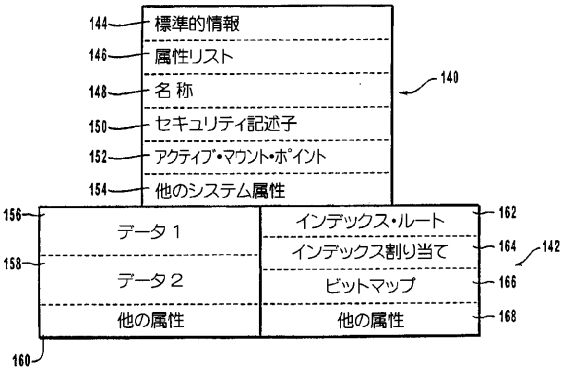
【図４】



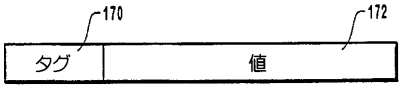
【図５】


110
クライアント・プロセス
114
138
ユーザ・モード
カーネル・モード
112
サービス
126
I/Oマネージャ
122
124
116
レイヤ１ドライバ
132
130
他のエンティティ
134
118
レイヤ２ドライバ
アクティブ・マウント・ポイント属性
120
レイヤＮドライバ
128
136
記憶装置

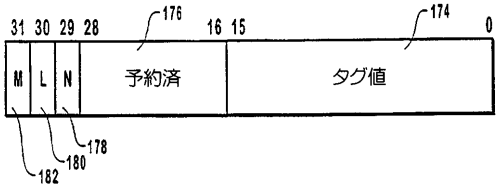
【図６】


144
標準的情報
146
属性リスト
148
名称
140
150
セキュリティ記述子
152
アクティブ・マウント・ポイント
154
他のシステム属性
156
データ１
インデックス・ルート
162
インデックス割り当て
164
142
158
データ２
ビットマップ
166
他の属性
他の属性
168
160

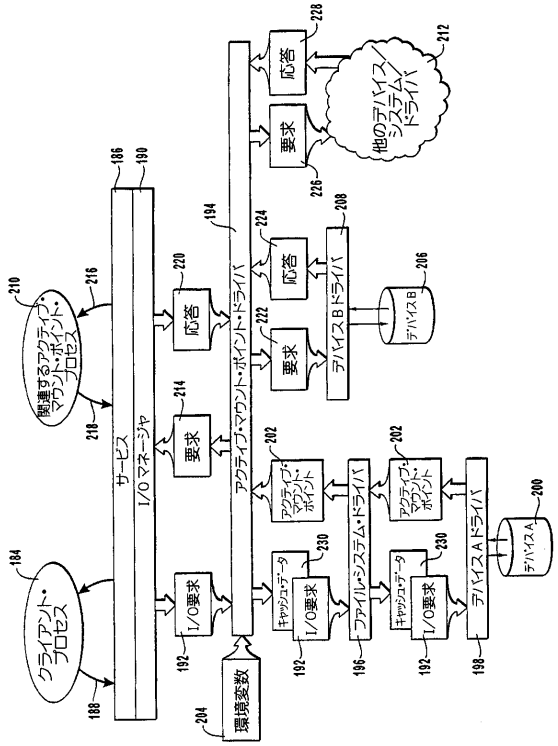
【図７】


170
172
タグ
値

【図８】


31 30 29 28
176
16 15
174
0
M
L
N
予約済
タグ値
178
180
182

【図９】


228
応答
212
他のデバイス・システム・ドライバ
要求
226
186
190
194
208
224
応答
デバイスBドライバ
206
デバイスB
210
関連するアクティブ・マウント・ポイント・プロセス
216
220
応答
222
要求
218
214
要求
サービス
I/Oマネージャ
アクティブ・マウント・ポイント・ドライバ
202
202
アクティブ・マウント・ポイント
アクティブ・マウント・ポイント
200
デバイスA
184
クライアント・プロセス
I/O要求
キャッシュ・データ
230
I/O要求
ファイル・システム・ドライバ
キャッシュ・データ
230
I/O要求
デバイスAドライバ
192
192
196
192
198
188
環境変数
204

フロントページの続き

(72)発明者 モモー，オショマ
アメリカ合衆国ワシントン州９８１０２，シアトル，サミット・アベニュー“イー” ４２１，ナンバー ３０３
(72)発明者 キャブレラ，ルイス・フェリペ
アメリカ合衆国ワシントン州９８００４，ベルビュー，キラーニー・ウェイ・サウス・イースト ２００９
(72)発明者 キムラ，ゲイリー・ディー
アメリカ合衆国ワシントン州９８０３３，カークランド，ノース・イースト・フォーティサード・プレイス １１８２０

審査官 高瀬 勤

(56)参考文献 特開平０４－２９１６３８（ＪＰ，Ａ）
特開平０８－３３９３５５（ＪＰ，Ａ）
Herman C. Rao, Towards a National Collaboratory: An Internet File System, ICSI '92. Proceedings of the Second International Conference on Systems Integration, 米国，１９９２年 6月１５日，pp.489-498
Maurice J. Bach，ＵＮＩＸカーネルの設計，日本，共立出版株式会社，１９９７年 3月１５日，初版，第１０２－１０９頁

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 12/00
JSTPlus(JDreamII)