

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2014-206884

(P2014-206884A)

(43) 公開日 平成26年10月30日(2014. 10. 30)

(51) Int.Cl.

G 0 6 F 12/00 (2006.01)

F I

G 0 6 F 12/00 5 0 1 H

テーマコード (参考)

審査請求 未請求 請求項の数 5 O L (全 24 頁)

(21) 出願番号 特願2013-84488 (P2013-84488)
 (22) 出願日 平成25年4月15日 (2013. 4. 15)

(71) 出願人 508089266
 株式会社フィックスターズ
 東京都品川区大崎1-11-1 ゲートシ
 ティ大崎ウエストタワー18階
 (74) 代理人 110000752
 特許業務法人朝日特許事務所
 (72) 発明者 三木 聡
 東京都品川区大崎1-11-1 ゲートシ
 ティ大崎ウエストタワー18階 株式会社
 フィックスターズ内

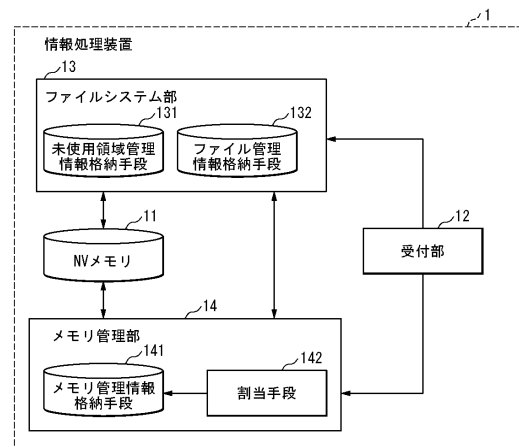
(54) 【発明の名称】 情報処理装置、情報処理方法、およびプログラム

(57) 【要約】 (修正有)

【課題】 NVメモリをメインメモリおよびストレージとして使用する場合において、メモリ割り当てに掛かる処理時間を短縮することができる情報処理装置、情報処理方法、およびプログラムを提供する。

【解決手段】 不揮発性の記録媒体であるNVメモリ11と、NVメモリに保存されている1以上のファイルを管理するファイルシステム部13と、実行中のプロセスからの要求に応じて、ファイルシステム部がファイルとして使用し得るNVメモリの領域をプロセスに割り当てるメモリ管理部14とを備え、ファイルシステム部は、前記NVメモリの未使用領域を管理するための情報であり、NVメモリの特性に合致するデータ構造を有する情報である未使用領域管理情報が格納されている前記NVメモリの領域にアクセスできる情報処理装置1。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

不揮発性の記録媒体である N V メモリと、
前記 N V メモリに保存されている 1 以上のファイルを管理するファイルシステム部と、
実行中のプロセスからの要求に応じて、前記ファイルシステム部がファイルとして使用し
得る前記 N V メモリの領域を確保し、当該確保した領域を当該プロセスに割り当てるメモ
リ管理部とを備え、

前記ファイルシステム部は、

前記 N V メモリの未使用領域を管理するための情報であり、N V メモリの特性に合致する
データ構造を有する情報である未使用領域管理情報が格納されている前記 N V メモリの領
域にアクセスでき、

前記 N V メモリの特性に合致するデータ構造は、

物理的に連続していない N V メモリの領域を、論理的に連続している領域として使用する
ためのデータ構造である情報処理装置。

【請求項 2】

前記データ構造は、リスト構造およびそれを基にしたメモリ管理構造である請求項 1 記載
の情報処理装置。

【請求項 3】

前記ファイルシステム部は、

実行中のプロセスからの要求に応じて、当該プロセスが要求したサイズと同サイズのファ
イルを、前記 N V メモリの未使用領域に書き込み、

前記メモリ管理部は、

実行中のプロセスからの要求に応じて、前記ファイルシステム部がファイルを書き込んだ
N V メモリの未使用領域を、当該プロセスに割り当てる請求項 1 または請求項 2 記載の情
報処理装置。

【請求項 4】

不揮発性の記録媒体である N V メモリと、

ファイルシステム部と、メモリ管理部とを用いて行われる情報処理方法であって、

前記ファイルシステム部が、

前記 N V メモリに保存されている 1 以上のファイルを管理するファイルシステムステップ
と、

前記メモリ管理部が、

実行中のプロセスからの要求に応じて、前記ファイルシステム部がファイルとして使用し
得る前記 N V メモリの領域を確保し、当該確保した領域を当該プロセスに割り当てるメモ
リ管理ステップとを備え、

前記ファイルシステムステップでは、

前記ファイルシステム部が、

前記 N V メモリの未使用領域を管理するための情報であり、N V メモリの特性に合致する
データ構造を有する情報である未使用領域管理情報が格納されている前記 N V メモリの領
域にアクセスでき、

前記 N V メモリの特性に合致するデータ構造は、

物理的に連続していない N V メモリの領域を、論理的に連続している領域として使用する
ためのデータ構造である情報処理方法。

【請求項 5】

不揮発性の記録媒体である N V メモリにアクセス可能なコンピュータを、

前記 N V メモリに保存されている 1 以上のファイルを管理するファイルシステム部と、

実行中のプロセスからの要求に応じて、前記ファイルシステム部がファイルとして使用し
得る前記 N V メモリの領域を確保し、当該確保した領域を当該プロセスに割り当てるメモ
リ管理部として機能させるためのプログラムであって、

前記ファイルシステム部は、

10

20

30

40

50

前記 N V メモリの未使用領域を管理するための情報であり、N V メモリの特性に合致するデータ構造を有する情報である未使用領域管理情報が格納されている前記 N V メモリの領域にアクセスでき、

前記 N V メモリの特性に合致するデータ構造は、

物理的に連続していない N V メモリの領域を、論理的に連続している領域として使用するためのデータ構造であるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、不揮発性メモリを使用する情報処理装置等に関するものである。

10

【背景技術】

【0002】

従来、SSD（非特許文献1参照）や、EEPROM（非特許文献2参照）などの不揮発性メモリ（Non-Volatile Memory、以下、適宜、N V メモリ）が開発されている。

【先行技術文献】

【非特許文献】

【0003】

【非特許文献1】“SSDとは”、[online]、IT用語辞典 e-Words、[2013年3月19日検索]、インターネット[URL ; <http://e-words.jp/w/SSD.html>]

20

【非特許文献2】“EEPROMとは”、[online]、IT用語辞典 e-Words、[2013年3月19日検索]、インターネット[URL ; <http://e-words.jp/w/EEPROM.html>]

【発明の概要】

【発明が解決しようとする課題】

【0004】

近年、N V メモリをメインメモリとして用いる研究、およびメインメモリをストレージとして用いる研究等により、N V メモリをメインメモリとして使用する領域とストレージとして使用する領域とを区別することなく、N V メモリをメインメモリおよびストレージとして使用することが可能となっている。N V メモリを、メインメモリおよびストレージの両方として使用することができるということは、その両方を1つに融合できることを意味する。また、当該融合により、システムの処理性能向上につながるという利点がある。しかしながら、ファイルシステムのブロックを、メインメモリとしてプロセスに割り当てる処理に、時間が掛かっていた。

30

【課題を解決するための手段】

【0005】

本第一の発明の情報処理装置は、不揮発性の記録媒体であるN V メモリと、N V メモリに保存されている1以上のファイルを管理するファイルシステム部と、実行中のプロセスからの要求に応じて、ファイルシステム部がファイルとして使用し得るN V メモリの領域を確保し、確保した領域をプロセスに割り当てるメモリ管理部とを備え、ファイルシステム部は、N V メモリの未使用領域を管理するための情報であり、N V メモリの特性に合致するデータ構造を有する情報である未使用領域管理情報が格納されているN V メモリの領域にアクセスでき、N V メモリの特性に合致するデータ構造は、物理的に連続していないN V メモリの領域を、論理的に連続している領域として使用するためのデータ構造である情報処理装置である。

40

【0006】

このような構成により、N V メモリをメインメモリおよびストレージとして使用する場合において、メモリ割り当てに掛かる処理時間を短縮することができる。

【0007】

また、本第二の発明の情報処理装置は、第一の発明に対して、データ構造は、リスト構造およびそれを基にしたメモリ管理構造である情報処理装置である。

50

【 0 0 0 8 】

このような構成により、N Vメモリをメインメモリおよびストレージとして使用する場
合において、メモリ割り当てに掛かる処理時間を短縮することができる。

【 0 0 0 9 】

また、本第三の発明の情報処理装置は、第一または第二の発明に対して、ファイルシ
ステム部は、実行中のプロセスからの要求に応じて、プロセスが要求したサイズと同サイ
ズのファイルを、N Vメモリの未使用領域に書き込み、メモリ管理部は、実行中のプロセ
スからの要求に応じて、ファイルシステム部がファイルを書き込んだN Vメモリの未使用
領域を、プロセスに割り当てる情報処理装置である。

【 0 0 1 0 】

このような構成により、ファイルシステムを介してN Vメモリの領域をプロセスに割
り当てる場合において、当該メモリ割り当てに掛かる処理時間を短縮することができる。

【発明の効果】

【 0 0 1 1 】

本発明による情報処理装置等によれば、N Vメモリをメインメモリおよびストレージと
して使用する場
合において、メモリ割り当てに掛かる処理時間を短縮することができる。

【図面の簡単な説明】

【 0 0 1 2 】

【図 1】実施の形態 1 における情報処理装置 1 のブロック図

【図 2】同未使用領域管理情報のデータ構造の例を示す図

【図 3】同情報処理装置 1 の全体動作について説明するフローチャート

【図 4】同領域の割り当てと解放処理について説明するフローチャート

【図 5】同ファイル管理情報の例を示す図

【図 6】同未使用領域管理情報の変更の様子を示す図

【図 7】同ファイル管理情報の例を示す図

【図 8】同メモリ管理情報の例を示す図

【図 9】同未使用領域管理情報の変更の様子を示す図

【図 1 0】同システムの構成例を示す図

【図 1 1】同閾値の変更例を示す図

【図 1 2】同メモリ割当コストの比較の実験結果を示す図

【図 1 3】同メモリ割当コストの比較の実験結果を示す図

【図 1 4】同実行命令数、キャッシュミス回数の計測結果を示す図

【図 1 5】上記実施の形態におけるコンピュータシステムの概観図

【図 1 6】上記実施の形態におけるコンピュータシステムのブロック図

【発明を実施するための形態】

【 0 0 1 3 】

以下、本発明による情報処理装置等の実施形態について図面を参照して説明する。なお
、実施の形態において同じ符号を付した構成要素は同様の動作を行うので、再度の説明を
省略する場合がある。また、本実施の形態において説明する各情報の形式、内容などは、
あくまで例示であり、各情報の持つ意味を示すことができれば、形式、内容などは問わな
い。

【 0 0 1 4 】

(実施の形態 1)

本実施の形態において、メインメモリとして使用する領域とストレージとして使用する
領域とを区別することなく、N Vメモリをメインメモリおよびストレージとして使用する
情報処理装置 1 であって、メインメモリとして使用する領域をファイルシステムにて管理
する情報処理装置 1 について説明する。なお、ここで、ストレージとは、いわゆる 2 次記
憶装置や、補助記憶装置と呼ばれる記憶装置のことである。また、メインメモリとは、い
わゆる 1 次記憶装置や、主記憶装置と呼ばれる記憶装置のことである。また、「メインメ
モリとして使用する領域」は、「実行中のプロセスが使用する領域」と言い換えてもよい

10

20

30

40

50

。

【 0 0 1 5 】

図 1 は、本実施の形態における情報処理装置 1 のブロック図である。情報処理装置 1 は、N V メモリ 1 1、受付部 1 2、ファイルシステム部 1 3、メモリ管理部 1 4 を備える。また、ファイルシステム部 1 3 は、未使用領域管理情報格納手段 1 3 1、ファイル管理情報格納手段 1 3 2 を備える。また、メモリ管理部 1 4 は、メモリ管理情報格納手段 1 4 1、割当手段 1 4 2 を備える。

【 0 0 1 6 】

また、本実施の形態における情報処理装置 1 が備えるファイルシステム部 1 3、およびメモリ管理部 1 4 は、例えば、いわゆる O S や、当該 O S のカーネルと呼ばれるものなどである。

10

【 0 0 1 7 】

N V メモリ 1 1 は、不揮発性の記録媒体である。N V メモリ 1 1 は、例えば、P C M (P a h s e C h a n g e M e m o r y、相変化メモリ)や、M R A M、R e M A M、E E P R O M、フラッシュメモリなどである。また、N V メモリ 1 1 は、通常、不揮発性メモリであれば、その形状や種類などは、問わない。また、N V メモリ 1 1 は、例えば、バイト単位アクセスが可能な不揮発性メモリであることが好適である。

【 0 0 1 8 】

また、N V メモリ 1 1 は、通常、1 以上の領域を有する。当該「領域」は、例えば、「ブロック」や、「ページ」などと呼ばれる領域である。また、当該領域には、通常、プロセスが使用するデータが格納される。当該プロセスは、例えば、ジョブや、タスク、スレッドなどと呼ばれる場合もある。また、当該データは、例えば、ファイルや、ファイルの一部、変数に代入された値などである。また、領域のサイズ(容量)は、通常、予め決められている。当該サイズは、例えば、4 K B (キロバイト)や、5 1 2 B (バイト)などである。また、当該一の領域は、例えば、後述のファイルシステム部 1 3 からは、「ブロック」と呼ばれる。また、当該一の領域は、例えば、後述のメモリ管理部 1 4 からは、「ページ」と呼ばれる。また、本実施の形態において、当該ブロックのサイズと、当該ページのサイズとは、同一であり、当該サイズは、4 K B であることが好適である。

20

【 0 0 1 9 】

なお、N V メモリ 1 1 が有する 1 以上の各領域を、以下、適宜、「N V メモリ 1 1 の領域」、または「領域」とする。また、データが書き込まれていない領域や、データの書き込みが許可されている領域などを、以下、適宜、未使用領域とする。また、データが書き込まれている領域や、データの書き込みが許可されていない領域などを、以下、適宜、使用領域とする。

30

【 0 0 2 0 】

また、N V メモリ 1 1 は、通常、一の物理アドレス空間に配置される。当該物理アドレス空間は、通常、連続した 1 以上の物理アドレスを有する。また、「物理アドレス空間に配置される」とは、N V メモリ 1 1 が有する 1 以上の各領域と、一の物理アドレスとが対応付けられることである。つまり、当該一の領域は、一の物理アドレスにより識別される。また、当該一の領域は、例えば、論理アドレスや仮想アドレスなどにより間接的に識別されてもよい。

40

【 0 0 2 1 】

また、N V メモリ 1 1 が有する 1 以上の各領域は、領域識別子により識別される。領域識別子とは、当該 1 以上の各領域を識別する情報である。領域識別子は、例えば、物理アドレスや、論理アドレス、仮想アドレスなどである。つまり、領域識別子は、当該 1 以上の各領域を、直接的または間接的に識別する情報であればよい。また、領域識別子は、通常、順序を有する。

【 0 0 2 2 】

また、上記、N V メモリ 1 1 の物理アドレス空間への配置は、通常、後述のファイルシステム部 1 3 が行う。また、当該配置は、例えば、後述のメモリ管理部 1 4 が行ってもよ

50

い。また、当該配置は、例えば、図示しないアドレス空間配置部が行ってもよい。

【0023】

受付部12は、指示や情報などを受け付ける。指示は、例えば、プログラムの実行の指示や、プロセスの実行の指示、プロセスへの領域の割当の指示、ファイル操作の指示などである。また、情報は、例えば、いわゆるコマンドや、当該コマンドに与える引数などである。また、これらの指示や情報は、通常、指示や情報を出力するプログラムが出力する。

【0024】

ファイルシステム部13は、いわゆるファイルシステムである。当該ファイルシステムは、通常、ディスクファイルシステム（ローカルファイルシステム）である。また、当該ファイルシステムは、通常、XIP（eXecute-In-Place）をサポートしており、NVメモリ11を対象としたファイルシステムであれば、何でもよい。当該ファイルシステムは、例えば、PRAMFS（Persistent and Protected RAM File System）や、Ext2（Second Extended Filesystem）、SCMFS（A File System for Storage Class Memory）などである。また、ファイルシステム部13は、NVメモリ11を、いわゆるストレージとして使用する。また、ファイルシステム部13は、通常、NVメモリ11を1以上の領域に分割し、当該一の領域を、ブロックとして使用する。

【0025】

また、当該ファイルシステムは、通常、NVメモリ11の特性に合致するデータ構造を有する情報（以下、適宜、未使用領域管理情報）により未使用領域を管理することのできるファイルシステムであることが好適である。当該「NVメモリ11の特性」とは、通常、ランダムアクセス（読み書き）が高速であることである。また、「NVメモリ11の特性」とは、例えば、HDD（ハード・ディスク・ドライブ）が備えるヘッドを備えていないこと、また、これに伴い、シークが不要であることなどであってもよい。また、当該「NVメモリ11の特性に合致するデータ構造」とは、NVメモリが有する2以上の領域について、物理的に連続していない2以上の領域を、論理的に連続している2以上の領域として使用するためのデータ構造である。

【0026】

ここで、「物理的に連続している2以上の領域」とは、当該2以上の各領域を識別する2以上の領域識別子が連続している領域のことである。また、「論理的に連続している2以上の領域」とは、当該2以上の各領域を識別する2以上の領域識別子は連続していないが、連続してアクセスすることが可能な領域のことである。

【0027】

また、「NVメモリ11の特性に合致するデータ構造」は、例えば、リスト構造や、木構造、これらのデータ構造を基にしたメモリ管理構造などである。「これらのデータ構造を基にしたメモリ管理構造」とは、通常、バディ・システムや、バディブロック・アロケータ、バディ・アルゴリズムなどと呼ばれるメモリ管理技術において使用されるデータ構造である。なお、「NVメモリ11の特性に合致するデータ構造」は、通常、片方向の線形リスト（Singly-Linked List）であることが好適である。また、当該データ構造には、通常、いわゆるビットマップ構造は含まれない。

【0028】

また、「未使用領域を管理する」とは、NVメモリ11のどの領域が未使用領域であるかを、予め決められたデータ構造を有する未使用領域管理情報により記録することである。当該「予め決められたデータ構造」とは、通常、上記「NVメモリ11の特性に合致するデータ構造」である。また、当該「記録する」とは、通常、未使用領域管理情報格納手段131に格納されている未使用領域管理情報を、変更することや、未使用領域管理情報を、未使用領域管理情報格納手段131に蓄積することなどである。当該「変更」は、通常、「書き換え」や、「更新」などを含み、広く解する。

【 0 0 2 9 】

ここで、未使用領域管理情報とは、未使用領域を管理するための情報である。未使用領域管理情報は、通常、少なくとも、未使用領域を識別する1以上の領域識別子を有する。当該1以上の領域識別子は、通常、1以上の先頭領域識別子であることが好適である。先頭領域識別子とは、連続未使用領域が有する先頭の一の未使用領域を識別する領域識別子である。また、連続未使用領域とは、物理的に連続している1以上の未使用領域の集合である。また、2以上の連続未使用領域は、通常、物理的に連続していない。また、未使用領域管理情報が有する1以上の領域識別子は、例えば、1以上の各未使用領域を識別する1以上の領域識別子であってもよい。この場合、当該1以上の領域識別子は、通常、予め決められた順序にソートされている。当該「予め決められた順序」とは、通常、昇順である。

10

【 0 0 3 0 】

また、未使用領域管理情報は、例えば、未使用領域のサイズを示す情報（以下、適宜、サイズ情報）を有していてもよい。当該サイズ情報は、通常、1以上の各連続未使用領域のサイズを示す1以上のサイズ情報であることが好適である。また、当該サイズ情報は、例えば、1以上の各未使用領域のサイズを示す1以上のサイズ情報であってもよい。

【 0 0 3 1 】

また、ファイルシステム部13は、通常、N Vメモリ11にファイルを書き込んだ場合や、N Vメモリ11からファイルを削除した場合などに、未使用領域管理情報を変更する。

20

【 0 0 3 2 】

（1）一のファイルをN Vメモリ11に書き込む場合：この場合、ファイルシステム部13は、通常、ファイルの書き込み先となる1以上の各未使用領域を識別する1以上の領域識別子を取得する。例えば、ファイルシステム部13は、当該書き込むファイルのサイズに合う1以上の領域識別子を、未使用領域管理情報を用いて取得する。「サイズに合う」とは、通常、当該サイズが収まることである。つまり、当該1以上の各領域識別子により識別される1以上の未使用領域のサイズは、通常、書き込むファイルのサイズ以上のサイズであり、当該書き込むファイルのサイズとの差が最小であるサイズである。また、当該取得する1以上の領域識別子は、連続していることが好適である。

【 0 0 3 3 】

例えば、未使用領域管理情報が、先頭領域識別子と、当該先頭領域識別子により識別される未使用領域を含む連続未使用領域のサイズを示すサイズ情報とが対応付いた情報であるとする。この場合、ファイルシステム部13は、例えば、書き込むファイルのサイズ以上のサイズを示すサイズ情報に対応付いている領域識別子を、未使用領域管理情報から取得する。また、ファイルシステム部13は、書き込むファイルのサイズに応じて、当該取得した領域識別子により識別される未使用領域に連続する1以上の各未使用領域を識別する1以上の領域識別子を取得する。

30

【 0 0 3 4 】

例えば、先頭領域識別子が「0 x F 0 0 0 0 0 0 1」であり、サイズ情報が「8 0 K B」であるとする。また、一の領域のサイズが「4 K B」であるとする。また、書き込むファイルのサイズが「1 0 K B」であるとする。このような場合、取得すべき領域識別子は、3つである。従って、ファイルシステム部13は、まず、当該先頭領域識別子「0 x F 0 0 0 0 0 0 1」を取得する。そして、ファイルシステム部13は、当該先頭領域識別子に連続する2つの領域識別子「0 x F 0 0 0 0 0 0 2」、「0 x F 0 0 0 0 0 0 3」を作成し、取得する。

40

【 0 0 3 5 】

（2）一のファイルをN Vメモリ11に書き込んだ場合：この場合、ファイルが書き込まれた1以上の各領域は、使用領域となる。従って、この場合、ファイルシステム部13は、通常、未使用領域管理情報により識別される1以上の未使用領域に、ファイルが書き込まれた1以上の領域が含まれないように、未使用領域管理情報を変更する。

50

【 0 0 3 6 】

例えば、未使用領域管理情報が、上記（１）の場合と同様に、先頭領域識別子と、当該先頭領域識別子により識別される未使用領域を含む連続未使用領域のサイズを示すサイズ情報とが対応付いた情報であるとする。この場合、当該未使用領域管理情報が有する先頭領域識別子とサイズ情報とにより、領域識別子の範囲が示される。従って、ファイルシステム部１３は、上記（１）の場合において１以上の領域識別子を取得する際に用いた未使用領域管理情報が有する先頭領域識別子を、当該取得した１以上の領域識別子が領域識別子の範囲に含まれないように変更する。

【 0 0 3 7 】

例えば、先頭領域識別子が「０×Ｆ００００００１」であり、サイズ情報が「８０ＫＢ」であるとする。また、一の領域のサイズが「４ＫＢ」であるとする。この様な場合、当該先頭領域識別子とサイズ情報とにより、領域識別子の範囲「０×Ｆ００００００１～０×Ｆ００００００２０」が示される。また、取得した１以上の領域識別子が「０×Ｆ００００００１」、「０×Ｆ００００００２」、「０×Ｆ００００００３」であるとする。この様な場合、当該領域識別子の範囲に、当該取得した３つの領域識別子が含まれる。従って、この様な場合、ファイルシステム部１３は、当該３つの領域識別子が、当該領域識別子の範囲に含まれないように、当該先頭領域識別子を「０×Ｆ００００００４」に変更する。また、この様な場合、ファイルシステム部１３は、当該取得した３つの領域識別子に相当するサイズ「１２ＫＢ」より、当該サイズ情報を「６８ＫＢ」に変更する。

【 0 0 3 8 】

（３）一のファイルをＮＶメモリ１１から削除した場合：この場合、ファイルが削除された１以上の各領域は、未使用領域となる。従って、この場合、ファイルシステム部１３は、通常、未使用領域管理情報により識別される１以上の未使用領域に、ファイルが削除された１以上の領域が含まれるように、未使用領域管理情報を変更する。

【 0 0 3 9 】

例えば、未使用領域管理情報が、上記（１）の場合と同様に、先頭領域識別子と、当該先頭領域識別子により識別される未使用領域を含む連続未使用領域のサイズを示すサイズ情報とが対応付いた情報であるとする。この場合、当該未使用領域管理情報が有する先頭領域識別子とサイズ情報とにより、領域識別子の範囲が示される。従って、ファイルシステム部１３は、未使用領域管理情報が有する先頭領域識別子を、当該取得した１以上の領域識別子が領域識別子の範囲に含まれるように変更する。なお、この場合、当該先頭領域識別子は、当該取得した１以上の領域識別子に連続する領域識別子であることが好適である。

【 0 0 4 0 】

例えば、先頭領域識別子が「０×Ｆ００００００４」であり、サイズ情報が「６８ＫＢ」であるとする。また、一の領域のサイズが「４ＫＢ」であるとする。この様な場合、当該先頭領域識別子とサイズ情報とにより、領域識別子の範囲「０×Ｆ００００００４～０×Ｆ００００００２０」が示される。また、取得した１以上の領域識別子が「０×Ｆ００００００１」、「０×Ｆ００００００２」、「０×Ｆ００００００３」であるとする。この様な場合、当該領域識別子の範囲に、当該取得した３つの領域識別子が含まれない。従って、この様な場合、ファイルシステム部１３は、当該３つの領域識別子が、当該領域識別子の範囲に含まれるように、当該先頭領域識別子を「０×Ｆ００００００１」に変更する。また、この様な場合、ファイルシステム部１３は、当該取得した３つの領域識別子に相当するサイズ「１２ＫＢ」より、当該サイズ情報を「８０ＫＢ」に変更する。

【 0 0 4 1 】

なお、ファイルシステム部１３は、ＮＶメモリ１１へのファイルの書き込みや、ＮＶメモリ１１からのファイルの削除などに応じて、当該書き込み後や当該削除後のＮＶメモリ１１の未使用領域の状況を示すように未使用領域管理情報を変更することができればよく、その方法や手順などは、問わない。

【 0 0 4 2 】

また、ファイルシステム部 13 は、ファイルシステムであるので、通常、N V メモリ 11 に保存されている 1 以上のファイルを管理する。「ファイルを管理する」とは、例えば、書き込みを要求されたファイルを N V メモリ 11 に書き込むこと、読み出しを要求されたファイルを N V メモリ 11 から読み出すこと、削除を要求されたファイルを N V メモリ 11 から削除すること、当該書き込みに応じて、当該書き込んだファイルを管理するための情報（以下、適宜、ファイル管理情報）をファイル管理情報格納手段 132 に蓄積すること、当該削除に応じて、当該削除したファイルに対応するファイル管理情報をファイル管理情報格納手段 132 から削除すること、ファイルが書き込まれていない領域を識別する領域識別子をファイル管理情報格納手段 132 に蓄積すること、ファイルが書き込まれている領域の識別子を要求に応じてファイル管理情報格納手段 132 から取得することなどである。また、ファイルを削除することとは、例えば、N V メモリ 11 からはファイルを削除せず、当該ファイルに対応するファイル管理情報を削除することや、当該ファイル管理情報にファイルが削除されたことを示すフラグを付与することなどであってもよい。

10

【0043】

なお、ファイルシステム部 13 は、ファイルシステムであり、O S の機能である。従って、「ファイルを管理する」とは、ファイルシステムが行う処理や、ファイルシステムの機能のすべてを含むと考えてよい。また、ファイルシステムには、いわゆる仮想ファイルシステムを含むと考えてもよい。

【0044】

ここで、ファイル管理情報とは、ファイルを管理するための情報である。ファイル管理情報は、通常、少なくとも、ファイル識別子と、領域識別子とを有する。また、一のファイル識別子は、通常、1 以上の領域識別子と対応付けている。ファイル識別子は、例えば、任意の文字列や、任意の数値、いわゆるファイル名などである。また、ファイル識別子は、ファイルを識別することができれば、その形式や内容などは、問わない。

20

【0045】

また、ファイル管理情報は、例えば、ファイルが作成された日時や、ファイルが更新された日時、ファイルのアクセス権（パーミッション）、ファイルを作成したユーザなどのファイルの属性を示す情報を有していてもよい。また、これらの情報は、通常、一のファイル識別子に対応付けている。

【0046】

（１）一のファイルを N V メモリ 11 に書き込む場合：この場合、ファイルシステム部 13 は、例えば、まず、当該一のファイルを、どの 1 以上の領域に書き込むのかを決定する。当該「決定する」とは、通常、ファイルを書き込む 1 以上の各未使用領域を識別する 1 以上の領域識別子を取得することである。また、当該 1 以上の領域識別子の取得は、通常、上記のとおり、未使用領域管理情報を用いて行う。そして、ファイルシステム部 13 は、当該一のファイルを、当該決定した領域ごとの 1 以上のデータに分割する。そして、ファイルシステム部 13 は、当該分割した 1 以上のデータを、当該決定した 1 以上の各領域に書き込む。

30

【0047】

（２）一のファイルを N V メモリ 11 に書き込んだ場合：この場合、ファイルシステム部 13 は、通常、書き込んだファイルを識別するファイル識別子と、書き込み先の 1 以上の各領域を識別する 1 以上の領域識別子との対応を示すファイル管理情報を、ファイル管理情報格納手段 132 に蓄積する。当該蓄積は、通常、ファイル管理情報格納手段 132 への追記である。

40

【0048】

（３）一のファイルを N V メモリ 11 から読み出す場合：この場合、ファイルシステム部 13 は、まず、当該一のファイルが書き込まれている 1 以上の領域を識別する領域識別子を、ファイル管理情報から取得する。そして、ファイルシステム部 13 は、取得した 1 以上の領域識別子が示す領域から、1 以上に分割されたファイルのデータを取得する。

【0049】

50

(4)一のファイルをN Vメモリ11から削除する場合：この場合、ファイルシステム部13は、まず、当該一のファイルが書き込まれている1以上の領域を識別するファイル識別子を、ファイル管理情報から取得する。そして、ファイルシステム部13は、取得した1以上の領域識別子が示す領域から、データを消去する。なお、当該「消去」には、通常、当該ファイル管理情報を削除することにも含まれる。

【0050】

また、この場合、ファイルシステム部13は、例えば、当該一のファイルを識別するファイル識別子を有するファイル管理情報を、ファイル管理情報格納手段132から削除することのみを行ってもよい。また、当該ファイル管理情報の削除は、例えば、当該ファイル管理情報に削除したことを示すフラグを付与することであってもよい。

10

【0051】

(5)一のファイルをN Vメモリ11から削除した場合：この場合、ファイルシステム部13は、通常、削除したファイルを識別するファイル識別子を有するファイル管理情報を、ファイル管理情報格納手段132から削除する。

【0052】

なお、ファイルシステムは、公知の技術であるので、ファイルシステム部13が行う処理や動作については、適宜、説明を省略する。

【0053】

未使用領域管理情報格納手段131には、未使用領域管理情報が格納される。未使用領域管理情報の詳細については、説明済みであるので、説明を省略する。

20

【0054】

また、未使用領域管理情報格納手段131は、通常、N Vメモリ11が有する1以上の領域で実現される。従って、未使用領域管理情報は、通常、N Vメモリ11に保存されている。例えば、未使用領域管理情報が片方向の線形リストのデータ構造を有する場合、当該未使用領域管理情報は、N Vメモリ11に、例えば、図2に示す様に保存される。図2において、各連続未使用領域における先頭の領域、およびその次の領域には、それぞれ、連続未使用領域のサイズと、次の連続未使用領域へのポインタ(次の連続未使用領域における先頭領域識別子)が保存されている。

【0055】

ファイル管理情報格納手段132には、1以上のファイル管理情報が格納される。当該1以上のファイル管理情報の集合を、以下、適宜、ファイル管理情報群とする。また、ファイル管理情報の詳細については、説明済みであるので、説明を省略する。

30

【0056】

メモリ管理部14は、いわゆるメモリ管理を行う。メモリ管理とは、例えば、実行中のプロセスからの要求に応じて、N Vメモリ11が有する1以上の領域を確保し、当該確保した領域を当該プロセスに割り当てることや、プロセスが終了した場合に、当該プロセスに割り当てられていた1以上の領域を、未使用領域として解放することなどである。

【0057】

「1以上の領域を確保する」とは、当該1以上の各領域を識別する1以上の領域識別子を取得することである。

40

【0058】

また、実行中のプロセスは、通常、1つの仮想アドレス空間を持つ。従って、「1以上の領域を割り当てる」とは、通常、当該仮想アドレス空間が有する1以上の各仮想アドレスと、1以上の各領域識別子とを対応付けることや、当該1以上の領域識別子をプロセスに渡すことなどである。当該領域識別子は、通常、物理アドレスである。また、このとき、メモリ管理部14は、プロセスを識別するプロセス識別子に対応付けてもよい。また、このとき、メモリ管理部14は、当該対応付けた結果を示す情報(以下、適宜、メモリ管理情報)を、メモリ管理情報格納手段141に蓄積する。また、当該領域の割り当てを、以下、適宜、領域割当、メモリ割当、ブロック割当、ページ割当などとする。

【0059】

50

また、「１以上の領域を解放する」とは、通常、仮想アドレスと領域識別子との対応付けを解除することである。この場合、メモリ管理部１４は、通常、当該対応付けを示すメモリ管理情報を、メモリ管理情報格納手段１４１から削除する。

【００６０】

ここで、メモリ管理情報とは、プロセスに割り当てたＮＶメモリ１１の領域を管理するための情報である。メモリ管理情報は、通常、少なくとも、１以上の仮想アドレスと、１以上の物理アドレスとを有する。また、当該１以上の各仮想アドレスと、１以上の各物理アドレスとは、通常、１対１で対応付けられている。また、当該対応付いた情報には、プロセスを識別するプロセス識別子が対応付けられていてもよい。プロセス識別子は、例えば、プロセス名や、プロセスの番号などである。また、プロセス識別子は、プロセスを識別することができれば、その形式や内容などは、問わない。

10

【００６１】

また、メモリ管理部１４が確保する１以上の領域は、通常、ファイルシステム部１３がファイルとして使用し得る１以上の領域である。「ファイルとして使用し得る１以上の領域」とは、ファイルが書き込まれる可能性のある１以上の領域や、ファイルが書き込まれている１以上の領域、書き込まれているデータがファイルとして扱われる可能性のある１以上の領域などである。つまり、当該１以上の領域は、１以上の未使用領域であってもよいし、１以上の使用領域であってもよい。また、メモリ管理部１４は、通常、ファイルシステム部１３を介して１以上の領域の確保を行う。「ファイルシステム部１３を介して」とは、ファイルシステム部１３が１以上の領域識別子を取得することにより、１以上の領域を確保することである。

20

【００６２】

なお、メモリ管理は、公知の技術であるので、メモリ管理部１４が行う処理や動作については、適宜、説明を省略する。また、メモリ管理部１４は、例えば、１以上の領域の確保や、当該確保した１以上の領域のプロセスへの割り当て、割り当てた１以上の領域の解放などを、通常、後述の各手段により行う。

【００６３】

メモリ管理情報格納手段１４１には、メモリ管理情報が格納される。メモリ管理情報の詳細については、説明済みであるので、説明を省略する。

【００６４】

30

割当手段１４２は、実行中のプロセスからの要求に応じて、１以上の領域を当該プロセスに割り当てる。また、割当手段１４２は、当該割り当ての結果を示すメモリ管理情報を、メモリ管理情報格納手段１４１に蓄積する。

【００６５】

例えば、実行中のプロセスからメモリ割当ての要求があった場合、割当手段１４２は、通常、ファイルシステム部１３から、ファイルが書き込まれた領域を識別する１以上の領域識別子を受信する。そして、割当手段１４２は、当該１以上の各領域識別子により識別される１以上の領域を、割り当てを要求したプロセスに割り当てる。当該割り当ての方法を、以下、適宜、ｍｍａｐ方式とする。

【００６６】

40

また、上記において、割当手段１４２は、例えば、当該実行中のプロセスからの要求に応じて、当該プロセスが要求した領域のサイズと同サイズのファイルをＮＶメモリ１１に書き込むことを、ファイルシステム部１３に指示してもよい。当該「ファイルを書き込む」ことは、ファイルを作成することである。そして、割当手段１４２は、ファイルシステム部１３から、ファイルが書き込まれた領域を識別する１以上の領域識別子を受信する。

【００６７】

なお、受付部１２における情報や指示などの入力手段は、メニュー画面によるものや、キーボードなど、何でもよい。受付部１２は、メニュー画面の制御ソフトウェアや、キーボード等の入力手段のデバイスドライバなどで実現され得る。

【００６８】

50

また、ファイルシステム部 1 3、メモリ管理部 1 4、割当手段 1 4 2 は、通常、M P U やメモリ等から実現され得る。また、ファイルシステム部 1 3 などの処理手順は、通常、ソフトウェアで実現され、当該ソフトウェアは R O M 等の記録媒体に記録されている。なお、ファイルシステム部 1 3 などは、ハードウェア（専用回路）で実現されてもよい。

【 0 0 6 9 】

また、未使用領域管理情報格納手段 1 3 1、ファイル管理情報格納手段 1 3 2、メモリ管理情報格納手段 1 4 1 は、通常、N V メモリ 1 1 が有する 1 以上の領域で実現される。また、未使用領域管理情報格納手段 1 3 1 などは、例えば、N V メモリ 1 1 以外の不揮発性の記録媒体や、揮発性の記録媒体で実現されてもよい。また、未使用領域管理情報格納手段 1 3 1 などに所定の情報が記憶される過程は、問わない。例えば、当該所定の情報は、記録媒体や、通信回線、入力デバイスなどを介して未使用領域管理情報格納手段 1 3 1 などに記憶されてもよい。

10

【 0 0 7 0 】

次に、情報処理装置 1 の全体動作について、フローチャートを用いて説明する。図 3 は、情報処理装置 1 の全体動作を示すフローチャートである。

【 0 0 7 1 】

（ステップ S 3 0 1）ファイルシステム部 1 3 は、N V メモリ 1 1 を物理アドレス空間に配置する。当該配置は、メモリ管理部 1 4 が行ってもよい。

【 0 0 7 2 】

（ステップ S 3 0 2）ファイルシステム部 1 3 は、受付部 1 2 がファイル書込指示を受け付けたか否かを判断する。ファイル書込指示とは、ファイルを N V メモリ 1 1 に書き込むための指示である。また、ファイル書込指示は、通常、ファイル識別子と、ファイルを構成するデータとを有する。そして、受け付けた場合は、ステップ S 3 0 3 に進み、そうでない場合は、ステップ S 3 0 7 に進む。

20

【 0 0 7 3 】

（ステップ S 3 0 3）ファイルシステム部 1 3 は、ファイルを書き込むための 1 以上の領域を確保する。具体的に、ファイルシステム部 1 3 は、未使用領域管理情報格納手段 1 3 1 に格納されている未使用領域管理情報を用いて、1 以上の領域識別子を取得する。

【 0 0 7 4 】

（ステップ S 3 0 4）ファイルシステム部 1 3 は、ステップ S 3 0 3 で取得した 1 以上の各領域識別子により識別される 1 以上の領域に、ファイルを書き込む。

30

【 0 0 7 5 】

（ステップ S 3 0 5）ファイルシステム部 1 3 は、ステップ S 3 0 3 で取得した 1 以上の各領域識別子により識別される 1 以上の領域が、未使用領域管理情報により識別される 1 以上の未使用領域に含まれないように、未使用領域管理情報を変更する。

【 0 0 7 6 】

（ステップ S 3 0 6）ファイルシステム部 1 3 は、ステップ S 3 0 3 で取得した 1 以上の領域識別子と、ステップ S 3 0 4 で書き込んだファイルを識別するファイル識別子とを対応付け、ファイル管理情報としてファイル管理情報格納手段 1 3 2 に蓄積する。

【 0 0 7 7 】

40

（ステップ S 3 0 7）ファイルシステム部 1 3 は、受付部 1 2 がファイル読出指示を受け付けたか否かを判断する。ファイル読出指示とは、ファイルを N V メモリ 1 1 から読み出すための指示である。また、ファイル読出指示は、通常、ファイル識別子を有する。そして、受け付けた場合は、ステップ S 3 0 8 に進み、そうでない場合は、ステップ S 3 1 0 に進む。

【 0 0 7 8 】

（ステップ S 3 0 8）ファイルシステム部 1 3 は、読み出すファイルが書き込まれている 1 以上の各領域を識別する 1 以上の領域識別子を、ファイル管理情報格納手段 1 3 2 から取得する。

【 0 0 7 9 】

50

(ステップS309) ファイルシステム部13は、ステップS308で取得した1以上の各領域識別子により識別される1以上の領域から、ファイルを読み出す。

【0080】

(ステップS310) ファイルシステム部13は、受付部12がファイル削除指示を受け付けたか否かを判断する。ファイル削除指示とは、ファイルをNVメモリ11から削除するための指示である。また、ファイル削除指示は、通常、ファイル識別子を有する。そして、受け付けた場合は、ステップS311に進み、そうでない場合は、ステップS315に進む。

【0081】

(ステップS311) ファイルシステム部13は、削除するファイルが書き込まれている1以上の各領域を識別する1以上の領域識別子を、ファイル管理情報格納手段132から取得する。

【0082】

(ステップS312) ファイルシステム部13は、ステップS311で取得した1以上の各領域識別子により識別される1以上の領域から、ファイルを削除する。

【0083】

(ステップS313) ファイルシステム部13は、ステップS311で取得した1以上の各領域識別子により識別される1以上の領域が、未使用領域管理情報により識別される1以上の未使用領域に含まれるように、未使用領域管理情報を変更する。

【0084】

(ステップS314) ファイルシステム部13は、ステップS311で取得した1以上の領域識別子を有するファイル管理情報を、ファイル管理情報格納手段132から削除する。

【0085】

(ステップS315) メモリ管理部14は、受付部12がプロセス実行指示を受け付けたか否かを判断する。プロセス実行指示とは、プロセスを実行するための指示である。また、プロセス実行指示は、通常、プロセス識別子を有する。そして、受け付けた場合は、ステップS316に進み、そうでない場合は、ステップS302に戻る。

【0086】

(ステップS316) 情報処理装置1は、プロセスを実行する。

【0087】

(ステップS317) メモリ管理部14は、実行中のプロセスへの1以上の領域の割り当て、および実行中のプロセスからの1以上の領域の解放を行う。この処理の詳細は、図4のフローチャートを用いて説明する。そして、ステップS302に戻る。

【0088】

なお、図3のフローチャートにおいて、電源オフや処理終了の割り込みにより処理を終了してもよい。

【0089】

図4は、図3のフローチャートのステップS317の領域の割り当てと解放処理を示すフローチャートである。

【0090】

(ステップS401) 割当手段142は、プロセスから要求されたサイズと同サイズのファイルを書き込むためのファイル書込指示を、ファイルシステム部13に送信する。

【0091】

(ステップS402) 割当手段142は、ファイルシステム部13から、ファイルが書き込まれた領域を識別する1以上の領域識別子を受信する。

【0092】

(ステップS403) 割当手段142は、ステップS402で受信した1以上の領域識別子を、領域の割り当てを要求したプロセスに割り当てる。具体的に、割当手段142は、通常、当該割り当ての結果を示すメモリ管理情報を、メモリ管理情報格納手段141に

10

20

30

40

50

蓄積する。

【0093】

(ステップS404)メモリ管理部14は、プロセスの実行が終了したか否かを判断する。終了した場合は、ステップS405に進み、そうでない場合は、ステップS404に戻る。

【0094】

(ステップS405)割当手段142は、プロセスに割り当てていた1以上の各領域を識別する1以上の領域識別子を、メモリ管理情報格納手段141から取得する。

【0095】

(ステップS406)割当手段142は、ステップS405で取得した1以上の各領域識別子により識別される1以上の領域に書き込まれているファイルを削除するためのファイル削除指示を、ファイルシステム部13に送信する。

10

【0096】

(ステップS407)割当手段142は、ステップS405で取得した1以上の領域識別子を、領域の割り当てを要求したプロセスから解放する。具体的に、割当手段142は、通常、ステップS405で取得した1以上の領域識別子を有するメモリ管理情報を、メモリ管理情報格納手段141から削除する。そして、上位処理にリターンする。

【0097】

(具体例)

次に、情報処理装置1の動作の具体例について説明する。なお、本具体例において、NVメモリ11は、一の物理アドレス空間に配置されているものとする。また、NVメモリ11は、サイズが4KBである1以上の領域を有しているものとする。

20

【0098】

(例1)

本例において、mmap方式による領域割当の例について説明する。なお、本例において、未使用領域管理情報格納手段131には、図2に示す様に、未使用領域管理情報が格納されているものとする。また、ファイル管理情報格納手段132には、図5に示すファイル管理情報が格納されているものとする。当該ファイル管理情報は、レコードを一意に特定するためのIDと、ファイル識別子(項目名:ファイル)と、物理アドレスとを有する。当該物理アドレスは、領域識別子である。また、図5のファイル管理情報は、例えば、物理アドレス「0x000000001」、「0x000000002」、「0x000000003」のそれぞれにより識別される3つの領域に、ファイル識別子「file01」により識別されるファイルが書き込まれていることを示す。

30

【0099】

まず、ユーザが、情報処理装置1を操作し、任意のプログラムを実行する操作を行ったとする。すると、受付部12は、当該プログラムをプロセスとして実行するためのプロセス実行指示を受け付ける。当該プロセス実行指示は、プロセス識別子「proc01」を有するものとする。そして、情報処理装置1は、当該プロセス識別子により識別される上記プログラムを、プロセスとして実行する。

【0100】

次に、実行中のプロセスが、10KBの領域を割り当てることを要求したとする。すると、受付部12は、領域を割り当てるための指示である領域割当指示を受け付ける。当該領域割当指示は、プロセス識別子「proc01」と、割り当てる領域のサイズ「10KB」を有しているものとする。

40

【0101】

次に、割当手段142は、サイズ「10KB」のファイルをNVメモリ11に書き込むためのファイル書込指示を、ファイルシステム部13に送信する。当該ファイル書込指示は、ファイルのサイズ「10KB」を有しているものとする。

【0102】

次に、受付部12は、上記のファイル書込指示を受け付ける。そして、ファイルシステ

50

ム部 1 3 は、当該ファイル書込指示に従い、任意のファイル識別子を取得する。また、ファイルシステム部 1 3 は、未使用領域管理情報格納手段 1 3 1 に格納されている未使用領域管理情報を用いて、ファイルのサイズ「1 0 K B」に合う 3 つの未使用領域を識別する物理アドレスを取得する。この結果、ファイルシステム部 1 3 は、ファイル識別子「f i l e 9 9」を取得したものとする。また、ファイルシステム部 1 3 は、「0 x 0 0 0 0 0 0 0 4」、「0 x 0 0 0 0 0 0 0 5」、「0 x 0 0 0 0 0 0 0 6」の物理アドレスを取得したものとする。

【0 1 0 3】

次に、ファイルシステム部 1 3 は、取得した 3 つの各物理アドレスにより識別される 3 つの領域に、ファイルを書き込む。そして、ファイルシステム部 1 3 は、未使用領域管理情報格納手段 1 3 1 に格納されている未使用領域管理情報により識別される 1 以上の未使用領域に、当該ファイルを書き込んだ 3 つの領域が含まれないように、当該未使用領域管理情報を変更する。当該未使用領域管理情報の変更の様子は、例えば、図 6 である。また、ファイルシステム部 1 3 は、当該書き込みの結果を示すファイル管理情報を、ファイル管理情報格納手段 1 3 2 に蓄積し、図 5 のファイル管理情報群に追記する。当該追記後のファイル管理情報は、図 7 である。当該ファイル管理情報が有する情報や、当該ファイル管理情報が示す意味などは、図 5 と同様であるので、説明を省略する。

10

【0 1 0 4】

次に、ファイルシステム部 1 3 は、ファイルを書き込んだ領域を識別する 3 つの物理アドレスを、メモリ管理部 1 4 に送信する。

20

【0 1 0 5】

次に、割当手段 1 4 2 は、3 つの物理アドレス「0 x 0 0 0 0 0 0 0 4」、「0 x 0 0 0 0 0 0 0 5」、「0 x 0 0 0 0 0 0 0 6」を受信する。そして、割当手段 1 4 2 は、当該 3 つの物理アドレスにより識別される 3 つの領域を、プロセス識別子「p r o c 0 1」により識別されるプロセスに割り当てる。そして、割当手段 1 4 2 は、当該割り当ての結果を示すメモリ管理情報を、メモリ管理情報格納手段 1 4 1 に蓄積する。当該蓄積したメモリ管理情報は、図 8 である。

【0 1 0 6】

次に、プロセス識別子「p r o c 0 1」により識別されるプロセスの実行が終了したとする。すると、割当手段 1 4 2 は、図 8 のメモリ管理情報から、プロセス識別子「p r o c 0 1」に対応する物理アドレス「0 x 0 0 0 0 0 0 0 4」、「0 x 0 0 0 0 0 0 0 5」、「0 x 0 0 0 0 0 0 0 6」を取得する。

30

【0 1 0 7】

次に、割当手段 1 4 2 は、上記で取得した 3 つの物理アドレスにより識別される領域に保存されているファイルを削除するためのファイル削除指示を、ファイルシステム部 1 3 に送信する。当該ファイル削除指示は、3 つの物理アドレス「0 x 0 0 0 0 0 0 0 4」、「0 x 0 0 0 0 0 0 0 5」、「0 x 0 0 0 0 0 0 0 6」を有しているものとする。

【0 1 0 8】

次に、受付部 1 2 は、上記のファイル削除指示を受け付ける。そして、ファイルシステム部 1 3 は、当該ファイル削除指示に従い、当該ファイル削除指示が有する 3 つの物理アドレス「0 x 0 0 0 0 0 0 0 4」、「0 x 0 0 0 0 0 0 0 5」、「0 x 0 0 0 0 0 0 0 6」に保存されているファイルを削除する。そして、ファイルシステム部 1 3 は、未使用領域管理情報格納手段 1 3 1 に格納されている未使用領域管理情報により識別される 1 以上の未使用領域に、当該ファイルを削除した 3 つの領域が含まれるように、当該未使用領域管理情報を変更する。当該未使用領域管理情報の変更の様子は、例えば、図 9 である。また、ファイルシステム部 1 3 は、当該削除したファイルが保存されていた 3 つの各物理アドレスを有するファイル管理情報を、図 7 のファイル管理情報群から削除する。当該削除後のファイル管理情報は、図 5 である。

40

【0 1 0 9】

次に、割当手段 1 4 2 は、図 8 のメモリ管理情報を、メモリ管理情報格納手段 1 4 1 か

50

ら削除する。

【0110】

（例2）

本例において、情報処理装置1の具体的な実現方法の例について説明する。なお、本例において、NVメモリ11は、単に「NVメモリ」と表記するものとする。また、情報処理装置1は、Linux（登録商標）カーネル（以下、カーネル）であるものとする。

【0111】

まず、情報処理装置1を構成するシステムの構成例について説明する。当該構成例を示す図は、図10である。図10は、NVメモリとファイルシステムが融合されたシステムを示す概念図である。NVメモリが有する領域を、ストレージとして使用する領域、メインメモリとして使用する領域とに区別することなく、NVメモリをストレージおよびメインメモリの両方に使用できるようにするために、情報処理装置1は、NVメモリが有する領域を、ファイルシステムにより管理する。NVメモリ上のファイルシステムにファイルを格納する場合は、通常と同様に、ファイルのデータを格納するためのブロックが割り当てられる。ファイルに格納されたプログラムテキストおよびデータは、XIPにより、メインメモリとして直接参照できる。それ以外に、空きメモリ領域を必要とする場合は、ファイルシステムから空きブロックを取得する。図10のプロセスは、ファイルのテキストをXIPによりマップし、ヒープに必要な領域は空きブロックを取得しマップしている。なお、空きメモリ領域、空きブロックは、共に、未使用領域のことである。

【0112】

次に、mmap方式の実現方法の例について説明する。mmap方式は、ファイルシステムからブロックを確保し、プロセスへのメモリ割当に使用するために、mmapシステムコール（以下、適宜、「mmap」）を用いる方式である。従って、mmap方式を実現するために、カーネル内への新規の実装は、不要である。mmap方式によるメモリ割当を行うには、プログラムでメモリ割当を行う際に、mmapシステムコールを用いるようにする。しかしながら、既存のプログラムの多くでは、メモリ割当には「malloc（）」が用いられている。Linux（登録商標）で標準的に用いられている標準Cライブラリ「GNU libc」では、小さなメモリ割当にはヒープ領域が、大きなメモリ割当には「mmap」により確保された領域が割り当てられるようになっている。この割り当てる領域が小さいか大きいかを判断するための閾値を、図11の様に変更することで、プログラムを修正せずに、常に「mmap」が用いられるようになる。

【0113】

（例3）

本例において、例2に示した実現方法により実現した情報処理装置1を動作させた実験結果などについて説明する。

【0114】

まず、メモリ割当コストの比較の実験結果について説明する。

【0115】

（実験環境）

NVメモリをメインメモリとして持つシステムは、一般的に入手可能な状態ではない。また、アクセス遅延の差、書き込み回数の制限は考慮しないと、DRAMの一部をNVメモリとみなして実験を行った。実験には、「Intel Atom D2700 2.13GHz」を用いた。ハイパースレッディング機能はBIOSで無効化し、1CPUの状態に計測した。DRAMの一部をNVメモリとみなすため、8GBのDRAMを搭載し、当該DRAMの後半の4GBの領域をNVメモリとして使用することにした。カーネルが認識するDRAM領域を制限するために、カーネルに「mem=128M」オプションを渡し、DRAMとしては128MBだけ使用する状態で実験を行った。なお、DRAMのみの場合の実験は、上記のメインメモリの分割を行わない状態で行った。

【0116】

（実験結果）

メモリ割当コストを計測するため、引数で指定された容量のメモリ領域を確保し、その領域の各ページの先頭部分に対し書き込みを行うプログラムを作成し、実行した。測定結果には、メモリ割当コストのみが含まれ、解放コストは含まれない。実行したのは、DRAMのみ、直接融合方式、半直接融合方式、間接融合方式（事前確保およびオンデマンド確保）、mmap方式の5つの方式の場合である。DRAMのみの場合を除き、Ext2とPRAMFSの2つのファイルシステムで実行した。

【0117】

ここで、直接融合方式とは、ファイルシステムを介することなく、プロセスが要求するサイズのNVメモリの領域を、直接、プロセスに割り当てる方法である。また、半直接融合方式とは、プロセスが要求するサイズのNVメモリの領域を、ファイルシステムが管理している任意のファイルに追加し、当該追加した領域を、プロセスに割り当てる方法である。また、事前確保の間接融合方式とは、ファイルシステムが管理しているすべてのファイルが格納されているNVメモリの領域を、事前にすべて確保し、当該確保した領域を、プロセスに割り当てる方法である。また、オンデマンド確保の間接融合方式とは、ファイルシステムが管理しているファイルが格納されているNVメモリの領域を、プロセスからの要求があるたびに確保し、当該確保した領域を、プロセスに割り当てる方法である。なお、これら4つの方式の詳細については、説明を省略する。

10

【0118】

また、上記の実行結果は、図12である。図12において、横軸は割当メモリサイズ(MB)、縦軸はRDTSC(Read Time Stamp Counter)により取得した値をクロック値で割ることで得た実行時間(ミリ秒)である。

20

【0119】

また、図12の実行結果から、以下の3つのことがわかる。

- (1) Ext2における半直接融合方式のコストが大きい
- (2) mmap方式のコストはExt2およびPRAMFSの両方で大きい
- (3) 最もコストが低いのは間接融合方式（事前確保）である

【0120】

また、図12において、間接融合方式（事前確保）と比較して、直接融合方式、半直接融合方式、間接融合方式（オンデマンド確保）、mmap方式の各方式は、PRAMFSでは、それぞれ2.3%、2.5%、1.2%、27.9%、コストが大きい。一方、Ext2では、それぞれ5.2%、111.5%、3.2%、73.6%、コストが大きい。間接融合方式（オンデマンド確保）では、より大きなメモリ割当が発生するたびに、ファイルへのブロック割当が発生する。しかしながら、一度割り当てられたブロックは、解放後はフリーリストで管理され再利用されるため、間接融合方式（オンデマンド確保）では、割当コストへの影響は小さい。半直接融合方式は、PRAMFSではコストへの影響は小さく済んでいる一方で、Ext2のコストは大きくなっている。半直接融合方式のExt2での実装は、余分なブロック確保と解放を伴う、冗長なものとなっているため、その影響が大きいと考えられる。mmap方式と他方式との最も際立った違いは、ファイルシステムからのブロック確保の方式にある。間接融合方式では、基本的にはフリーリストから1ブロック取得するだけである。一方、直接融合方式では、フリーリストが空になるたびに、ファイルシステムから複数ブロックを一括確保する。mmap方式では、ページ割当要求に対し、1ブロックずつファイルシステムから確保する必要がある。従って、実験結果から、ファイルシステムからのブロック確保コストの影響を受けること、一括確保の効果が大きいことがわかる。

30

40

【0121】

次に、ファイルシステムにおける空き領域管理方式の影響についての検証結果について説明する。

【0122】

図12の実験結果より、ファイルシステムからのブロック確保コストの影響が大きいことがわかる。特に、mmap方式では、1ブロックずつ確保するため、ファイルシステム

50

の空き領域管理方式の影響を大きく受ける。そこで、単純な内部構造を持つ P R A M F S を対象とし、ファイルシステムにおける空き領域管理方式を、メモリ上の操作に適した構造とすることで、ブロック確保コストを軽減し、その影響を検証する。

【 0 1 2 3 】

ファイルシステムから 1 ブロックの確保に必要な操作は、直接融合方式および間接融合方式では、基本的にはフリーリストから要素を 1 つ取り出すだけである。一方、m m a p 方式では、ファイルシステムの空き領域管理方式を通してブロック確保を行う。P R A M F S は、空き領域管理にビットマップを採用しているため、ビットマップの検索コストがかかる。シークに時間のかかる H D D では、ファイル内部のフラグメントの原因となるため、空きブロックをリストで管理する方式は、通常、使用されない。一方、ビットマップは、あるブロックに近い別の空きブロックを見つけやすく、内部フラグメントを抑制できるため、多くのファイルシステムで用いられている。しかしながら、シーク時間を考慮する必要のない N V メモリ上にファイルシステムを構築する場合、検索に時間のかかるビットマップの利点はない。

10

【 0 1 2 4 】

また、ファイルシステムの空き領域管理方式の軽量化が、m m a p 方式でのメモリ割当コストの軽減につながることを検証するため、P R A M F S の空き領域管理にリストを用いるように変更した。それにより、間接融合方式と同じコストで、ファイルシステムから 1 ブロックの確保が可能になる。また、ビットマップデータを置く必要がなくなるため、ファイルシステムの管理領域が小さくなり、その分をデータ領域に使用することができるようになる。

20

【 0 1 2 5 】

また、上記と同様に、各方式でメモリ割当を行うプログラムを実行した。当該実行結果は、図 1 3 である。図 1 3 において、空き領域管理方式にリストを用いるように変更した結果は、「P R A M F S 2」として示されている。空き領域管理方式の軽量化により、m m a p 方式によるメモリ割当コストは 5 8 . 2 % 減少している。一方、他方式のコストへの影響は、ほとんど見られない。

【 0 1 2 6 】

また、図 1 3 において、m m a p 方式のメモリ割当コストが大きく減少したことで、間接融合方式（事前確保）と比較しても 4 6 . 8 % 低く、m m a p 方式は、最もコストの低い方式となった。このような大きな減少の理由を調査するために、P e r f o r m a n c e M o n i t o r i n g C o u n t e r (P M C) を用いて、実行命令数、キャッシュミスの回数を計測した。また、同じ実験を Q E M U でも行い、実行命令数を計測した。当該結果は、図 1 4 である。図 1 4 において、Q E M U による T S C (T i m e S t a m p C o u n t e r) の計測は、実行命令数を表すのに対し、A t o m による T S C の計測は、実行時間を表す。また、図 1 4 において、「I n s t r u c t i o n R e t i r e d」は、実行命令数、「L L C (L a s t L e v e l C a c h e) M i s s e s」は、すべての L 2 キャッシュミスの回数、「M E M L O A D R E T I R E D . L 2 H I T」は、L 1 データキャッシュをミスし L 2 キャッシュにヒットしたロード命令数、「M E M L O A D R E T I R E D . L 2 M I S S」は、L 2 キャッシュをミスしたロード命令数である。

30

40

【 0 1 2 7 】

図 1 4 において、Q E M U における実行命令数と、A t o m における実行命令数は、割り込みの影響があるため完全には一致しないが、近い値となっている。実行命令数では、空き領域管理方式の軽量化により、m m a p 方式のメモリ割当コストの減少は 3 7 . 5 % となり、実行時間ほどの大きな減少とはなっていない。実行時間の差を広げる原因となったと考えられるのが、キャッシュミスの回数である。図 1 4 において、「L L C M i s s e s」、「M E M L O A D R E T I R E D . L 2 M I S S」の両方で、大きな差が見られる。「A t o m D 2 7 0 0」の L 2 キャッシュは、コアあたり 5 1 2 K B であり、現在のノート PC 用プロセッサと比較しても小さい。従って、空き領域管理方式の軽

50

量化により、アクセスするデータの多くが、Atomのキャッシュにのる効果があったと考えられる。実際に、8MBのL3キャッシュを持つ「Xeon E3-1270」で実験を行ったところ、メモリ割当コストの減少は26.0%となり、キャッシュサイズが影響していることを裏付けている。

【0128】

以上の実験から、NVメモリ上に構築するファイルシステムは、HDDとの性質の違いから、異なった内部構造を持つべき場合があり、空き領域管理方式はその1つであることがわかる。

【0129】

以上、本実施の形態による情報処理装置1によれば、NVメモリをメインメモリおよびストレージとして使用する場合において、メモリ割り当てに掛かる処理時間を短縮することができる。

10

【0130】

また、上記各実施の形態において、各処理または各機能は、単一の装置または単一のシステムによって集中処理されることによって実現されてもよいし、あるいは、複数の装置または複数のシステムによって分散処理されることによって実現されてもよい。

【0131】

また、上記各実施の形態において、各構成要素は専用のハードウェアにより構成されてもよいし、あるいは、ソフトウェアにより実現可能な構成要素については、プログラムを実行することによって実現されてもよい。例えば、ハードディスクや半導体メモリ等の記録媒体に記録されたソフトウェア・プログラムをCPU等のプログラム実行部が読み出して実行することによって、各構成要素が実現され得る。

20

【0132】

また、上記各実施の形態における情報処理装置を実現するソフトウェアは、例えば、以下のようなプログラムである。つまり、このプログラムは、不揮発性の記録媒体であるNVメモリにアクセス可能なコンピュータを、前記NVメモリに保存されている1以上のファイルを管理するファイルシステム部と、実行中のプロセスからの要求に応じて、前記ファイルシステム部がファイルとして使用し得る前記NVメモリの領域を確保し、当該確保した領域を当該プロセスに割り当てるメモリ管理部として機能させるためのプログラムであって、前記ファイルシステム部は、前記NVメモリの未使用領域を管理するための情報であり、NVメモリの特性に合致するデータ構造を有する情報である未使用領域管理情報が格納されている前記NVメモリの領域にアクセスでき、前記NVメモリの特性に合致するデータ構造は、物理的に連続していないNVメモリの領域を、論理的に連続している領域として使用するためのデータ構造であるプログラムである。

30

【0133】

なお、上記プログラムにおいて、上記プログラムが実現する機能には、ハードウェアでしか実現できない機能は含まれない。

【0134】

また、上記プログラムは、サーバなどからダウンロードされることによって実行されてもよいし、所定の記録媒体（例えば、CD-ROMなどの光ディスクや磁気ディスク、半導体メモリなど）に記録されたプログラムが読み出されることによって実行されてもよい。また、このプログラムは、プログラムプロダクトを構成するプログラムとして用いられ

40

【0135】

また、上記プログラムを実行するコンピュータは、単数であってもよいし、複数であってもよい。つまり、集中処理を行ってもよいし、あるいは分散処理を行ってもよい。

【0136】

また、図15は、前述のプログラムを実行して、前述の実施の形態の情報処理装置等を実現するコンピュータシステム9の概観図である。前述の実施の形態は、コンピュータハードウェア、およびその上で実行されるコンピュータプログラムで実現され得る。

50

【 0 1 3 7 】

図 1 5 において、コンピュータシステム 9 は、C D - R O M ドライブ 9 0 1 1、F D ドライブ 9 0 1 2 を含むコンピュータ 9 0 1 と、キーボード 9 0 2 と、マウス 9 0 3 と、モニタ 9 0 4 とを備える。

【 0 1 3 8 】

図 1 6 は、コンピュータシステム 9 のブロック図である。図 1 6 において、コンピュータ 9 0 1 は、C D - R O M ドライブ 9 0 1 1、F D ドライブ 9 0 1 2 に加えて、M P U 9 0 1 3 と、ブートアッププログラム等のプログラムを記憶するための R O M 9 0 1 4 と、M P U 9 0 1 3 に接続され、アプリケーションプログラムの命令を一時的に記憶するとともに一時記憶空間を提供するための R A M 9 0 1 5 と、アプリケーションプログラム、システムプログラム、およびデータを記憶するためのハードディスク 9 0 1 6 と、C D - R O M ドライブ 9 0 1 1、F D ドライブ 9 0 1 2、M P U 9 0 1 3 等を相互に接続するバス 9 0 1 7 とを備える。ここでは図示しないが、コンピュータ 9 0 1 は、さらに、L A N への接続を提供するネットワークカードを備えていてもよい。

10

【 0 1 3 9 】

コンピュータシステム 9 に、前述の実施の形態の情報処理装置等の機能を実行させるプログラムは、C D - R O M 9 1 0 1、または F D 9 1 0 2 に記憶されて、C D - R O M ドライブ 9 0 1 1 または F D ドライブ 9 0 1 2 に挿入され、さらにハードディスク 9 0 1 6 に転送されてもよい。これに代えて、プログラムは、図示しないネットワークを介してコンピュータ 9 0 1 に送信され、ハードディスク 9 0 1 6 に記憶されてもよい。プログラムは実行の際に R A M 9 0 1 5 にロードされる。プログラムは、C D - R O M 9 1 0 1、F D 9 1 0 2 またはネットワークから直接、ロードされてもよい。

20

【 0 1 4 0 】

プログラムは、コンピュータ 9 0 1 に、前述の実施の形態の情報処理装置等の機能を実行させるオペレーティングシステム (O S)、またはサードパーティプログラム等は、必ずしも含まなくてもよい。プログラムは、制御された態様で適切な機能 (モジュール) を呼び出し、所望の結果が得られるようにする命令の部分のみを含んでいればよい。コンピュータシステム 9 がどのように動作するかは周知であり、詳細な説明は省略する。

【 0 1 4 1 】

本発明は、以上の実施の形態に限定されることなく、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

30

【 産業上の利用可能性 】

【 0 1 4 2 】

以上のように、本発明にかかる情報処理装置は、N V メモリをメインメモリおよびストレージとして使用する場合において、メモリ割り当てに掛かる処理時間を短縮することができるという効果を有し、オペレーティングシステムの一機能等として有用である。

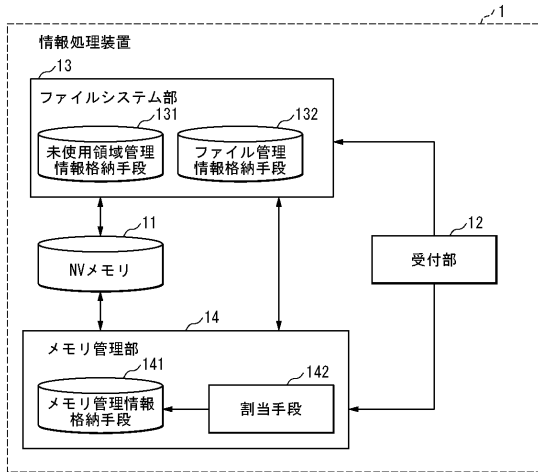
【 符号の説明 】

【 0 1 4 3 】

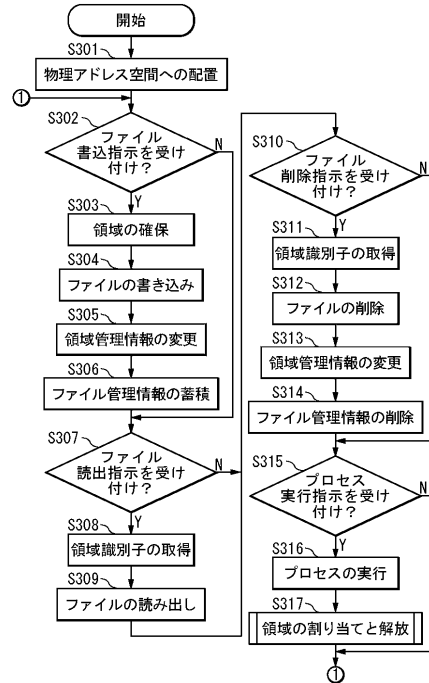
- 1 情報処理装置
- 1 1 N V メモリ
- 1 2 受付部
- 1 3 ファイルシステム部
- 1 4 メモリ管理部
- 1 3 1 未使用領域管理情報格納手段
- 1 3 2 ファイル管理情報格納手段
- 1 4 1 メモリ管理情報格納手段
- 1 4 2 割当手段

40

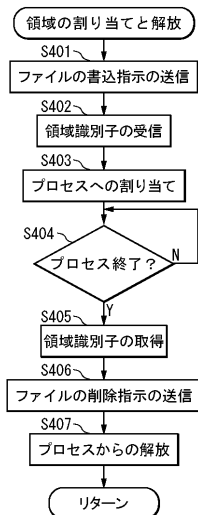
【図 1】



【図 3】



【図 4】



【図 5】

ID	ファイル	物理アドレス
001	file01	0x00000001
002	file01	0x00000002
003	file01	0x00000003
004	file03	0x00000011
005	file03	0x00000012
006	file03	0x00000013
007	file03	0x00000014

【図 7】

ID	ファイル	物理アドレス
001	file01	0x00000001
002	file01	0x00000002
003	file01	0x00000003
004	file03	0x00000011
005	file03	0x00000012
006	file03	0x00000013
007	file03	0x00000014
008	file99	0x00000004
009	file99	0x00000005
010	file99	0x00000006

【図 8】

ID	プロセス	仮想アドレス	物理アドレス
001	proc01	0xF0000001	0x00000004
002	proc01	0xF0000002	0x00000005
003	proc01	0xF0000003	0x00000006

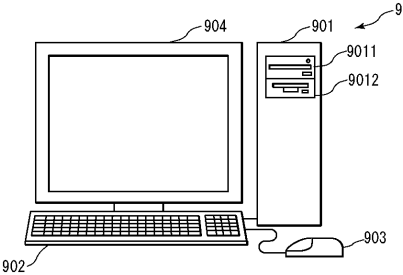
【図 11】

```
malloc(M_MMAP_THRESHOLD, 0);
```

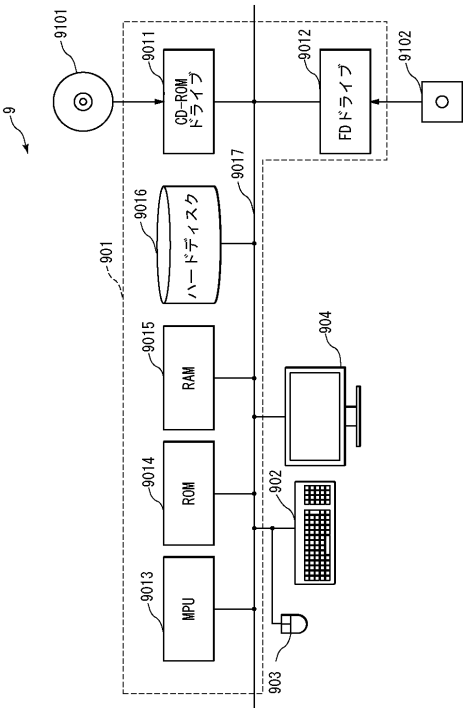
【図 1 4】

方式	OEMU		Atom D2700		LLC Misses
	TSC		TSC	Instruction Retired	
mmmap方式 (PRAMFS)	283,077,871		1,323,548,403	279,735,226	244,193
mmmap方式 (PRAMFS2)	174,381,505		542,393,113	174,759,261	158,849
間接融合方式 (prealloc)	194,853,076		1,011,510,282	195,495,299	376,673
DRAM	232,532,655		1,084,599,768	231,098,787	316,281
方式	Atom D2700		MEM LOAD RETIRED L2 HIT		MEM LOAD RETIRED L2 MISS
mmmap方式 (PRAMFS)			843,429		29,668
mmmap方式 (PRAMFS2)			842,063		8,843
間接融合方式 (prealloc)			191,909		59,815
DRAM			84,207		41,583

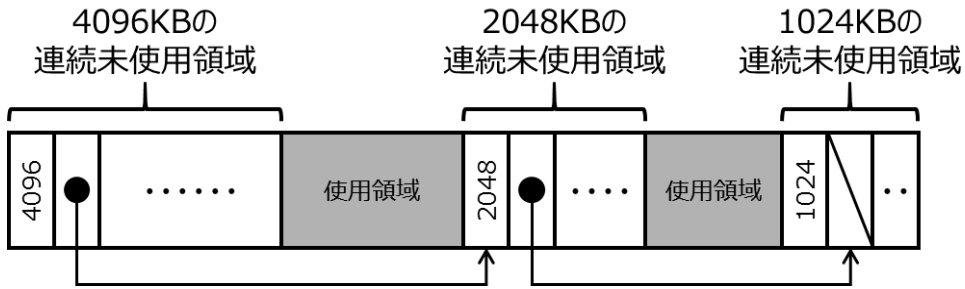
【図 1 5】



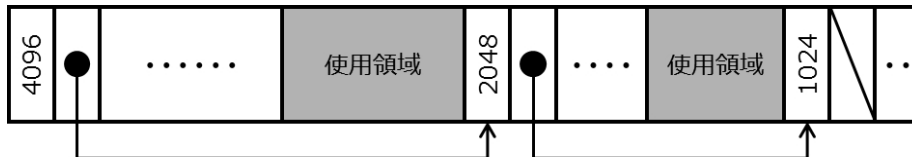
【図 1 6】



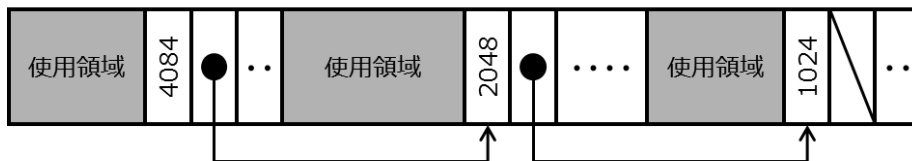
【図 2】



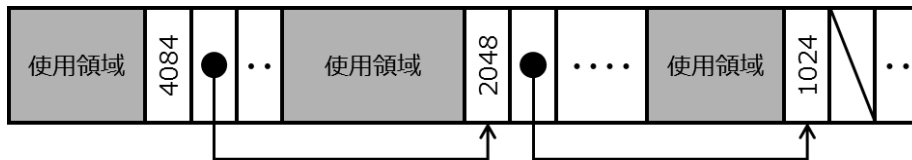
【図 6】



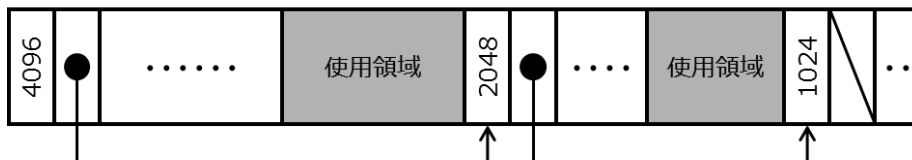
↓ 未使用領域管理情報の変更



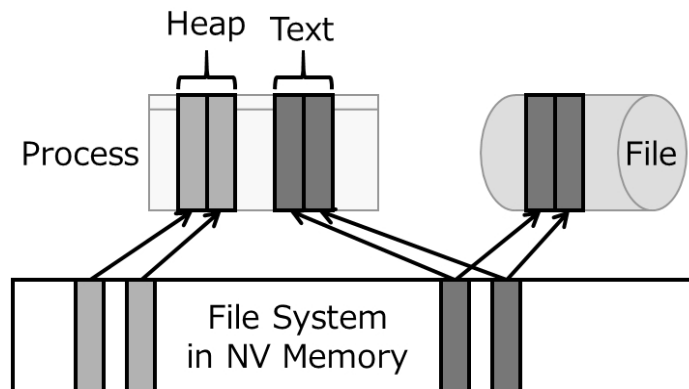
【図 9】



↓ 未使用領域管理情報の変更



【図 10】



[illegible]

Allocation size [MB]	PRAMFS (direct)	PRAMFS (semi-direct)	PRAMFS (indirect prealloc)	PRAMFS (indirect ondemand)	PRAMFS (mmap)	PRAMFS2 (direct)	PRAMFS2 (semi-direct)	PRAMFS2 (indirect prealloc)	PRAMFS2 (indirect ondemand)	PRAMFS2 (mmap)
1	0	0	0	0	0	0	0	0	0	0
32	30	30	30	30	30	30	30	30	30	30
64	60	60	60	60	60	60	60	60	60	60
128	120	120	120	120	120	120	120	120	120	120
256	240	240	240	240	240	240	240	240	240	240
512	490	490	480	480	260	610	490	480	260	260