

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3759969号
(P3759969)

(45) 発行日 平成18年3月29日(2006.3.29)

(24) 登録日 平成18年1月13日(2006.1.13)

(51) Int. Cl.

F I

G 0 6 F 12/02 (2006.01)

G 0 6 F 12/02 5 3 0 B

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 0 1 M

請求項の数 24 (全 15 頁)

(21) 出願番号 特願平6-139810
 (22) 出願日 平成6年6月22日(1994.6.22)
 (65) 公開番号 特開平7-28693
 (43) 公開日 平成7年1月31日(1995.1.31)
 審査請求日 平成13年6月20日(2001.6.20)
 (31) 優先権主張番号 08/086334
 (32) 優先日 平成5年6月30日(1993.6.30)
 (33) 優先権主張国 米国(US)

前置審査

(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 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 100077481
 弁理士 谷 義一
 (74) 代理人 100088915
 弁理士 阿部 和夫
 (74) 復代理人 100120581
 弁理士 市原 政喜

最終頁に続く

(54) 【発明の名称】 ディスクスペースを管理する変型バディシステム

(57) 【特許請求の範囲】

【請求項1】

所与のサイズのディスクスペースをもつディスク記憶装置を備えたデータ処理システムにおいて、ディスクスペースを管理する方法が、

(a) ディスクスペースを、最上レベルから最下レベルまでの範囲の多数の次々に減少するレベルのノードを有するツリー構造としてモデリングし、各ノードは、ディスクスペースのイクステントを表し、各レベルごとに、そのレベル内のノードは同様のサイズのイクステントを表し、最下レベル以外の各レベルの各ノードは親ノードであって、この親ノードを、次に続くレベルにおいて兄弟となる子ノードに接続するリンクを有し、子ノードによって表されたイクステントは、親ノードによって表されたイクステントの完全な細分を構成し、これらノードによって表されたイクステントのサイズは各次々のレベルにおいて減少し、そして各次々のレベルごとにノードによって表されたイクステントのサイズの和は、ディスクスペースの所与のサイズに等しく、

(b) 各ノードに関する状態情報を、そのノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かっているか否かを指定するビットマップに記憶し、そして

(c) 上記ビットマップ及びツリー構造を使用して、ディスクスペースを管理する、という段階を備え、

ビットマップ及びツリー構造を使用してディスクスペースを管理するという上記段階は、更に、ノードによって表されたディスクスペースのイクステントを割り当てる段階を含

10

20

み、

ノードによって表されたディスクスペースのイクステントを割り当てる上記段階は、更に、

(i) ノードに関してビットマップに記憶された状態情報を更新して、ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようし、

(ii) 上記ノードの親ノードに関してビットマップに記憶された状態情報を検討して、その親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるか否かを判断し、

(iii) 親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分からないと判断された場合は、親ノードに関してビットマップに記憶された状態情報を更新して、親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようし、そして

10

(iv) 親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるか判断された場合は、親ノードに関してビットマップに記憶された状態情報を維持する、という段階を備えたことを特徴とする方法。

【請求項2】

(v) 親ノードを現在ノードとして表し、

(vi) 現在ノードに対して付加的な親ノードがあるかどうかを判断し、

(vii) 現在ノードに対して付加的な親ノードがある場合は、その付加的な親ノードに関してビットマップに記憶された状態情報を検討して、その付加的な親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるか否かを判断し、

20

(viii) 現在ノードに対して付加的な親ノードがない場合、又は現在ノードに対する親ノードによって表された付加的なイクステントがあり且つ付加的な親ノードが少なくとも部分的に割り当てられていると分かっている場合には、ビットマップに記憶された状態情報を維持し、そして

(ix) 現在ノードに対する付加的な親ノードがあり且つ付加的な親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分からない場合は、付加的な親ノードに関してビットマップに記憶された状態情報を更新して、付加的な親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようし、という段階を更に備えた請求項1に記載の方法。

30

【請求項3】

(x) 付加的な親ノードを現在ノードとして表し、

(xi) 現在ノードに対して新たな付加的な親ノードがあるかどうかを判断し、そして

(xii) 新たな付加的な親ノードがある場合は、その新たな付加的な親ノードを付加的な親ノードとして使用して上記段階(vii)ないし(ix)を繰り返す、という段階を更に備えた請求項2に記載の方法。

【請求項4】

所与のサイズのディスクスペースをもつディスク記憶装置を備えたデータ処理システムにおいて、ディスクスペースを管理する方法が、

40

(a) ディスクスペースを、最上レベルから最下レベルまでの範囲の多数の次々に減少するレベルのノードを有するツリー構造としてモデリングし、各ノードは、ディスクスペースのイクステントを表し、各レベルごとに、そのレベル内のノードは同様のサイズのイクステントを表し、最下レベル以外の各レベルの各ノードは親ノードであって、この親ノードを、次に続くレベルにおいて兄弟となる子ノードに接続するリンクを有し、子ノードによって表されたイクステントは、親ノードによって表されたイクステントの完全な細分を構成し、これらノードによって表されたイクステントのサイズは各次々のレベルにおいて減少し、そして各次々のレベルごとにノードによって表されたイクステントのサイズの和は、ディスクスペースの所与のサイズに等しく、

(b) 各ノードに関する状態情報を、そのノードによって表されたイクステントが少な

50

くとも部分的に割り当てられていると分かっているか否かを指定するビットマップに記憶し、そして

(c) 上記ビットマップ及びツリー構造を使用して、ディスクスペースを管理する、という段階を備え、

ビットマップ及びツリー構造を用いてディスクスペースを管理する上記段階は、1つのノードによって表されたディスクスペースのイクステントを割り当て解除する段階を更に備え、

ノードによって表されたディスクスペースのイクステントを割り当て解除する上記段階は、更に、

(i) ノードに関してビットマップに記憶された状態情報を更新して、ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分からないことを指示するようにし、 10

(ii) ノードが少なくとも1つの兄弟ノードを有するかどうか判断し、

(iii) ノードが少なくとも1つの兄弟ノードを有する場合には、各兄弟ノードに関してビットマップに記憶された状態情報を検討して、兄弟ノードによって表されたイクステントの各々が少なくとも部分的に割り当てられていると分からないかどうかを判断し、そして兄弟ノードによって表されたイクステントの各々が少なくとも部分的に割り当てられていると分からないと判断された場合は、親ノードに関してビットマップに記憶された状態情報を更新して、親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分からないことを指定するようにする、という段階を備えたことを特徴とする方法。 20

【請求項5】

ディスクスペースの割り当てを、多数の次々に減少するレベルのノードを有するツリー構造としてモデリングする上記段階は、ツリー構造の最上レベルにある単一のノードに、全てのディスクスペースを含むイクステントを表示させる段階を更に備えている請求項1または4記載の方法。

【請求項6】

ディスクスペースを、多数の次々に減少するレベルのノードを有するツリー構造としてモデリングする上記段階は、ツリー構造の最下レベルにあるノードのそれぞれに、サイズがディスクセクタであるイクステントを表示させる段階を更に備えた請求項1または4記載の方法。 30

【請求項7】

ビットマップ及びツリー構造を用いてディスクスペースを管理する上記段階は、更に、

(i) ツリー構造においてノードの1つによって表されたディスクスペースのイクステントを割り当てて、情報を保持するようにし、そして

(ii) ビットマップに保持された状態情報を更新して、その割り当てられたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようにする、という段階を備えた請求項1または4記載の方法。

【請求項8】

ビットマップ及びツリー構造を用いてディスクスペースを管理する上記段階は、更に、 40

(i) ツリー構造においてノードの1つによって表されたディスクスペースのイクステントを割り当て解除し、そして

(ii) ビットマップに保持された状態情報を更新して、その割り当て解除されたイクステントノードが少なくとも部分的に割り当てられていると分からないことを指定するようにする、という段階を備えた請求項1または4記載の方法。

【請求項9】

所与のサイズのディスクスペースをもつディスク記憶装置を備えたデータ処理システムにおいて、

(a) ディスクスペースを、最上レベルから最下レベルまでの範囲の多数の次々に減少するレベルのノードを有するツリー構造としてモデリングし、各ノードは、ディスクス 50

ースのイクステントを表し、各レベルごとに、そのレベル内のノードは同様のサイズのイクステントを表し、最下レベル以外の各レベルの各ノードは親ノードであって、この親ノードを、次に続くレベルにおいて兄弟となる子ノードに接続するリンクを有し、子ノードによって表されたイクステントは、親ノードによって表されたイクステントの完全な細分を構成し、これらノードによって表されたイクステントのサイズは各次々のレベルにおいて減少し、そして各次々のレベルごとにノードによって表されたイクステントのサイズの和は、ディスクスペースの所与のサイズに等しく、

(b) 最上レベルで始まって各次々のレベルごとに幅優先形態で続けられるような幅優先形態でノードに対する状態ビットが順序付けされるような順序付けビットマップに各ノードごとに少なくとも1つの状態ビットを記憶し、各ノードの状態ビットは、そのノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるかどうかを指定するものであり、そして

10

(c) 上記ビットマップ及びツリー構造を使用して、ディスクスペースを管理する、という段階を備え、

ビットマップ及びツリー構造を使用してディスクスペースを管理する上記段階は、更に、ノードの1つによって表されたイクステントを割り当てる段階を備え、

ノードによって表されたイクステントを割り当てる上記段階は、

(i) ノードに関してビットマップに記憶された状態情報を更新して、ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようにし、

20

(ii) 上記ノードの親ノードに関してビットマップに記憶された状態情報を検討して、その親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるか否かを判断し、

(iii) 親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分からないと判断された場合は、親ノードに関してビットマップに記憶された状態情報を更新して、親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようにし、そして

(iv) 親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるか判断された場合は、親ノードに関してビットマップに記憶された状態情報を維持する、という段階を備えたことを特徴とする方法。

30

【請求項10】

(v) 親ノードを現在ノードとして表し、

(vi) 現在ノードに対して付加的な親ノードがあるかどうかを判断し、

(vii) 現在ノードに対して付加的な親ノードがある場合は、その付加的な親ノードに関してビットマップに記憶された状態ビットを検討して、その付加的な親ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるか否かを判断し、

(viii) 現在ノードに対して付加的な親ノードがない場合、又は現在ノードに対して付加的な親ノードがあり且つ付加的な親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分かっている場合には、ビットマップに記憶された状態ビットを維持し、そして

40

(ix) 現在ノードに対して付加的な親ノードがあり且つ付加的な親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分からない場合には、付加的な親ノードに関してビットマップに記憶された状態ビットを更新して、付加的な親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定するようにする、という段階を更に備えた請求項9に記載の方法。

【請求項11】

(x) 付加的な親ノードを現在ノードとして表し、

(xi) 現在ノードに対して新たな付加的な親ノードがあるかどうかを判断し、そして

(xii) 新たな付加的な親ノードがある場合は、その新たな付加的な親ノードを付加的な親ノードとして使用して上記段階(vii)ないし(ix)を繰り返す、という段階を更に備え

50

た請求項 1 0 に記載の方法。

【請求項 1 2】

所与のサイズのディスクスペースをもつディスク記憶装置を備えたデータ処理システムにおいて、

(a) ディスクスペースを、最上レベルから最下レベルまでの範囲の多数の次々に減少するレベルのノードを有するツリー構造としてモデリングし、各ノードは、ディスクスペースのイクステントを表し、各レベルごとに、そのレベル内のノードは同様のサイズのイクステントを表し、最下レベル以外の各レベルの各ノードは親ノードであって、この親ノードを、次に続くレベルにおいて兄弟となる子ノードに接続するリンクを有し、子ノードによって表されたイクステントは、親ノードによって表されたイクステントの完全な細分を構成し、これらノードによって表されたイクステントのサイズは各次々のレベルにおいて減少し、そして各次々のレベルごとにノードによって表されたイクステントのサイズの和は、ディスクスペースの所与のサイズに等しく、

10

(b) 最上レベルで始まって各次々のレベルごとに幅優先形態で続けられるような幅優先形態でノードに対する状態ビットが順序付けされるような順序付けビットマップに各ノードごとに少なくとも1つの状態ビットを記憶し、各ノードの状態ビットは、そのノードによって表されたイクステントが少なくとも部分的に割り当てられていると分かるかどうかを指定するものであり、そして

(c) 上記ビットマップ及びツリー構造を使用して、ディスクスペースを管理する、
という段階を備え、

20

ビットマップ及びツリー構造を用いてディスクスペースを管理する上記段階は、1つのノードによって表されたディスクスペースのイクステントを割り当て解除する段階を更に備え、

(i) ノードに関してビットマップに記憶された状態ビットを更新して、ノードによって表されたイクステントが少なくとも部分的に割り当てられていると分からないことを指示するようにし、

(ii) ノードが少なくとも1つの兄弟ノードを有するかどうか判断し、

(iii) ノードが少なくとも1つの兄弟ノードを有する場合には、各兄弟ノードに関してビットマップに記憶された状態ビットを検討して、兄弟ノードによって表されたイクステントの各々が少なくとも部分的に割り当てられていると分かるかどうかを判断し、そして親ノードに関してビットマップに記憶された状態ビットを更新して、親ノードにより表されたイクステントが少なくとも部分的に割り当てられていると分からないことを指定するようにする、という段階を備えたことを特徴とする方法。

30

【請求項 1 3】

順序付けされたビットマップに各ノードに対する状態ビットを記憶する上記段階は、更に、順序付けされたビットマップにおいて兄弟ノードに対する状態ビットのすぐ隣にノードに対する状態ビットを記憶するという段階を備えた請求項 9 または 1 2 に記載の方法。

【請求項 1 4】

ディスクスペースをツリー構造としてモデリングする上記段階は、更に、ディスクスペースを2進ツリー構造としてモデリングする段階を含む請求項 9 または 1 2 に記載の方法

40

【請求項 1 5】

ノードのサイズは、各次々のレベルにおいて半分ずつ減少する請求項 1 4 に記載の方法。

【請求項 1 6】

ディスクスペースを、多数の次々に減少するレベルのノードを有するツリー構造としてモデリングする上記段階は、ツリー構造の最上レベルにある単一のノードに、全てのディスクスペースを含むイクステントを表示させる段階を更に含む請求項 9 または 1 2 に記載の方法。

【請求項 1 7】

50

ディスクスペースを、多数の次々に減少するレベルのノードを有するツリー構造としてモデリングする上記段階は、ツリー構造の最下レベルにあるノードのそれぞれに、サイズがディスクセクタであるイクステントを表示させる段階を更に備えた請求項 9 または 12 に記載の方法。

【請求項 18】

所与のサイズのディスクスペースをもつディスク記憶装置を備えたデータ処理システムにおいて、ディスク記憶装置に自由ディスクスペースを位置決めする方法が、

(a) ディスクスペースを、最上レベルから最下レベルまでの範囲の多数の次々に減少するレベルのノードを有するツリー構造としてモデリングし、各ノードは、ディスクスペースのイクステントを表し、各レベルごとに、そのレベル内のノードは同様のサイズのイクステントを表し、最下レベル以外の各レベルの各ノードは親ノードであって、この親ノードを、次に続くレベルにおいて兄弟となる子ノードに接続するリンクを有し、子ノードによって表されたイクステントは、親ノードによって表されたイクステントの完全な細分を構成し、これらノードによって表されたイクステントのサイズは各次々のレベルにおいて減少し、そして各次々のレベルごとにノードによって表されたイクステントのサイズの和は、ディスクスペースの所与のサイズに等しく、

(b) 各ノードに関する状態情報を、イクステントが少なくとも部分的に割り当てられていると分かっているか否かを指定するビットマップに記憶し、

(c) 所望サイズのディスクスペースのイクステントを表すツリー構造におけるノードの選択された1つに対してビットマップの状態情報を検討し、

(d) 上記選択されたノードについてのビットマップの状態情報が、その選択されたノードにより表されたイクステントが少なくとも部分的に割り当てられていると分からないことを指定する場合には、(i) その選択されたノードの各兄弟に関してビットマップの状態情報を検討して、その兄弟のいずれかが少なくとも部分的に割り当てられていると分かるかどうか判断し、そして(ii) いずれの兄弟も少なくとも部分的に割り当てられていると分かった場合には、その選択されたノードが自由であると結論する、という段階を備えたことを特徴とする方法。

【請求項 19】

上記選択されたノードについての状態情報が、その選択されたノードにより表されたイクステントが少なくとも部分的に割り当てられていると分かることを指定し、そしてその選択されたノードと同じレベルにおいてツリー構造に付加的なノードが存在する場合には、その選択されたノードと同じレベルにおいてツリー構造の次のノードを選択されたノードとして指定しそして上記段階(d)で始まる方法を繰り返すという段階を更に備えた請求項 18 に記載の方法。

【請求項 20】

最上レベルで始まって各次々のレベルのノードごとに幅優先形態で続けるようにツリー構造におけるノードの順序に基づいて幅優先形態でビットマップに状態情報を記憶する段階を更に含む請求項 19 に記載の方法。

【請求項 21】

上記次のノードは、上記ビットマップにおいて上記選択されたノードのすぐ隣に状態情報を持つノードである請求項 20 に記載の方法。

【請求項 22】

上記次のノードは、ビットマップにおいて上記選択されたノードと同じレベルのノードの中で所定ノードに対する状態情報の最も近くに記憶された状態情報をもつと共に上記方法でまだ検討されていない状態情報をもつノードである請求項 20 に記載の方法。

【請求項 23】

少なくとも部分的に割り当てられていると分かった兄弟情報がないことが決定された場合には、その選択されたノードの親ノードに関して上記ビットマップの状態情報を検討して、その親ノードが少なくとも部分的に割り当てられていると分かるかどうか判断し、そしてその親ノードが少なくとも部分的に割り当てられていると分からない場合には、その

親ノードを選択されたノードとして指定しそして上記段階（d）で始まる方法を繰り返す、という段階を更に備えた請求項 18 に記載の方法。

【請求項 24】

親ノードが少なくとも部分的に割り当てられていると分かると決定された場合には、その選択されたノードと同じレベルにあるツリー構造内の次のノードであって、その選択されたノード及びその選択されたノードの兄弟以外のノードを、選択されたノードとして指定し、そして上記段階（d）で始まる方法を繰り返す、という段階を更に備えた請求項 23 に記載の方法。

【発明の詳細な説明】

【0001】

10

【産業上の利用分野】

本発明は一般にデータ処理システムに係り、より詳細には、データ処理システムにおいてディスクスペースを管理するための変型バディシステムに係る。

【0002】

【従来の技術】

ディスク上の自由スペースは、一般に、3つの方法の1つで管理されている。第1の手法においては、ファイルに対するディスクスペースの割り当てが、ファイルに割り当てられたセクタのリンクされたリストによって記述される。第2の手法においては、ディスクスペースのどのセクタが割り当てられたかを監視するようにビットマップが設けられる。ビットマップには各セクタごとにビットが与えられる。セクタに関連したビットの値が、そのセクタが割り当てられたか否かを指定する。第3の手法においては、イクステント（即ち、ディスク上のセクタの隣接セット）において割り当てが行われる。このように、個々のセクタではなくて隣接セクタの範囲でスペースが割り当てられる。

20

【0003】

【発明の構成】

本発明の第1の特徴によれば、所与のサイズのディスクスペースをもつディスク記憶装置を備えたデータ処理システムにおける方法が実施される。この方法においては、ディスクスペースは、多数のレベルのノードを有するツリー構造として最初にディスクスペースをモデリングすることによって管理され、上記レベルは最上レベルから最下レベルまで順次に減少する。各レベルの各ノードは、ディスクスペースのイクステントを表し、そして各レベルごとに、そのレベル内のノードは、同様のサイズのイクステントを表す。ノードによって表されたイクステントのサイズは、レベルの低下と共に減少するが、ノードの数は、レベルの低下と共に増加する。各次々のレベルに対するノードのサイズの和は、所与のサイズのディスクスペースに等しい。各レベルの各ノードはディスクスペースのイクステントを表し、そして各レベルごとに、そのレベル内のノードは、同様のサイズのイクステントを表す。最下レベル以外の各レベルの各ノードは親ノードであって、この親ノードを次に続くレベルにおける子ノードに接続するリンクを有している。子ノードは互いに兄弟となる。子ノードは、親ノードによって表されたイクステントの細分であるイクステントを表す。各ノードに関する状態情報がビットマップに記憶され、そのノードによって表されたイクステントが少なくとも部分的に割り当てられると分かっているか否かについて指定する。ツリー構造のビットマップは、ディスクスペースを管理するのに使用される。

30

40

【0004】

状態情報は、ツリー構造における各ノードごとに少なくとも1つの状態ビットを構成する。ビットマップは、最上レベルで始まって各次々のレベルに対して巾優先形態で続けられるような巾優先形態でノードに対する状態ビットが順序付けされる順序付けビットマップであるのが好ましい。このようにビットマップを順序付けすることは、兄弟ノード及び親ノードに対して参照の局所性を与えることができる。

【0005】

【実施例】

本発明の好ましい実施例においては、ディスクスペースがノードのツリー構造としてモデ

50

リングされ、ツリーの各ノードはディスクスペースのイクステントを表しそしてそれに関連した状態を有している。各ノードの状態情報は、参照の局所性を利用した形態で順序付けされたビットマップに記憶される。このビットマップは、ツリー構造に対応する形態で順序付けされ、兄弟の状態ビットを隣接位置に保持する。本発明の好ましい実施例にツリー構造及びビットマップを使用することにより、自由ディスクスペースを迅速に位置決めして割り当てることができる。更に、ツリー構造及びビットマップを使用することに関連したオーバーヘッドは最小である。というのは、ビットマップ及びツリー構造によってメモリスペースが消費されないからである。

【0006】

図1は、本発明の好ましい実施例に適したデータ処理システム10のブロック図である。このデータ処理システム10は、中央処理ユニット(CPU)12を備え、これはメモリ14及びディスク記憶装置16へアクセスする。本発明の好ましい実施例は、以下に詳細に述べる変型バディシステムを採用することによりディスク記憶装置16のディスクの自由スペースを管理する。又、データ処理システム10は、キーボード18、マウス20及びビデオディスプレイ22も備えている。

【0007】

本発明の好ましい実施例は、メモリ14に保持されたオペレーティングシステム24において実施される。本発明は、オペレーティングシステムで実施される必要はなく、オペレーティングシステムと共に機能する個別のコードモジュールで実施されてもよいことが当業者に明らかであろう。図2は、本発明の好ましい実施例の変型バディシステムを実施するためのツリー構造及びビットマップを構成するようにオペレーティングシステム24によって実行される段階を示すフローチャートである。図2は、図3及び4と共に説明する。最初に、ディスク記憶装置16(図1)の使用できるディスクスペースが、図3に示すようなツリー構造に論理的に仕切られる(図2のステップ26)。このツリー構造32(図3)の最上レベルは、ツリー構造のルートノード34である。これは、ディスク記憶装置16の使用できる全ディスクスペース(即ち、図3では4K)のイクステントを表している。ツリー構造32の次のレベルは、ディスクスペースのより小さなイクステントを表すノードを保持する。このノードによって表されたイクステントは、ルートノード34によって表されたイクステントの完全な細分を構成する。図3に示された例では、ノード36及び38は、使用できるディスクスペースの半分のイクステント(即ち、ディスクスペースの使用可能な4Kのうちの2K)を表す。ツリー構造32の次のレベルは、手前のレベルのノード36及び38によって表されたものより小さいイクステントを表すノード40、42、44及び46を保持する。より詳細には、ノード40及び42によって表されたイクステントは、ノード36によって表されたイクステントの完全な細分を表し、そしてノード44及び46によって表されたイクステントは、ノード38によって表されたイクステントの完全な細分を表す。この最下レベル40、42、44及び46におけるノードの各々は、サイズが1Kであり、ディスク記憶装置16に使用できる割り当ての最小サイズ単位であるイクステントを表す。

【0008】

ツリー構造32のノードの各々は、次々のレベル間でノードを接続する論理的なリンクを含んでいる。例えば、ノード34は、リンク47によってノード36及び38に接続される。ノード34は親ノードと称され、ノード36及び38は子ノードと称される。構造体32の各レベルにおいて、子ノードによって表されたイクステントは、親ノードによって表されたイクステントの完全な細分を構成する。ノード36及び38は互いに兄弟である。これらの兄弟ノードは、同じイクステント(親ノードによって表された)細分であるイクステントを表すという点でバディとしても知られている。「バディシステム」は、バディ表示法を用いることによってメモリスペースを割り当て/割り当て解除するための公知解決策である。

【0009】

図3に示されたツリー構造は単なる説明に過ぎず、各親ノードは3つ以上の子ノードを含

10

20

30

40

50

んでもよいことを理解されたい。

【 0 0 1 0 】

上記ツリー構造 3 2 の各ノード 3 4、3 6、3 8、4 0、4 2、4 4 及び 4 6 には状態が指定される。この状態はビットで表される。ノードの状態ビットの値「1」は、そのノードにより表されたイクステントが割り当てられるか又は部分的に割り当てられると分かっていることを示している。これに対して、ノードの状態ビットの値「0」は、そのノードにより表されたイクステントが割り当てられるか又は部分的に割り当てられると分からないことを示している。最初に、全てのノードは、その状態ビットが値「0」である。これらの状態ビットは、ツリー構造に基づいて順序付けされたビットマップ 4 8 (図 4) に記憶される。このビットマップ 4 8 は、ツリー構造 3 2 (図 3) のノードに対する全ての状態ビットを保持する。ビットマップ 4 8 のビットは、ルートノード 3 4 で始めて巾優先形態で順序付けされる。従って、ビットマップ 4 8 のビット 1 (図 4) は、ルートノード 3 4 の状態情報を保持する。ビット 2 は、ノード 3 6 の状態情報を保持し、そしてビット 3 は、ノード 3 8 の状態情報を保持する。同様に、ビットマップ 4 8 のビット 4 は、ノード 4 0 の状態情報を保持し、ビット 5、6 及び 7 は、各々、ノード 4 2、4 4 及び 4 6 の状態情報を保持する。この構成は、親のビット位置が分かったときに子ノードの状態ビットを迅速に位置決めできるようにしそしてその逆も行うことができる。特に、ノン・リーフノード n (その状態ビットがビットマップの位置 n にある) の場合に、その子供はビットマップにおいて位置 $2n$ 及び $2n+1$ となる。

【 0 0 1 1 】

ツリー構造 3 2 及びビットマップ 4 8 を使用するために、オペレーティングシステム 2 4 は、割り当てファンクション、割り当て解除ファンクション、探索ファンクション及びテストファンクションをサポートする。これらファンクションの各々はツリー構造 3 2 のノードに基づいて動作し、これについては以下で詳細に説明する。これらファンクションの説明は、最初にテストファンクションに焦点を合わせる。というのは、テストファンクションは他のファンクションのうちの選択されたファンクションによって使用されるからである。

【 0 0 1 2 】

図 5 は、2 進ツリーの実施に対するテストファンクションによって行われるステップのフローチャートである。このファンクションは、各々 N 個の子供を有する N 進ツリーに対して機能するように形成できることも理解されたい。テストファンクションは、ノードによって表されたイクステントが自由であるか否かをテストする。テストされるべきノードは、パラメータとしてテストファンクションへ通される。オペレーティングシステム 2 4 は、最初に、当該ノードが値「1」の状態ビットを有するかどうかチェックし、この値「1」は、そのノードにより表されたイクステントが割り当てられるか又は部分的に割り当てられると分かっていることを指示するものである (図 5 のステップ 5 0)。ノードの状態ビットが値「1」を有する場合は、テストファンクションは、そのノードによって表されたイクステントが自由でないという結果を返送する (図 5 のステップ 5 2)。一方、ノードの状態ビットが値「0」を有する場合には、そのノードによって表されたイクステントが自由であるかどうか分からない。ノードは値「0」の状態ビットを有していて、その親又は他の祖先が既に割り当てられているために既に割り当てられていることもあるし、或いはノードによって割り当てられたイクステントが自由の場合もある。テストファンクションは、次いで、ノードがルートノードであるかどうかチェックする (ステップ 5 4)。ノードがルートノードである場合には、値「0」の状態ビットを有するので、そのノードによって表されたイクステントは自由であり (割り当てることのできる親を持たない)、そしてテストファンクションは、ノードによって表されたイクステントが自由であるという指示を返送する (ステップ 5 6)。

【 0 0 1 3 】

ノードがルートノードでない場合には、テストファンクションは、兄弟ノードの状態ビットの値を検査する (ステップ 5 8)。例えば、ノード 3 6 (図 3) の状態が検査される場

10

20

30

40

50

合には、ノード 38 が唯一の兄弟ノードであり、ノード 38 の状態が図 5 のステップ 58 で検査される。テストファンクションは、いずれかの兄弟ノードが値「1」の状態ビットを有するかどうかチェックする（ステップ 60）。より一般化されたテストファンクションにおいては、値「1」の状態ビットをもつ兄弟が見つかるか、又は兄弟に対する全ての状態ビットが検査されてしまうまで、全ての兄弟ノードに対する状態ビットが検査される。いずれかの兄弟ノードが値「1」の状態ビットを有する場合には、親ノードによって表されたイクステントは完全に割り当てられない。従って、テストファンクションは、当該ノードによって表されたイクステントが自由であるという指示を返送する（ステップ 62）。

【0014】

全ての兄弟ノードの状態ビットが値「0」をもつ場合には、親ノードを現在ノードとして考え（ステップ 64）そしてステップ 50 で始まる前記のステップを繰り返すことにより、プロセスはツリー構造 32 を登るように続けられる。

【0015】

上記したように、オペレーティングシステム 24 は、割り当てファンクションをサポートする。この割り当てファンクションは、ディスク記憶装置 16 のディスクスペースに自由スペース（即ち、イクステント）を割り当てる。図 6 は、割り当てファンクションによって実行されるステップのフローチャートである。割り当てられるべきイクステントを表すノードに対する状態ビットには、値「1」が指定される（図 6 のステップ 76）。次いで、ノードが親ノードをもつかどうかの判断がなされる（ステップ 78）。ノードが親ノードをもたない場合には、それ以上のステップを実行する必要はない。これに対し、ノードが親ノードをもつ場合には、その親ノードの状態ビットの値が検査され、親ノードが割り当てられたか又は部分的に割り当てられたと表示された（即ち、値「1」を有する）かどうか判断される（ステップ 80）。親ノードが割り当てられたか又は部分的に割り当てられたと既に表示された場合には、それ以上のステップを実行する必要はない。しかしながら、親ノードが割り当てられたか又は部分的に割り当てられたと表示されていないときには、親ノードの状態ビットが値「1」をもつように更新されて、親ノードによって表されたイクステントの少なくとも一部分が割り当てられると分かることを指示しなければならない（ステップ 82）。このプロセスは、次いで、親ノードを現在ノードとして考えることによりツリー構造 32 を登るように続けられる（ステップ 84）。次いで、ステップ 78 で始まる上記ステップが繰り返される。各親ノードの少なくとも一部分が割り当てられているので、ビットマップ 48 に保持された状態ビットは、上記ファンクションによって値「1」に更新されねばならない。

【0016】

又、オペレーティングシステム 24 は、ディスク記憶装置 16 におけるディスクスペースのイクステントを割り当て解除するファンクションもサポートする。割り当て解除されるべきイクステントを表すノードは、パラメータとして、割り当て解除ファンクションへ通される。図 7 は、オペレーティングシステム 24 によってサポートされる割り当て解除ファンクションのフローチャートである。ノードによって表されたイクステントを割り当て解除すべきときには、そのノードの状態ビットが値「0」をもつように表示される（図 7 のステップ 86）。そのノードによって表されたイクステントは、親ノードによって表されたイクステントの唯一の部分であるか又は割り当てられた他の祖先によって表されたイクステントであるから、親及び祖先ノードも、値「0」をとるように更新された状態ビットをもつべきであるかどうか判断しなければならない。親ノード及び祖先ノードの状態ビットを更新するプロセスは、ノードがルートノードであるかチェックする（ステップ 88）によって開始される。ノードがルートノードである場合には、親ノードも祖先ノードもたず、それ故、それ以上のステップは必要とされない。これに対して、ノードがルートノードでない場合には、兄弟ノードの状態ビットが検査される（ステップ 90）。いずれかの兄弟ノードの状態ビットが値「1」である（即ち、兄弟ノードの全ての状態ビットが「0」でない）場合は、親ノードの状態ビットを更新することができない。というのは、

10

20

30

40

50

親ノードが表すイクステントは、少なくとも部分的に割り当てられると分かっているからである。全ての兄弟ノードの状態ビットが値「0」を有する場合には、親ノードによって表されたイクステントは当然自由でなければならない。というのは、子ノードによって表された全てのイクステントが割り当てられないか又は部分的に割り当てられないからである。従って、親ノードの状態ビットは値「0」に変更される（ステップ92）。このプロセスは、親ノードを現在ノードとして考えそしてステップ88で始まる上記ステップを繰り返すことにより付加的な祖先ノードに対して続けられる。この形態では、全ての祖先は必要に応じて更新される。

【0017】

オペレーティングシステム24は、更に、ツリー構造32（図3）内の自由ノードを探索する探索ファンクションもサポートする。図8は、この探索ファンクションによって実行されるステップのフローチャートである。探索ファンクションによって実行されるサーチは、所望サイズのイクステントを有するツリーのレベルにおいて次のノードの状態ビットを見ることによって開始される（ステップ96）。特に、所望量のディスクスペースが、ツリー構造32のノードによって表されたいずれのイクステントのサイズにも一致しない場合には、必要とされるものよりも大きな最も一致に近いイクステントがステップ96において選択される。このイクステントのサイズを以下「所望のサイズ」と称する。ディスクスペースの所望サイズイクステントを有するツリーのレベルにおいて、探索ファンクションは、値「0」の状態ビットを有するレベルにノードがあるかどうかを決定する（ステップ98）。値「0」の状態ビットを有するレベルにノードがない場合には、使用できる所望サイズのイクステントはない。

【0018】

しかしながら、値「0」の状態ビットを有するレベルにノードがある場合は、そのノードの兄弟ノードが検査され、いずれかの兄弟ノードが値「1」の状態ビットを有するかどうか判断される（ステップ100）。いずれかの兄弟ノードが値「1」の状態ビットを有する場合には、そのノードによって表されたイクステントが探索ファンクションによって返送され、使用される（ステップ102）。一方、全ての兄弟ノードに対する状態ビットが値「1」を有する場合には、親ノードの状態ビットが検査される（ステップ104）。特に、親ノードの状態ビットが値「1」をもつかどうかの判断がなされる（ステップ106）。親ノードが値「1」の状態ビットをもつ場合には、親ノードをルートとして始まるサブツリーはそれ以上検査される必要がないことに注意されたい（ステップ108）。自由ノードのサーチは、ステップ96で再び続けられる。親ノードの状態ビットが値「1」を有しそして親ノードがルートノードでない場合には（ステップ112）、親ノードの兄弟ノードを検査することにより上記ステップがステップ100で始めて繰り返される。親ノードがルートノードである場合には（ステップ112を参照）、探索ファンクションは、現在ノードによって表されたイクステントを自由なものとして返送する（ステップ110）。

【0019】

探索ファンクションの上記説明においては、テストファンクションが探索ファンクションに直接一体化されたことが明らかであろう。テストファンクションと探索ファンクションを一体化すると、テストされねばならない非常に多数のノードの検査をアルゴリズムがスキップできることにより、効率利得を得ることができる。

【0020】

上記の探索ファンクションは、ツリー構造体のあるレベルにおいて自由な第1イクステントを位置決めするように設計されている。この探索ファンクションに採用された手法を若干変更することにより、ディスクの所与の位置の付近に置かれた自由なイクステントを探すようなファンクションを形成することができる。このような変更は、サーチプロセスが開始する場所で重要である。ツリーのターゲットレベルにおいて第1のイクステントを探すのではなく、この変更された手法は、所望の位置に最も近いノードを最初に探す。おそらく自由であるイクステントをサーチするときは、この変更された手法は、その最も近い

ノードから外方へ進んで、他の近傍のノードを検査し、そしてその近傍のノードによって表されたイクステントをテストし、それらが自由であるかどうか調べる。見つかった第1の自由なイクステントが、最も近い自由なイクステントである。

【0021】

好ましい実施例について本発明を説明したが、特許請求の範囲に規定された本発明の範囲から逸脱せずに、その形態及び細部に種々の変更がなされ得ることが当業者に明らかであろう。例えば、本発明に使用されたツリー構造32は、2進ツリーである必要はない。各親ノードは3つ以上の子ノードを有してもよい。更に、割り当てられた又は部分的に割り当てられた状態が「1」ビット値に関連される必要はなく、「1」ビット値と「0」ビット値の意味を交換してもよい。

10

【図面の簡単な説明】

【図1】本発明の好ましい実施例に適したデータ処理システムのブロック図である。

【図2】本発明の好ましい実施例を実施するために行われるステップを示すフローチャートである。

【図3】本発明の好ましい実施例に使用されるツリー構造体を示す図である。

【図4】図3のツリー構造体のビットマップを示す図である。

【図5】本発明の好ましい実施例のテストファンクションにおいて実行されるステップを示すフローチャートである。

【図6】本発明の好ましい実施例の割り当てファンクションにおいて実行されるステップを示すフローチャートである。

20

【図7】本発明の好ましい実施例の割り当て解除ファンクションにおいて実行されるステップを示すフローチャートである。

【図8】本発明の好ましい実施例の探索ファンクションにおいて実行されるステップを示すフローチャートである。

【符号の説明】

10 データ処理システム

12 中央処理ユニット

14 メモリ

16 ディスク記憶装置

18 キーボード

30

20 マウス

22 ビデオディスプレイ

24 オペレーティングシステム

32 ツリー構造

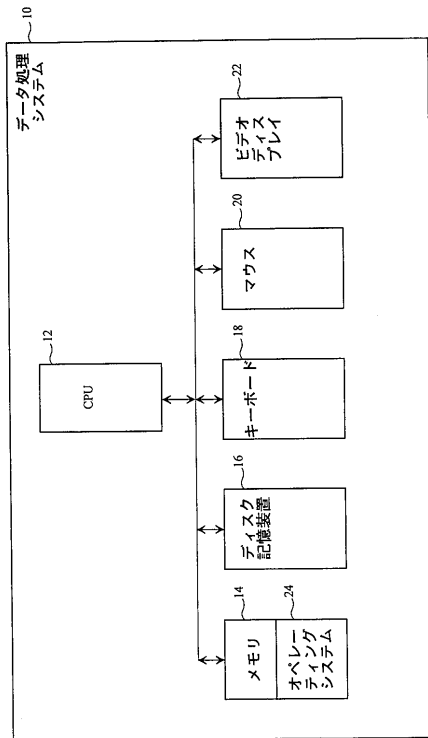
34 ルートノード

36、38、40、42、44、46 ノード

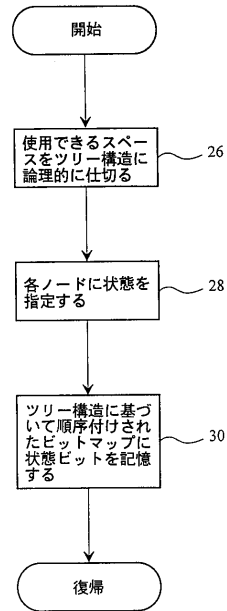
47 リンク

48 ビットマップ

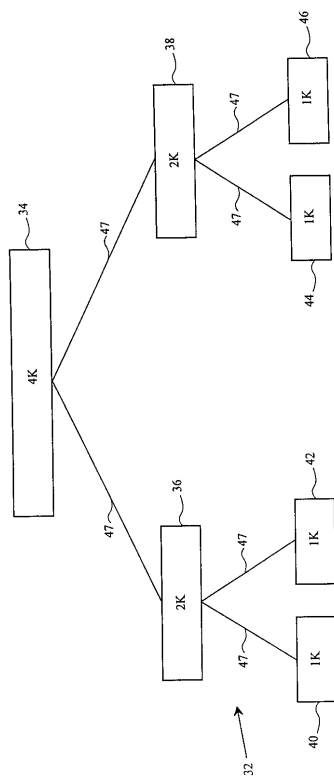
【図 1】



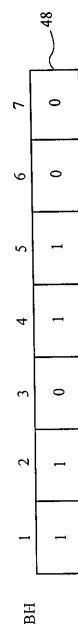
【図 2】



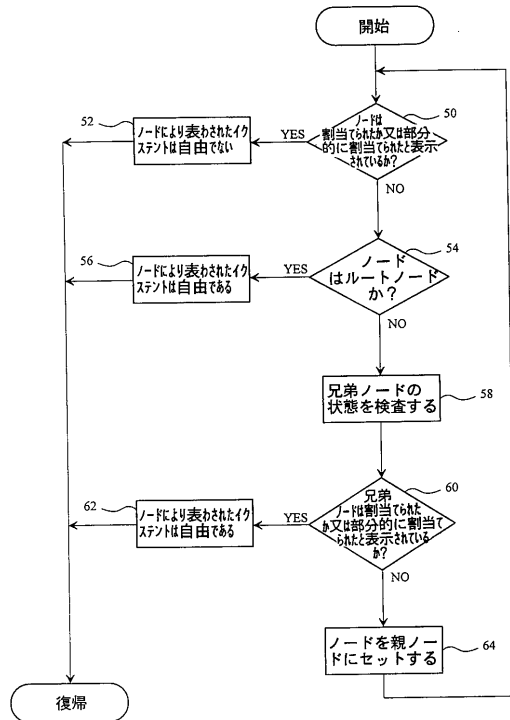
【図 3】



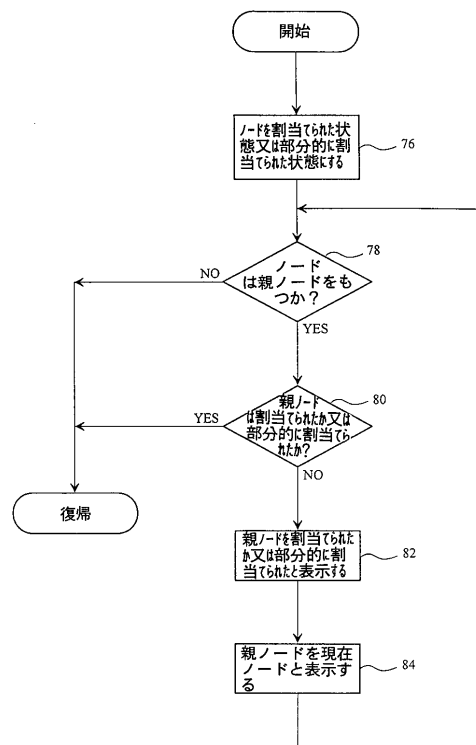
【図 4】



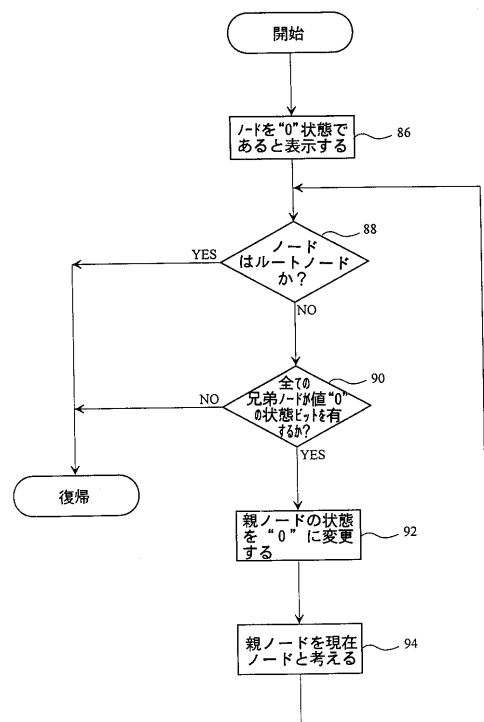
【図5】



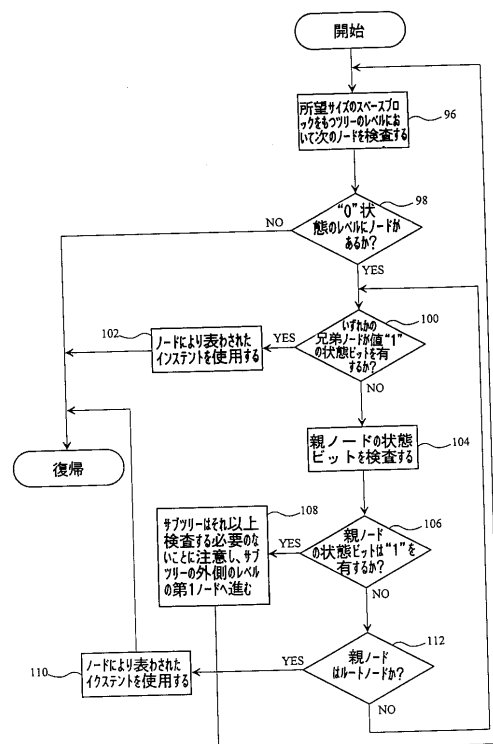
【図6】



【図7】



【図8】



フロントページの続き

- (72)発明者 マーク ズビコウスキ
アメリカ合衆国 ワシントン州 98072 ウッディンヴィル ノース イースト ワンハンド
レッドアンドセヴンティエイス ブレイス 15817
- (72)発明者 ロバート アイ ファーガソン
アメリカ合衆国 ワシントン州 98119 シアトル ナインス アベニュー ウェスト 29
10

審査官 多賀 実

- (56)参考文献 特開平03 - 092941 (JP, A)
特開平05 - 046447 (JP, A)
特開平02 - 071342 (JP, A)
特開平02 - 096213 (JP, A)
特開昭61 - 253530 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F3/06, 12/00-02