

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2020-531951

(P2020-531951A)

(43) 公表日 令和2年11月5日(2020.11.5)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 12/109 (2016.01)</b>	G06F 12/109 1 O 5	5 B O 1 7
<b>G06F 12/1009 (2016.01)</b>	G06F 12/1009	5 B 2 O 5
<b>G06F 12/14 (2006.01)</b>	G06F 12/14 5 1 O D	

審査請求 未請求 予備審査請求 未請求 (全 40 頁)

(21) 出願番号	特願2020-506745 (P2020-506745)	(71) 出願人	511050697
(86) (22) 出願日	平成30年8月21日 (2018. 8. 21)		アリババ グループ ホウルディング リ
(85) 翻訳文提出日	令和2年3月31日 (2020. 3. 31)		ミテッド
(86) 国際出願番号	PCT/US2018/047201		英国領ケイマン諸島 グランド ケイマン
(87) 国際公開番号	W02019/040417		ジョージ タウン ビーオーボックス
(87) 国際公開日	平成31年2月28日 (2019. 2. 28)		8 4 7 ワン キャピタル プレイス フ
(31) 優先権主張番号	15/682, 437		ォース フロア
(32) 優先日	平成29年8月21日 (2017. 8. 21)	(74) 代理人	100079108
(33) 優先権主張国・地域又は機関	米国 (US)		弁理士 稲葉 良幸
		(74) 代理人	100109346
			弁理士 大貫 敏史
		(74) 代理人	100117189
			弁理士 江口 昭彦
		(74) 代理人	100134120
			弁理士 内藤 和彦

最終頁に続く

(54) 【発明の名称】 カーネル空間及びユーザ空間のメモリ管理のための方法及びシステム

## (57) 【要約】

本出願は、カーネル空間及びユーザ空間のメモリ管理のための方法及びシステムを提供する。カーネル空間及びユーザ空間のメモリ管理のための例示的システムは、カーネル空間に対応する第1のルートページテーブルインデックスを保存するように構成された第1の記憶ユニットを含み得る。このシステムは、ユーザ空間に対応する第2のルートページテーブルインデックスを保存するように構成された第2の記憶ユニットも含み得る。このシステムは、第1のレジスタ及び第2のレジスタと通信可能に結合され、並びにオペレーティングシステムカーネルのために、第1のルートページテーブルインデックスに従って第1の仮想アドレスを第1の物理アドレスに変換し、及びユーザプロセスのために、第2のルートページテーブルインデックスに従って第2の仮想アドレスを第2の物理アドレスに変換するように構成された制御ユニットをさらに含み得る。

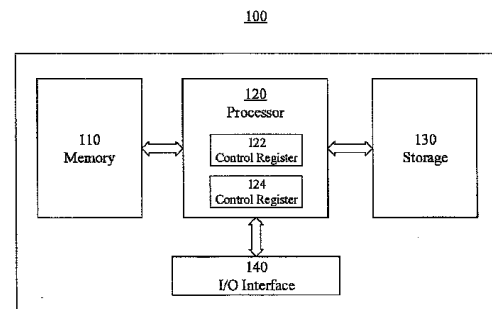


Fig. 1

**【特許請求の範囲】****【請求項 1】**

カーネル空間及びユーザ空間のメモリ管理のためのシステムであって、  
前記カーネル空間に対応する第 1 のルートページテーブルインデックスを保存するように構成された第 1 の記憶ユニットと、  
前記ユーザ空間に対応する第 2 のルートページテーブルインデックスを保存するように構成された第 2 の記憶ユニットと、  
前記第 1 の記憶ユニット及び前記第 2 の記憶ユニットと通信可能に結合され、並びにオペレーティングシステムカーネルのために、前記第 1 のルートページテーブルインデックスに従って第 1 の仮想アドレスを第 1 の物理アドレスに変換すること、及び  
ユーザプロセスのために、前記第 2 のルートページテーブルインデックスに従って第 2 の仮想アドレスを第 2 の物理アドレスに変換すること、  
を行うように構成された制御ユニットと、  
を含む、システム。

10

**【請求項 2】**

前記カーネル空間及び前記ユーザ空間が、同じ物理メモリ空間に対応し、前記同じ物理メモリ空間が、物理メモリ空間全体又は前記物理メモリ空間全体の一部である、請求項 1 に記載のシステム。

**【請求項 3】**

前記第 1 の仮想アドレスが、  
第 1 のページディレクトリインデックス、第 1 のページテーブルインデックス、及び第 1 のページオフセット、又は  
第 1 のページディレクトリインデックス及び第 1 のページオフセット、  
を含み、並びに  
前記第 2 の仮想アドレスが、  
第 2 のページディレクトリインデックス、第 2 のページテーブルインデックス、及び第 2 のページオフセット、又は  
第 2 のページディレクトリインデックス、第 2 のページテーブルインデックス、及び第 2 のページオフセット、  
を含む、請求項 1 又は 2 に記載のシステム。

20

30

**【請求項 4】**

前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換するように構成されることは、  
前記第 1 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って第 1 のベースアドレスを取得することと、  
前記第 1 のベースアドレス及び前記第 1 のページテーブルインデックスに従って第 1 のエントリアドレスを取得することと、  
前記第 1 のエントリアドレスに従って第 1 の物理ページインデックスを取得することと、  
前記第 1 の物理アドレスとなるように前記第 1 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせることと、  
を行うように構成されることを含む、請求項 3 に記載のシステム。

40

**【請求項 5】**

前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換するように構成されることは、  
前記カーネル空間のページサイズを示す第 1 のページサイズインジケータを取得することを行うように構成されることを含み、前記カーネル空間の前記ページサイズが、第 1 又は第 2 のカーネル空間ページサイズを含み、  
前記第 1 のカーネル空間ページサイズを示す前記第 1 のページサイズインジケータに回答して、前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換

50

するように構成されることは、

前記第 1 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って第 1 のベースアドレスを取得することと、

前記第 1 のベースアドレス及び前記第 1 のページテーブルインデックスに従って第 1 のエントリアドレスを取得することと、

前記第 1 のエントリアドレスに従って第 1 の物理ページインデックスを取得することと、

前記第 1 の物理アドレスとなるように前記第 1 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせたことと、

を行うように構成されることをさらに含む、請求項 3 に記載のシステム。

10

【請求項 6】

前記第 2 のカーネル空間ページサイズを示す前記第 1 のページサイズインジケータにตอบสนองして、前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換するように構成されることは、

前記第 1 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って前記第 1 のエントリアドレスを取得することと、

第 1 のアドレスに従って前記第 1 の物理ページインデックスを取得することと、

前記第 1 の物理アドレスとなるように前記第 1 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせたことと、

を行うように構成されることをさらに含む、請求項 5 に記載のシステム。

20

【請求項 7】

前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換するように構成されることは、

前記第 1 の物理アドレスのページが実行可能でないことを示す第 1 の実行不可インジケータを取得するように構成されることをさらに含む、請求項 1 ~ 6 の何れか一項に記載のシステム。

【請求項 8】

前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するように構成されることは、

前記第 2 のルートページテーブルインデックス及び前記第 2 のページディレクトリインデックスに従って第 2 のベースアドレスを取得することと、

前記第 2 のベースアドレス及び前記第 2 のページテーブルインデックスに従って第 2 のエントリアドレスを取得することと、

前記第 2 のエントリアドレスに従って第 2 の物理ページインデックスを取得することと、

前記第 2 の物理アドレスとなるように前記第 2 の物理ページインデックス及び前記第 2 のページオフセットを組み合わせたことと、

を行うように構成されることを含む、請求項 3 ~ 7 の何れか一項に記載のシステム。

30

【請求項 9】

前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するように構成されることは、

前記ユーザ空間のページサイズを示す第 2 のページサイズインジケータを取得することを行うように構成されることを含み、前記ユーザ空間の前記ページサイズが、第 1 又は第 2 のユーザ空間サイズを含み、

前記第 1 のユーザ空間サイズを示す前記第 2 のページサイズインジケータにตอบสนองして、前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するように構成されることは、

前記第 2 のルートページテーブルインデックス及び前記第 2 のページディレクトリインデックスに従って第 2 のベースアドレスを取得することと、

前記第 2 のベースアドレス及び前記第 2 のページテーブルインデックスに従って第 2 の

40

50

エントリアドレスを取得することと、

前記第 2 のエントリアドレスに従って第 2 の物理ページインデックスを取得することと

、

前記第 2 の物理アドレスとなるように前記第 2 の物理ページインデックス及び前記第 2 のページオフセットを組み合わせることと、

を行うように構成されることをさらに含む、請求項 3 ~ 7 の何れか一項に記載のシステム。

【請求項 10】

前記第 2 のユーザ空間サイズを示す前記第 2 のページサイズインジケータに応答して、前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するように構成されることは、

前記第 2 のルートページテーブルインデックス及び前記第 2 のページディレクトリインデックスに従って前記第 2 のエントリアドレスを取得することと、

前記第 2 のエントリアドレスに従って前記第 2 の物理ページインデックスを取得することと、

前記第 2 の物理アドレスとなるように前記第 2 の物理ページインデックス及び前記第 2 のページオフセットを組み合わせることと、

を行うように構成されることをさらに含む、請求項 9 に記載のシステム。

【請求項 11】

前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するように構成されることは、

前記第 2 の物理アドレスのページが実行可能でないことを示す第 2 の実行不可インジケータを取得するように構成されることをさらに含む、請求項 1 ~ 8 の何れか一項に記載のシステム。

【請求項 12】

前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換するように構成されることは、

第 1 のユーザ空間アクセスインジケーションを取得することと、

前記第 2 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って、第 2 のベースアドレスを取得することと、

前記第 2 のベースアドレス及び前記第 1 のページテーブルインデックスに従って第 2 のエントリアドレスを取得することと、

前記第 2 のエントリアドレスに従って第 2 の物理ページインデックスを取得することと

、

前記第 2 の物理アドレスとなるように前記第 2 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせることと、

を行うように構成されることを含む、請求項 3 に記載のシステム。

【請求項 13】

前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するように構成されることは、

第 2 のユーザ空間アクセスインジケーションを取得することと、

一般保護違反を設定することと、

を行うように構成されることを含む、請求項 1 ~ 8 の何れか一項に記載のシステム。

【請求項 14】

前記制御ユニットが、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換するように構成されることは、

前記第 1 の物理アドレスとなるように前記第 1 の仮想アドレスを直接変換するように構成されることを含む、請求項 1 ~ 3 の何れか一項に記載のシステム。

【請求項 15】

前記制御ユニットが、前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換するよ

10

20

30

40

50

うに構成されることは、

前記第 2 の物理アドレスとなるように前記第 2 の仮想アドレスを直接変換するように構成されることを含む、請求項 1 ~ 8 の何れか一項に記載のシステム。

【請求項 16】

カーネル空間及びユーザ空間のメモリ管理のための方法であって、

前記カーネル空間に対応する第 1 のルートページテーブルインデックスを取得することであって、前記第 1 のルートページテーブルインデックスが、前記ユーザ空間に対応する第 2 のルートページテーブルインデックスとは異なる、取得することと、

オペレーティングシステムカーネルのために、前記第 1 のルートページテーブルインデックスに従って第 1 の仮想アドレスを第 1 の物理アドレスに変換することと、

を含む、方法。

【請求項 17】

前記カーネル空間及び前記ユーザ空間が、同じ物理メモリ空間に対応し、前記同じ物理メモリ空間が、物理メモリ空間全体又は前記物理メモリ空間全体の一部である、請求項 16 に記載の方法。

【請求項 18】

前記第 1 の仮想アドレスが、

第 1 のページディレクトリインデックス、第 1 のページテーブルインデックス、及び第 1 のページオフセット、又は

第 1 のページディレクトリインデックス及び第 1 のページオフセット、

を含む、請求項 16 又は 17 に記載の方法。

【請求項 19】

前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換することが、

前記第 1 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って第 1 のベースアドレスを取得することと、

前記第 1 のベースアドレス及び前記第 1 のページテーブルインデックスに従って第 1 のエントリアドレスを取得することと、

前記第 1 のエントリに従って第 1 の物理ページインデックスを取得することと、

前記第 1 の物理アドレスとなるように前記第 1 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせることと、

を含む、請求項 18 に記載の方法。

【請求項 20】

前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換することが、

前記カーネル空間のページサイズを示す第 1 のページサイズインジケータを取得することを含み、前記カーネル空間の前記ページサイズが、第 1 又は第 2 のカーネル空間ページサイズを含み、

前記第 1 のカーネル空間ページサイズを示す前記第 1 のページサイズインジケータに回答して、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換することが、

前記第 1 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って第 1 のベースアドレスを取得することと、

前記第 1 のベースアドレス及び前記第 1 のページテーブルインデックスに従って第 1 のエントリアドレスを取得することと、

前記第 1 のエントリアドレスに従って第 1 の物理ページインデックスを取得することと、

、

前記第 1 の物理アドレスとなるように前記第 1 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせることと、

をさらに含む、請求項 18 に記載の方法。

【請求項 21】

前記第 2 のカーネル空間ページサイズを示す前記第 1 のページサイズインジケータに回答して、前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換することが、

前記第 1 のルートページテーブルインデックス及び前記第 1 のページディレクトリインデックスに従って前記第 1 のエントリアドレスを取得することと、

前記第 1 のアドレスに従って前記第 1 の物理ページインデックスを取得することと、

前記第 1 の物理アドレスとなるように前記第 1 の物理ページインデックス及び前記第 1 のページオフセットを組み合わせたことと、

を含む、請求項 20 に記載の方法。

【請求項 22】

前記第 1 の仮想アドレスを前記第 1 の物理アドレスに変換することが、

前記第 1 の物理アドレスのページが実行可能でないことを示す第 1 の実行不可インジケータを取得することをさらに含む、請求項 16 ~ 21 の何れか一項に記載の方法。

10

【請求項 23】

前記ユーザ空間に対応する前記第 2 のルートページテーブルインデックスを取得することと、

ユーザプロセスのために、前記第 2 のルートページテーブルインデックスに従って第 2 の仮想アドレスを第 2 の物理アドレスに変換することと、

をさらに含む、請求項 16 ~ 22 の何れか一項に記載の方法。

【請求項 24】

前記第 2 の仮想アドレスが、

第 2 のページディレクトリインデックス、第 2 のページテーブルインデックス、及び第 2 のページオフセット、又は

20

第 2 のページディレクトリインデックス、第 2 のページテーブルインデックス、及び第 2 のページオフセット、

を含む、請求項 16 ~ 23 の何れか一項に記載の方法。

【請求項 25】

前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換することが、

前記第 2 のルートページテーブルインデックス及び前記第 2 のページディレクトリインデックスに従って第 2 のベースアドレスを取得することと、

前記第 2 のベースアドレス及び前記第 2 のページテーブルインデックスに従って第 2 のエントリアドレスを取得することと、

前記第 2 のエントリアドレスに従って第 2 の物理ページインデックスを取得することと

30

、  
前記第 2 の物理アドレスとなるように前記第 2 の物理ページインデックス及び前記第 2 のページオフセットを組み合わせたことと、

を含む、請求項 24 に記載の方法。

【請求項 26】

前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換することが、

前記ユーザ空間のページサイズを示す第 2 のページサイズインジケータを取得することを含み、前記ユーザ空間の前記ページサイズが、第 1 又は第 2 のユーザ空間サイズを含み

、  
前記第 1 のユーザ空間サイズを示す前記第 2 のページサイズインジケータに応答して、  
前記第 2 の仮想アドレスを前記第 2 の物理アドレスに変換することが、

40

前記第 2 のルートページテーブルインデックス及び前記第 2 のページディレクトリインデックスに従って第 2 のベースアドレスを取得することと、

前記第 2 のベースアドレス及び前記第 2 のページテーブルインデックスに従って第 2 のエントリアドレスを取得することと、

前記第 2 のエントリアドレスに従って第 2 の物理ページインデックスを取得することと

、  
前記第 2 の物理アドレスとなるように前記第 2 の物理ページインデックス及び前記第 2 のページオフセットを組み合わせたことと、

をさらに含む、請求項 24 に記載の方法。

50

**【請求項 27】**

前記第2のユーザ空間サイズを示す前記第2のページサイズインジケータに応答して、  
前記第2の仮想アドレスを前記第2の物理アドレスに変換することが、

前記第2のルートページテーブルインデックス及び前記第2のページディレクトリインデックスに従って前記第2のエントリアドレスを取得することと、

前記第2のエントリアドレスに従って前記第2の物理ページインデックスを取得することと、

前記第2の物理アドレスとなるように前記第2の物理ページインデックス及び前記第2のページオフセットを組み合わせたことと、

を含む、請求項26に記載の方法。

10

**【請求項 28】**

前記第2の仮想アドレスを前記第2の物理アドレスに変換することが、

前記第2の物理アドレスのページが実行可能でないことを示す第2の実行不可インジケータを取得することをさらに含む、請求項23～27の何れか一項に記載の方法。

**【請求項 29】**

前記第1の仮想アドレスを前記第1の物理アドレスに変換することが、

前記第1の物理アドレスとなるように前記第1の仮想アドレスを直接変換することを含む、請求項16～18の何れか一項に記載の方法。

**【請求項 30】**

前記第2の仮想アドレスを前記第2の物理アドレスに変換することが、

前記第2の物理アドレスとなるように前記第2の仮想アドレスを直接変換することを含む、請求項23、24、又は28の何れか一項に記載の方法。

20

**【請求項 31】**

前記第1の仮想アドレスが、

第1のページディレクトリインデックス、第1のページテーブルインデックス、及び第1のページオフセット、又は

第1のページディレクトリインデックス、及び第1のページオフセット、  
を含み、

前記第2の仮想アドレスが、

第2のページディレクトリインデックス、第2のページテーブルインデックス、及び第2のページオフセット、又は

30

第2のページディレクトリインデックス、第2のページテーブルインデックス、及び第2のページオフセット、

を含み、並びに

前記第1の仮想アドレスを前記第1の物理アドレスに変換することが、

第1のユーザ空間アクセスインジケーションを取得することと、

前記第2のルートページテーブルインデックス及び前記第1のページディレクトリインデックスに従って第2のベースアドレスを取得することと、

前記第2のベースアドレス及び前記第1のページテーブルインデックスに従って第2のエントリアドレスを取得することと、

40

前記第2のエントリアドレスに従って第2の物理ページインデックスを取得することと、

前記第2の物理アドレスとなるように前記第2の物理ページインデックス及び前記第1のページオフセットを組み合わせたことと、

を含む、請求項23に記載の方法。

**【請求項 32】**

前記第2の仮想アドレスを前記第2の物理アドレスに変換することが、

第2のユーザ空間アクセスインジケーションを取得することと、

一般保護違反を設定することと、

を含む、請求項23又は24に記載の方法。

50

## 【請求項 3 3】

命令のセットを保存する非一時的コンピュータ可読媒体であって、前記命令のセットは、カーネル空間及びユーザ空間のメモリ管理のための方法を装置に行わせるために、前記装置の 1 つ又は複数のプロセッサによって実行可能であり、前記方法が、

前記カーネル空間に対応する第 1 のルートページテーブルインデックスを取得することであって、第 1 のルートページテーブルインデックスが、前記ユーザ空間に対応する第 2 のルートページテーブルインデックスとは異なる、取得することと、

オペレーティングシステムカーネルのために、前記第 1 のルートページテーブルインデックスに従って第 1 の仮想アドレスを第 1 の物理アドレスに変換することと、

を含む、非一時的コンピュータ可読媒体。

10

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

## 技術分野

[001] 本出願は、メモリ管理に関し、より詳細には、コンピュータにおけるカーネル空間及びユーザ空間のメモリ管理のための方法及びシステムに関する。

## 【背景技術】

## 【0002】

## 背景

[002] 従来のコンピュータでは、メモリに保存された命令及び / 又はデータは、仮想アドレスを使用してアドレス指定され得る。プロセスが命令及び / 又はデータにアクセスする必要がある場合、仮想アドレスをメモリの物理アドレスに変換し、それに応じて命令及び / 又はデータを取り出す必要がある場合がある。オペレーティングシステム (OS) カーネルは、仮想アドレスを物理アドレスに変換するための 1 つ又は複数のページテーブルを使用し得る。例えば、Linuxカーネルは、4 キロバイト (KB) のページサイズを用いて、32 ビットの仮想アドレスを 32 ビットの物理アドレスに変換するための 3 レベルページテーブルを使用し得る。

20

## 【0003】

[003] 現在、OSカーネルは、当該OSカーネル (又は別のOSカーネル) 及びユーザプロセスのために、仮想アドレスをカーネル空間及びユーザ空間にそれぞれ分けることができる。カーネル空間は、一般に、高い中央処理装置 (CPU) 特権レベルで処理されるが、ユーザ空間は、低いCPU特権レベルで対処され得る。OSカーネル及びユーザプロセスはそれぞれ、独自のルートページテーブルインデックスを有し得るが、それらは一般に、仮想 - 物理アドレスマッピングに関して同じページテーブルを共有する。しかし、このような設計は、メモリ空間を効率的に利用していない場合がある。さらに、セキュリティ上の懸念を生じ得る。例えば、攻撃者が、OSカーネル及びユーザプロセス間の共通のページテーブルを用いて、ユーザ空間で悪意あるプログラムを実行するようにOSカーネルをだまし得る。また、別の態様では、一般的なページテーブルエントリは、32 ビットコンピュータシステムにおいてウイルス防護を強化するためのNo-eXcute (NX) ビットなどの新たなフィーチャのために利用可能な余裕がない場合がある。マルウェア、スパ

30

40

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0004】

## 概要

[004] 本開示の実施形態は、コンピュータ、装置、又はシステムにおけるカーネル空間及びユーザ空間のメモリ管理のための改良された方法及びシステムを提供する。

## 【課題を解決するための手段】

## 【0005】

50



【005】 これらの実施形態は、カーネル空間及びユーザ空間のメモリ管理のためのシステムを含む。このシステムは、カーネル空間に対応する第1のルートページテーブルインデックスを保存するように構成された第1の記憶ユニットを含み得る。このシステムは、ユーザ空間に対応する第2のルートページテーブルインデックスを保存するように構成された第2の記憶ユニットも含み得る。このシステムは、第1のレジスタ及び第2のレジスタと通信可能に結合され、並びにオペレーティングシステムカーネルのために、第1のルートページテーブルインデックスに従って第1の仮想アドレスを第1の物理アドレスに変換し、及びユーザプロセスのために、第2のルートページテーブルインデックスに従って第2の仮想アドレスを第2の物理アドレスに変換するように構成された制御ユニットをさらに含み得る。

10

【0006】

【006】 これらの実施形態は、カーネル空間及びユーザ空間のメモリ管理のための方法も含む。この方法は、カーネル空間に対応する第1のルートページテーブルインデックスを取得することを含み得る。この方法は、ユーザ空間に対応する第2のルートページテーブルインデックスを取得することも含み得る。この方法は、オペレーティングシステムカーネルのために、第1のルートページテーブルインデックスに従って第1の仮想アドレスを第1の物理アドレスに変換することをさらに含み得る。さらに、この方法は、ユーザプロセスのために、第2のルートページテーブルインデックスに従って第2の仮想アドレスを第2の物理アドレスに変換することを含み得る。

20

【0007】

【007】 また、これらの実施形態は、命令を保存する非一時的コンピュータ可読媒体を含み、当該命令は、カーネル空間及びユーザ空間のメモリ管理のための方法を行うために装置の1つ又は複数のプロセッサによって実行可能である。この方法は、カーネル空間に対応する第1のルートページテーブルインデックスを取得することを含み得る。この方法は、ユーザ空間に対応する第2のルートページテーブルインデックスを取得することも含み得る。この方法は、オペレーティングシステムカーネルのために、第1のルートページテーブルインデックスに従って第1の仮想アドレスを第1の物理アドレスに変換することをさらに含み得る。さらに、この方法は、ユーザプロセスのために、第2のルートページテーブルインデックスに従って第2の仮想アドレスを第2の物理アドレスに変換することを含み得る。

30

【0008】

【008】 開示した実施形態のさらなる目的及び利点は、一部は以下の説明に記載され、及び一部は、その説明から明らかとなり、又は実施形態の実施によって学ばれ得る。開示した実施形態の目的及び利点は、特許請求の範囲に記載した要素及び組み合わせによって実現及び達成され得る。

【0009】

【009】 上述の概要及び以下の詳細な説明は、単なる例示及び説明のためのものであり、請求される本発明の制約ではないことが理解されるものとする。

【0010】

図面の簡単な説明

40

【010】 これより、本開示の例示的实施形態を示す添付の図面に言及する。

【図面の簡単な説明】

【0011】

【図1】 【011】本開示の幾つかの実施形態による、カーネル空間及びユーザ空間のメモリ管理のための例示的コンピュータシステムのブロック図を示す。

【図2A】 【012】本開示の幾つかの実施形態による、カーネル空間のメモリ管理のための例示的方法の模式図である。

【図2B】 【013】本開示の幾つかの実施形態による、カーネル空間の例示的なページマッピングの模式図である。

【図3A】 【014】本開示の幾つかの実施形態による、カーネル空間のメモリ管理のための

50

例示的方法の模式図である。

【図 3 B】[015]本開示の幾つかの実施形態による、カーネル空間の例示的なページマッピングの模式図である。

【図 4 A】[016]本開示の幾つかの実施形態による、ユーザ空間のメモリ管理のための例示的方法の模式図である。

【図 4 B】[017]本開示の幾つかの実施形態による、ユーザ空間の例示的なページマッピングの模式図である。

【図 5 A】[018]本開示の幾つかの実施形態による、ユーザ空間のメモリ管理のための例示的方法の模式図である。

【図 5 B】[019]本開示の幾つかの実施形態による、ユーザ空間の例示的なページマッピングの模式図である。

【図 6】[020]本開示の幾つかの実施形態による、例示的メモリ管理方法のフローチャートである。

【発明を実施するための形態】

【0012】

詳細な説明

[021] これより、例示的实施形態に詳細に言及し、それらの例を添付の図面に示す。

以下の説明は、別段の表示がなければ、異なる図面の同一の番号が、同一又は類似の要素を表す添付の図面を参照する。例示的实施形態の以下の説明に記載する実施態様は、本発明に合致する全ての実施態様を表すわけではない。実際、それらは、添付の特許請求の範囲に記載されるような本発明に関する態様に合致した装置及び方法の単なる例である。

【0013】

[022] 本開示の実施形態は、コンピュータ、装置、又はシステムにおけるカーネル空間及びユーザ空間のメモリ管理のための改良された方法及びシステムを提供する。本明細書に記載の実施形態は、カーネル空間及びユーザ空間のメモリ管理のためにそれぞれのルートページテーブルインデックスを保存するための専用レジスタをOSカーネル及びユーザプロセスにそれぞれ割り当てることができる。OSカーネル及びユーザプロセスには、ページディレクトリ、ページテーブル、及び仮想 - 物理アドレスマッピングに関するページテーブルエントリもそれぞれ与えられる。OSカーネル及びユーザプロセスの仮想 - 物理アドレスマッピングは、物理メモリ空間全体にそれぞれマッピングされ得る。例えば、OSカーネル及びユーザプロセスは、32ビットコンピュータシステムにおける4ギガバイト (GB) の物理メモリ空間全体に対応する4GBのカーネル空間及びユーザ空間をそれぞれ有し得る。代替的に、OSカーネル及びユーザプロセスの仮想 - 物理アドレスマッピングは、物理メモリ空間全体の一部にそれぞれマッピングされてもよい。例えば、OSカーネル及びユーザプロセスは、64ビットコンピュータシステムにおける16エクサバイト (EB) (すなわち、 $16 \times 10^9$  GB) の物理メモリ空間全体の一部に対応する32GBのカーネル空間及びユーザ空間をそれぞれ有し得る。それによって、メモリ空間の使用を向上させ、且つ関連のセキュリティの脆弱性を除去することができる。

【0014】

[023] それに応じて、OSカーネルは、OSカーネル及びユーザプロセスに関してそれぞれ仮想アドレスを物理アドレスに変換することによって、マルウェア、スパイウェア、及び/又はウイルスから防護され得る。加えて、仮想アドレスから物理アドレスへのこれらの変換は、個別に行われるが、一般的なページテーブルエントリは、32ビットコンピュータシステムにおいてウイルス防護を強化するためのNo-eXecute (NX) ビットなどの新たなフィーチャのために生み出された余裕を持ち得る。それに応じて、コンピュータシステムのセキュリティが強化及び向上され得る。

【0015】

[024] 幾つかの実施形態によれば、本明細書に記載の動作、技術、及び/又はコンポーネントは、1つ又は複数の専用コンピューティングデバイスを含み得る電子デバイスによって実施され得る。専用コンピューティングデバイスは、本明細書に記載の動作、技術

、及び／又はコンポーネントを行うために配線接続されてもよく、或いは本明細書に記載の動作、技術、及び／又はコンポーネントを行うために永続的にプログラム化された１つ又は複数の特定用途向け集積回路（ＡＳＩＣ）又はフィールドプログラマブルゲートアレイ（ＦＰＧＡ）などのデジタル電子デバイスを含んでもよく、或いはファームウェア、メモリ、他のストレージ、又は組み合わせにおけるプログラム命令に従って、本開示の上記機能を実現するようにプログラム化された１つ又は複数のハードウェアプロセッサを含んでもよい。このような専用コンピューティングデバイスは、本開示の技術及び他のフィーチャを達成するために、カスタムハードワイヤードロジック、ＡＳＩＣ、又はＦＰＧＡをカスタムプログラミングと組み合わせることもできる。専用コンピューティングデバイスは、デスクトップコンピュータシステム、ポータブルコンピュータシステム、ハンドヘルドデバイス、ネットワーキングデバイス、又は本開示の技術及び他のフィーチャを実施するためにハードワイヤード及び／又はプログラムロジックを組み込むその他のデバイスでもよい。

10

20

30

40

50

#### 【００１６】

[025] １つ又は複数の専用コンピューティングデバイスは、一般に、iOS、Android（登録商標）、Blackberry（登録商標）、Chrome OS、Windows（登録商標）XP、Windows Vista、Windows 7、Windows 8、Windows Server、Windows CE、Unix（登録商標）、Linux（登録商標）、SunOS、Solaris、VxWorks、又は他の互換性があるオペレーティングシステムなどのオペレーティングシステムソフトウェアによって制御及び調整され得る。他の実施形態では、コンピューティングデバイスは、所有オペレーティングシステムによって制御され得る。オペレーティングシステムは、実行するコンピュータプロセスの制御及びスケジューリングを行い、メモリ管理を行い、ファイルシステム、ネットワーキング、Ｉ／Ｏサービスを提供し、並びに特にグラフィカルユーザインタフェース（「ＧＵＩ」）などのユーザインタフェース機能を提供する。

#### 【００１７】

[026] 図１は、本開示の幾つかの実施形態による、カーネル空間及びユーザ空間のメモリ管理のための例示的コンピュータシステム１００のブロック図を示す。幾つかの実施形態では、コンピュータシステム１００は、メモリ１１０、プロセッサ１２０、ストレージ１３０、及び入出力（Ｉ／Ｏ）インタフェース１４０を含み得る。

#### 【００１８】

[027] メモリ１１０は、プロセッサ１２０が演算を行う必要があり得るあらゆるタイプの情報を保存するために設けられた適宜のタイプのマスストレージを含み得る。例えば、メモリ１１０は、ダイナミックランダムアクセスメモリ（ＤＲＡＭ）を含んでもよく、及びコンピュータシステム１００のメインメモリとなるように構成されてもよい。幾つかの実施形態では、メモリ１１０は、揮発性若しくは不揮発性、磁気、半導体、テープ、光学式、取り外し可能、固定型、若しくは他のタイプのストレージデバイス、又は有形及び／又は非一時的コンピュータ可読媒体でもよい。非一時的媒体の一般的形態には、例えば、フロッピーディスク、フレキシブルディスク、ハードディスク、固体ドライブ、磁気テープ、又はその他の磁気データストレージ媒体、ＣＤ－ＲＯＭ、その他の光学式データストレージ媒体、孔のパターンを有した物理的媒体、ＲＡＭ、ＰＲＯＭ、及びＥＰＲＯＭ、ＦＬＡＳＨ（商標登録）－ＥＰＲＯＭ又はその他のフラッシュメモリ、ＮＶＲＡＭ、キャッシュ、その他のメモリチップ又はカートリッジ、並びに上記のネットワークバージョンが含まれる。

#### 【００１９】

[028] 幾つかの実施形態では、メモリ１１０は、本出願で開示する例示的メモリ管理方法を行うためにプロセッサ１２０によって実行され得る１つ又は複数のコンピュータプログラムを保存するように構成され得る。例えば、メモリ１１０は、本開示に記載するようにメモリのカーネル空間及びユーザ空間を管理するためにプロセッサ１２０によって実行され得る１つ又は複数のプログラムを保存するように構成され得る。別の例として、メモリ１１０は、メモリ管理、プロセス管理、資源配分、Ｉ／Ｏデバイス管理、プロセス間

通信、マルチタスクスケジューリング、システムコール及び割込み処理、並びにセキュリティ又は防護管理などのコンピュータシステム 100 のシステム全体を動作させるために、プロセッサ 120 によって実行され得る OS カーネルを保存するように構成され得る。

【0020】

[029] 幾つかの実施形態では、メモリ 110 は、プロセッサ 120 がアクセスする情報及びデータを保存するようにも構成され得る。例えば、メモリ 110 は、ユーザプロセスを実行する際にプロセッサ 120 がアクセスし得る音声、映像、又は文書データを保存するように構成され得る。幾つかの実施形態では、メモリ 110 が、プロセッサ 120 がアクセスしたいデータ又はプログラムを含まない可能性がある場合には、プロセッサ 120 によって実行される OS カーネルが、メモリ 110 から外に幾つかのページを移動させ、及びストレージ 130 からメモリ 110 内に必要とされるデータ又はプログラムを移動させることができる。このプロセスは、スワッピングと呼ばれる場合もあり、プロセッサ 120 にとって必要とされるデータ又はプログラムがメモリ 110 内で利用可能であることを確実にするために使用され得る。

【0021】

[030] 物理アドレスは、メモリ 110 の位置を指す又はアドレス指定するメモリアドレスでよく、それに応じてプロセッサ 120 がメモリ 110 内のその位置にあるデータ又はプログラムにアクセスすることを可能にし得る。例えば、32 ビットの OS は、32 ビットのメモリアドレスを含み得る。32 ビットのメモリアドレスは、メモリ 110 の最大 4 ギガバイト (GB) を指し得る。

【0022】

[031] 一方、仮想アドレスは、プロセスに関する仮想メモリを指す又はアドレス指定するメモリアドレスでよい。OS カーネルは、OS カーネル自体、別の OS カーネル、又は各ユーザプロセスなどの各プロセスに関する仮想メモリの生成及び管理を行い得る。仮想メモリにより、プロセスは、共有物理メモリ空間を管理する必要をなくすことができ、メモリの分離によるセキュリティの向上を得ることができ、及びページング技術を用いた場合に物理的に利用可能となり得るよりも多くのメモリを概念的に使用することができる。例えば、32 ビットの Microsoft Windows は、4 GB の仮想メモリを含み得る。32 ビットの Windows の各プロセスは、そのメモリとして 4 GB の仮想アドレス空間を使用し得る。各プロセスは、共有物理メモリ空間を管理する必要がなくともよく、及び仮想アドレスから物理アドレスへの変換を管理する必要がなくともよい。

【0023】

[032] 幾つかのプロセスに関して、仮想アドレスは、メモリ 110 の物理アドレスにマッピングされ得る。代替的に、仮想アドレスは、メモリ 110 及び / 又はストレージ 130 の物理アドレスにマッピングされ得る。OS カーネルは、そのメモリ管理機能により、各プロセスに関する仮想アドレス及び物理アドレス間の変換を管理し得る。OS カーネルは、各プロセスに関する仮想アドレスに対応する物理アドレスに変換するための 1 つ又は複数のページテーブルを生成することができる。例えば、x86 中央処理装置 (CPU) 上で走り、且つ 4 KB のページサイズを使用する 32 ビットの Linux カーネルは、各プロセスに関して、そのメインメモリ内で 3 レベルページテーブル構造の生成及び管理を行い得る。3 つのレベルは、ページディレクトリ、ページテーブル、及びページテーブルエントリのレベルを含み得る。OS カーネルは、各プロセスに関するルートページテーブルインデックスを保持することができ、このようなルートページテーブルインデックス及び 3 レベルページテーブルを使用して、仮想アドレスをメモリ 110 の物理アドレスに変換することができる。

【0024】

[033] プロセッサ 120 は、マイクロプロセッサ、デジタル信号プロセッサ、コントローラ、又はマイクロコントローラを含み得る。プロセッサ 120 は、OS カーネル及びユーザプロセスのためにカーネル空間及びユーザ空間をそれぞれ管理するように構成され得る。プロセッサ 120 は、図 1 に示すように、制御レジスタ 122 及び制御レジスタ 1

24を含み得る。プロセッサ120は、制御レジスタ122内にカーネル空間に対応するルートページテーブルインデックスを保存するように構成され得る。カーネル空間に対応するルートページテーブルインデックスは、図2Aに示すように、及び以下にさらに記載するように、仮想アドレスを物理アドレスに変換するためにOSカーネルに使用され得る。

【0025】

[034] プロセッサ120は、制御レジスタ124内にユーザ空間に対応する別のルートページテーブルインデックスを保存するように構成されてもよい。ユーザ空間に対応するルートページテーブルインデックスは、図3Aに示すように、及び以下に記載するように、仮想アドレスを物理アドレスに変換するためにユーザプロセスに使用され得る。各ユーザプロセスは、ユーザ空間に対応するそのルートページテーブルインデックスを有し得る。プロセッサ120は、プロセッサ120が別のユーザプロセスを実行するように構成され得る場合にはいつでも、別のルートページテーブルインデックスを用いて制御レジスタ124を更新するように構成され得る。

10

【0026】

[035] 上記の通り、制御レジスタ122及び制御レジスタ124は、OSカーネル及びユーザプロセスのルートページテーブルインデックスを保存するための記憶ユニットとなるように構成され得る。OSカーネルに関して仮想アドレスから物理アドレスへの変換がある場合、制御レジスタ122は、OSカーネルのルートページテーブルインデックスを保存するための記憶ユニットとして構成され得る。一方、ユーザプロセスに関して仮想アドレスから物理アドレスへの変換がある場合はいつでも、制御レジスタ124は、ユーザプロセスのルートページテーブルインデックスを保存するための記憶ユニットとして構成され得る。幾つかの実施形態では、制御レジスタ122は、ユーザプロセスのルートページテーブルインデックスを保存するように構成されなくてもよい。幾つかの実施形態では、制御レジスタ124は、OSカーネルのルートページテーブルインデックスを保存するように構成されなくてもよい。

20

【0027】

[036] 例えば、本開示による新たなプロセッサは、OSカーネル及びユーザプロセスそれぞれのルートページテーブルインデックスを保存するためのOSカーネル用の制御レジスタ3(CR3K)及びユーザプロセス用の別の制御レジスタ3(CR3U)を含み得る。CR3K及びCR3Uは、2つの別個のレジスタである。カーネルOSが起動すると、カーネルOSに関する仮想-物理マッピングを使用可能にする前に、CPUは、最初にページング使用不能モードで動作し得る。カーネル空間が物理メモリの空間全体にマッピングされ得るので、OSカーネルは、カーネル空間からシステムの物理メモリへの1対1マッピングをセットアップし得る。例えば、32GBの物理メモリを有する64ビットシステムでは、OSカーネルは、その仮想アドレス空間の最初の32GBを32GB物理メモリにマッピングし得る。初期化後に、OSカーネルは、OSカーネルのルートページテーブルインデックスをCR3Kに書き込み、CPUをページング使用可能モードに切り替える。OSカーネルは、仮想-物理アドレスマッピングの動作を開始する。

30

【0028】

[037] OSカーネルは、各ユーザプロセスのページテーブルの管理及びセットアップを行う責任も負う。ユーザプロセスが開始されると、OSカーネルは、ユーザプロセスに関するページディレクトリ、ページテーブル、及びページテーブルエントリをセットアップする。ユーザ空間も物理メモリの空間全体にマッピングされ得るので、ユーザプロセスは、柔軟性があり、且つ効率的な仮想-物理アドレスマッピングを有し得る。ユーザプロセスに関する初期化後に、OSカーネルは、ユーザプロセスのルートページテーブルインデックスをCR3Uに書き込むとともに、プロセスのタスク空間にそれを保存する。コンテキスト切換えがOSカーネルのスケジューラで生じると、別のユーザプロセスが、現在のユーザプロセスを置換する必要がある。CPUは、別のユーザプロセスのルートページテーブルインデックスを読み出すことができ、それをCR3Uに書き込む。次いで、CP

40

50

Uは、別のユーザプロセスを実行するために切り替わる。

【0029】

[038] 幾つかの実施形態では、x86CPUは、その制御レジスタ3(CR3)にOSカーネルのルートページテーブルインデックスを保存するように構成され得る。また、x86CPUは、その制御レジスタ4(CR4)にユーザプロセスのルートページテーブルインデックスを保存するように構成されてもよい。

【0030】

[039] 代替的に、コンピュータシステム100内であるが、プロセッサ120の外にある2つの記憶ユニット(不図示)が、OSカーネル及びユーザプロセスそれぞれに関する仮想アドレスを物理アドレスに変換するためのOSカーネル及びユーザプロセスのルートページテーブルインデックスを保存するように構成され得る。例えば、キャッシュの2つの32ビット空間が、Linuxカーネル及びユーザプロセスのルートページテーブルインデックスを保存するための2つの記憶ユニットとなるように構成されてもよい。別の例として、メモリ110の2つの32ビット空間が、Linuxカーネル及びユーザプロセスのルートページテーブルインデックスを保存するための記憶ユニットとなるように構成されてもよい。幾つかの実施形態では、コンピュータシステム100内であるが、プロセッサ120の外にある2つの記憶ユニット(不図示)の一方が、OSカーネル又はユーザプロセスに関する仮想アドレスを物理アドレスに変換するためのOSカーネル又はユーザプロセスのルートページテーブルインデックスを保存するように構成され得る。他方の記憶ユニットは、プロセッサ120における制御レジスタでもよい。例えば、制御レジスタ122は、Linuxカーネルのルートページテーブルインデックスを保存するように構成され得るが、メモリ110の32ビット空間は、ユーザプロセスのルートページテーブルインデックスを保存するように構成され得る。

【0031】

[040] プロセッサ120は、OSカーネル及び/又はユーザプロセスに関して本開示全体のメモリ管理を行うためのメモリ管理ユニットを含み得る。

【0032】

[041] ストレージ130は、プロセッサ120が演算を行う必要があり得るあらゆるタイプの情報を保存するために設けられた適宜のタイプのマスメモリを含み得る。ストレージ130は、揮発性若しくは不揮発性、磁気、半導体、テープ、光学式、取り外し可能、固定型、若しくは他のタイプのストレージデバイス、又は有形及び/又は非一時的コンピュータ可読媒体でもよい。ストレージ130は、本出願で開示する例示的メモリ管理方法を行うためにプロセッサ120によって実行され得る1つ又は複数のコンピュータプログラムを保存するように構成され得る。例えば、ストレージ130は、上記のように、OSカーネル及びユーザプロセスに関して仮想アドレスを物理アドレスに変換するためにプロセッサ120によって実行され得る1つ又は複数のプログラムを保存するように構成され得る。

【0033】

[042] ストレージ130は、プロセッサ120によって使用される情報及びデータを保存するようにさらに構成され得る。例えば、ストレージ130は、メモリ110が保存できない可能性があるデータを保存するように構成され得る。メモリ110が一杯になった場合、メモリ110内の一部のデータが、ストレージ130へとスワッピングされ得る。

【0034】

[043] I/Oインタフェース140は、コンピュータシステム100と他の装置との間の通信を容易にするように構成され得る。例えば、I/Oインタフェース140は、別の装置、例えば別のコンピュータからデータ又は命令を受信するように構成され得る。I/Oインタフェース140は、他の装置、例えばラップトップコンピュータ又はスピーカにデータ又は命令を出力するようにも構成され得る。

【0035】

10

20

30

40

50

[044] プロセッサ 120 は、カーネル空間及びユーザ空間のメモリ管理のためのそれぞれのルートページテーブルインデックスを保存するための専用レジスタ CR3K 及び CR3U を OS カーネル及びユーザプロセスにそれぞれ割り当てるように構成され得る。プロセッサ 120 は、OS カーネル及びユーザプロセスに関するページディレクトリ、ページテーブル、及びページテーブルエントリをそれぞれ構築し、及び取り出すようにも構成され得る。専用 CR3K 及び CR3U と、各ページテーブルとを用いて、プロセッサ 120 は、カーネル空間及びユーザ空間を物理メモリ空間全体にマッピングするように構成され得る。例えば、プロセッサ 120 は、カーネル空間及びユーザ空間の両方を 32 ビットシステムの 4 GB の物理メモリ空間全体にマッピングするように構成され得る。つまり、カーネル空間及びユーザ空間は共に、4 GB の空間である。代替的に、プロセッサ 120 は、カーネル空間及びユーザ空間を物理メモリ空間全体の一部にマッピングするように構成され得る。例えば、プロセッサ 120 は、64 ビットコンピュータシステムにおける 16 エクサバイト (EB) (すなわち、 $16 \times 10^9$  GB) の物理メモリ空間全体の一部に対応する 32 GB のカーネル空間及びユーザ空間をマッピングするように構成され得る。

10

【0036】

[045] プロセッサ 120 は、仮想 - 物理アドレスマッピングのために OS カーネル又はユーザプロセスから仮想アドレスを取得するように構成され得る。OS カーネルから仮想アドレスを取得した後に、プロセッサ 120 は、デフォルト結果としてカーネル空間にアクセスすることを決定するように構成され得る。ユーザプロセスから仮想アドレスを取得した後に、プロセッサ 120 は、デフォルト結果としてユーザ空間にアクセスすることを決定するように構成され得る。幾つかの実施形態では、プロセッサ 120 は、OS カーネルのために、ユーザ空間アクセスインジケーションを取得するように構成され得る。OS カーネルからユーザ空間アクセスインジケーションを取得した後に、プロセッサ 120 は、ユーザ空間にアクセスすることを決定するように構成され得る。

20

【0037】

[046] 図 2A 及び 2B は、本開示の幾つかの実施形態による、カーネル空間のメモリ管理の例示的方法、及び例示的な対応するページマッピングの模式図である。OS カーネルに関する仮想アドレスは、ページディレクトリインデックス、ページテーブルインデックス、及びページオフセットを含み得る。例えば、図 2A に示すように、32 ビットの OS カーネルに関する仮想アドレス 270 は、ページディレクトリインデックス 271、ページテーブルインデックス 272、及びページオフセット 273 を含み得る。

30

【0038】

[047] ページディレクトリインデックス 271 は、ページディレクトリのインデックスでよく、ページテーブルのベースアドレスを含み得るページディレクトリを指すルートアドレスと共に使用され得る。図 2A では、制御レジスタ 122 が、例えば、OS カーネルに関するルートページテーブルインデックスをルートアドレス 210 として保存し得る。ページディレクトリインデックス 271 は、ページテーブル 240 のベースアドレス 230 を含むページディレクトリ (PD) 223 を指すルートアドレス 210 と共に使用され得る。

【0039】

40

[048] ページテーブルインデックス 272 は、ページテーブルのインデックスでよく、ページテーブルエントリのエントリアドレスを含み得るページテーブル (PT) を指すベースアドレス 230 と共に使用され得る。例えば、ページテーブルインデックス 272 は、図 2A に示すように、ページテーブルエントリ 260 のエントリアドレス 250 を含み得るページテーブル 242 を指すベースアドレス 230 と共に使用され得る。

【0040】

[049] ページオフセット 273 は、ページのオフセットでよく、変換された物理アドレスにおいてページオフセットとして使用され得る。例えば、仮想アドレス 270 のページオフセット 273 は、図 2A に示すように、物理アドレス 290 のページオフセット 293 としてそのまま使用され得る。

50

## 【 0 0 4 1 】

[050] OSカーネルに関して仮想アドレスを物理アドレスに変換するために、プロセッサ120は、記憶ユニットにOSカーネルのルートページテーブルインデックスを保存するように構成され得る。例えば、プロセッサ120は、Linuxカーネルに関するルートページテーブルインデックスを制御レジスタ122に保存するように構成され得る。プロセッサ120は、図2Aに示すように、ページディレクトリ220において、ある位置（例えばページディレクトリ221）を指すルートアドレス210として、保存されたルートページテーブルインデックスを使用するように構成され得る。プロセッサ120は、制御レジスタ122に保存されたルートページテーブルインデックスをページディレクトリインデックス271と組み合わせることによって、ページテーブルのベースアドレスを含み得るページディレクトリを見つけ出すようにも構成され得る。例えば、プロセッサ120は、図2Bに示すように、ルートアドレス210を仮想アドレス270のページディレクトリインデックス271と組み合わせることによって、ページディレクトリ223を見つけ出すように構成され得る。つまり、プロセッサ120は、図2A及び2Bに示すように、ページディレクトリ220における開始位置としてページディレクトリ221を使用し、及びページディレクトリインデックス271をオフセットとして使用することによって、ページディレクトリ223を見つけ出すように構成される。

10

## 【 0 0 4 2 】

[051] ページディレクトリ223は、3レベルページテーブル構造の次のレベルであるページテーブルのベースアドレスを指すベースアドレスを含み得る。図2Bに示すように、ページディレクトリ223は、3レベルページテーブルの次のレベルのページテーブルを指す4KBのベースアドレス223-1を含む。プロセッサ120は、図2Aに示すように、ページディレクトリ223のコンテンツを読み出すことによって、ページテーブル240内のページテーブル241を指すベースアドレス230を取得するようにさらに構成され得る。

20

## 【 0 0 4 3 】

[052] プロセッサ120は、見つけ出されたページディレクトリ223に保存されたベースアドレスをページテーブルインデックス272と組み合わせることによって、ページテーブルエントリのエントリアドレスを含むページテーブルを見つけ出すようにも構成され得る。例えば、プロセッサ120は、図2Bに示すように、ベースアドレス230（すなわち、4KBのベースアドレス223-1）を仮想アドレス270のページテーブルインデックス272と組み合わせることによって、ページテーブル242を見つけ出すように構成され得る。つまり、プロセッサ120は、図2A及び2Bに示すように、ページテーブル240における開始位置としてページテーブル241を使用し、及びページテーブルインデックス272をオフセットとして使用することによって、ページテーブル242を見つけ出すように構成される。

30

## 【 0 0 4 4 】

[053] ページテーブル242は、3レベルページテーブル構造における次のレベルであるページテーブルエントリ中のある位置を指すエントリアドレスを含み得る。図2Bに示すように、ページテーブル242は、3レベルページテーブルの次のレベルのページテーブルエントリを指す4KBのエントリアドレス242-1を含む。プロセッサ120は、図2Aに示すように、ページテーブル242のコンテンツを読み出すことによって、ページテーブルエントリ260のページテーブルエントリ262を指すエントリアドレス250を取得するように構成され得る。

40

## 【 0 0 4 5 】

[054] プロセッサ120は、ページテーブルエントリ262のコンテンツを読み出すことによって、物理ページインデックスを取得するようにさらに構成され得る。図2Bに示すように、ページテーブルエントリ262は、4KBの物理ページインデックス262-1及びページ属性262-2を含む。エントリアドレス250に従って、プロセッサ120は、図2A及び2Bに示すように、物理ページインデックス291となるページテ

50



ブルエントリ 2 6 2 の 4 K B の物理ページインデックス 2 6 2 - 1 を読み出すように構成され得る。

【 0 0 4 6 】

[055] プロセッサ 1 2 0 は、OS カーネルに関する物理アドレスとなるように、物理ページインデックス及びページオフセットを組み合わせるようにさらに構成され得る。例えば、プロセッサ 1 2 0 は、OS カーネルに関する物理アドレス 2 9 0 となるように、取得された物理ページインデックス 2 9 1 (すなわち、4 K B の物理ページインデックス 2 6 2 - 1) 及びページオフセット 2 9 3 (すなわち、ページオフセット 2 7 3) を組み合わせるように構成され得る。それに応じて、プロセッサ 1 2 0 は、変換された物理アドレス 2 9 0 でメモリ 1 1 0 にアクセスすることにより、OS カーネルに関する必要とされるデータ又は命令を取得するように構成され得る。

10

【 0 0 4 7 】

[056] 幾つかの実施形態では、プロセッサ 1 2 0 は、カーネル空間のページサイズ又はカーネル空間ページサイズを示すページサイズインジケータを取得するように構成され得る。図 2 B に示すように、ページテーブルエントリ 2 6 2 は、ページ属性 2 6 2 - 2 を含む。プロセッサ 1 2 0 は、ページテーブルエントリ 2 6 2 におけるページ属性 2 6 2 - 2 を読み出すように構成され得る。ページ属性 2 6 2 - 2 は、カーネル空間のページサイズを示すビット (例えば、ビット 7) を含み得る。例えば、ページ属性 2 6 2 - 2 のビット 7 が「1」である場合、カーネル空間のページサイズは、4 K B でもよい。ページ属性 2 6 2 - 2 のビット 7 が「0」である場合、カーネル空間のページサイズは、4 メガバイト (M B) でもよい。

20

【 0 0 4 8 】

[057] カーネル空間が第 1 のページサイズを含むことをページサイズインジケータが示す場合、プロセッサ 1 2 0 は、上記のように、仮想アドレスを物理アドレスに変換するように構成され得る。例えば、ページ属性 2 6 2 - 2 のビット 7 が、カーネル空間の 4 K B のページサイズを示す「1」である場合、プロセッサ 1 2 0 は、上記のように、並びに図 2 A 及び 2 B に示すように、仮想アドレス 2 7 0 を物理アドレス 2 9 0 に変換するように構成され得る。ページ属性 2 6 2 - 2 のビット 7 が、カーネル空間の 4 M B のページサイズを示す「0」である場合、プロセッサ 1 2 0 は、以下に記載するように、並びに図 3 A 及び 3 B に示すように、仮想アドレス 3 7 0 を物理アドレス 3 9 0 に変換するように構成され得る。

30

【 0 0 4 9 】

[058] カーネル空間のページサイズを示すビットは、ページディレクトリ又はページテーブルの 1 つに含まれてもよい。例えば、ページサイズビットは、図 2 B のページディレクトリ 2 2 3 の予約フィールド 2 2 3 - 2 又はページテーブル 2 4 2 の予約フィールド 2 4 2 - 2 に含まれ得る。例えば、予約フィールド 2 4 2 - 2 のビット 7 が、ページサイズビットとして使用され得る。プロセッサ 1 2 0 は、予約フィールド 2 4 2 - 2 のビット 7 を読み出し、それに応じてカーネル空間のページサイズを取得するように構成され得る。

。

【 0 0 5 0 】

40

[059] 図 2 B を参照すると、ページテーブルエントリ 2 6 2 は、物理ページインデックス 2 6 2 - 1 及びページ属性 2 6 2 - 2 を含む。プロセッサ 1 2 0 は、上記のように仮想アドレスを物理アドレスに変換するために物理ページインデックス 2 6 2 - 1 を使用するように構成され得る。物理ページインデックス 2 6 2 - 1 は、例えば、3 2 ビットコンピュータシステムのカーネル空間において 4 K B のページをアドレス指定するためのページテーブルエントリ 2 6 2 の最上位ビット (M S B) の 2 0 ビットを含み得る。

【 0 0 5 1 】

[060] ページ属性 2 6 2 - 2 は、例えば、3 2 ビットコンピュータシステムにおけるページテーブルエントリ 2 6 2 の最下位ビット (L S B) の 1 2 ビットを含み得る。ページ属性 2 6 2 - 2 のこれらの 1 2 ビットは、変換された物理アドレスのページの属性を示

50

すように構成され得る。例えば、ページ属性 2 6 2 - 2 のビット 7 は、カーネル空間のページサイズを示すように構成され得る。例えば、上記のように、ページ属性 2 6 2 - 2 のビット 7 が「1」である場合、カーネル空間のページサイズは、4 K B となるように構成され得る。ページ属性 2 6 2 - 2 のビット 7 が「0」である場合、カーネル空間のページサイズは、4 M B となるように構成され得る。プロセッサ 1 2 0 は、上記のように、ページ属性 2 6 2 - 2 のビット 7 に応じて、仮想アドレスを物理アドレスに変換するように構成され得る。

【0052】

[061] 3種類以上のページサイズが必要とされる場合、ページ属性 2 6 2 - 2 は、カーネル空間のページサイズを示す 2 ビット（例えば、ビット 9 及びビット 7）を含み得る。図面では、未使用と示されているが、このような使用のために、ページ属性 2 6 2 - 2 のビット 9 がさらに構成され得る。例えば、ページ属性 2 6 2 - 2 のビット 9 及びビット 7 が「11」である場合、カーネル空間のページサイズは、4 K B でもよい。ページ属性 2 6 2 - 2 のビット 9 及びビット 7 が「10」である場合、カーネル空間のページサイズは、4 M B でもよい。ページ属性 2 6 2 - 2 のビット 9 及びビット 7 が「01」である場合、カーネル空間のページサイズは、16 M B でもよい。ページ属性 2 6 2 - 2 のビット 9 及びビット 7 が「00」である場合、カーネル空間のページサイズは、64 M B でもよい。プロセッサ 1 2 0 は、上記に類似したページ属性 2 6 2 - 2 のビット 9 及びビット 7 に応じて仮想アドレスを物理アドレスに変換するように構成され得る。

【0053】

[062] また、ページ属性 2 6 2 - 2 は、変換された物理アドレスのページが実行不可であることを示す NX インジケータ（例えば、ビット 2）を含み得る。例えば、ページ属性 2 6 2 - 2 のビット 2 が「1」である場合、変換された物理アドレス 290 のページは、実行可能でなくてもよい。プロセッサ 1 2 0 は、上記のようにページ属性 2 6 2 - 2 のビット 2 に応じて、変換された物理アドレス 290 のページを実行しないように、又はそのページにアクセスしないように構成され得る。

【0054】

[063] カーネル空間のページサイズを示すビットは、ページディレクトリ又はページテーブルの 1 つに含まれてもよい。例えば、ページサイズビットが、図 2 B のページディレクトリ 223 の予約フィールド 223 - 2 又はページテーブル 242 の予約フィールド 242 - 2 に含まれ得る。例えば、予約フィールド 242 - 2 のビット 7 が、ページサイズビットとして使用され得る。プロセッサ 1 2 0 は、予約フィールド 242 - 2 のビット 7 を読み出し、それに応じてカーネル空間のページサイズを取得するように構成され得る。

【0055】

[064] ページサイズを示すビットに類似して、NX インジケータは、ページディレクトリ又はページテーブルの 1 つに含まれてもよい。例えば、NX ビットは、図 2 B のページディレクトリ 223 の予約フィールド 223 - 2 又はページテーブル 242 の予約フィールド 242 - 2 に含まれ得る。プロセッサ 1 2 0 は、それに応じて変換された物理アドレスのページに関する NX ビットを読み出すように構成され得る。

【0056】

[065] 幾つかの実施形態では、プロセッサ 1 2 0 は、仮想アドレスを物理アドレスに直接変換するように構成され得る。つまり、仮想アドレスは、物理アドレスと同一である。例えば、図 2 A の仮想アドレス 270 は、図に示されるようにページディレクトリ 220、ページテーブル 240、及びページテーブルエントリ 260 を通して変換されることなく、直接物理アドレス 290 となるように変換され得る。仮想アドレスから物理アドレスへの直接変換は、OS カーネルがメモリ 110 に素早く且つ効率的にアクセスする助けとなり得る。

【0057】

[066] カーネル空間が第 2 のページサイズを含むことをページサイズインジケータが

示す場合、プロセッサ 120 は、図 3 A 及び 3 B に示すように、仮想アドレスを物理アドレスに変換するように構成され得る。例えば、図 3 B のページ属性 362 - 2 のビット 7 が、カーネル空間の 4 MB のページサイズを示す「0」である場合に、プロセッサ 120 は、図 3 A に示すように、ページディレクトリ 320 及びページテーブルエントリ 360 を含む 2 レベルページテーブルを通して、仮想アドレス 370 を物理アドレス 390 に変換するように構成され得る。

【0058】

[067] 図 3 A 及び 3 B は、本開示の幾つかの実施形態による、カーネル空間のメモリ管理の例示的方法、及び例示的な対応するページマッピングの模式図である。OS カーネルに関する仮想アドレスは、ページディレクトリインデックス及びページオフセットを含み得る。例えば、図 3 A に示すように、32 ビットの OS カーネルに関する仮想アドレス 370 は、ページディレクトリインデックス 371 及びページオフセット 373 を含み得る。

10

【0059】

[068] ページディレクトリインデックス 371 は、ページディレクトリのインデックスでよく、ページテーブルエントリのエントリアドレスを含み得るページディレクトリを指すルートアドレスと共に使用され得る。図 3 A では、制御レジスタ 122 は、例えば、ルートアドレス 310 として OS カーネルに関するルートページテーブルインデックスを保存するように構成され得る。ページディレクトリインデックス 371 は、ページテーブルエントリ 360 のエントリアドレス 350 を含むページディレクトリ (PD) 323 を指すルートアドレス 310 と共に使用され得る。

20

【0060】

[069] ページオフセット 373 は、ページのオフセットでよく、変換された物理アドレスにおいてページオフセットとして使用され得る。例えば、仮想アドレス 370 のページオフセット 373 は、図 3 A に示すように、物理アドレス 390 のページオフセット 393 としてそのまま使用され得る。

【0061】

[070] OS カーネルに関して仮想アドレスを物理アドレスに変換するために、プロセッサ 120 は、記憶ユニットに OS カーネルのルートページテーブルインデックスを保存するように構成され得る。例えば、プロセッサ 120 は、図 3 A において、Linux カーネルに関するルートページテーブルインデックスを制御レジスタ 122 に保存するように構成され得る。プロセッサ 120 は、図 3 A に示すように、ページディレクトリ 320 において、ある位置 (例えばページディレクトリ 321) を指すルートアドレス 310 として、保存されたルートページテーブルインデックスを使用するように構成され得る。プロセッサ 120 は、制御レジスタ 122 に保存されたルートページテーブルインデックスをページディレクトリインデックス 371 と組み合わせることによって、ページテーブルのエントリアドレスを含むページディレクトリを見つけ出すようにも構成され得る。例えば、プロセッサ 120 は、図 3 B に示すように、ルートアドレス 310 を仮想アドレス 370 のページディレクトリインデックス 371 と組み合わせることによって、ページディレクトリ 323 をを見つけ出すように構成され得る。つまり、プロセッサ 120 は、図 3 A 及び 3 B に示すように、ページディレクトリ 320 における開始位置としてページディレクトリ 321 を使用し、及びページディレクトリインデックス 371 をオフセットとして使用することによって、ページディレクトリ 323 をを見つけ出すように構成される。

30

40

【0062】

[071] ページディレクトリ 323 は、2 レベルページテーブル構造の次のレベルであるページテーブルエントリ中のある位置を指すエントリアドレスを含み得る。図 3 B に示すように、ページディレクトリ 323 は、2 レベルページテーブルの次のレベルのページテーブルエントリを指す 4 MB のエントリアドレス 323 - 1 を含む。プロセッサ 120 は、図 3 A に示すように、ページディレクトリ 323 のコンテンツを読み出すことによって、ページテーブルエントリ 360 のページテーブルエントリ 362 を指すエントリアド

50

レス 3 5 0 を取得するように構成され得る。

【 0 0 6 3 】

[072] プロセッサ 1 2 0 は、ページテーブルエントリ 3 6 2 のコンテンツを読み出すことによって、物理ページインデックスを取得するようにも構成され得る。図 3 B に示すように、ページテーブルエントリ 3 6 2 は、4 MB の物理ページインデックス 3 6 2 - 1、ページ属性 3 6 2 - 2、及び予約フィールド 3 6 2 - 3 を含む。エントリアドレス 3 5 0 に従って、プロセッサ 1 2 0 は、図 3 A 及び 3 B に示すように、物理ページインデックス 3 9 1 となるページテーブルエントリ 3 6 2 の 4 MB の物理ページインデックス 3 6 2 - 1 を読み出すように構成され得る。

【 0 0 6 4 】

10

[073] プロセッサ 1 2 0 は、OS カーネルに関する物理アドレスとなるように、物理ページインデックス及びページオフセットを組み合わせるようにさらに構成され得る。例えば、プロセッサ 1 2 0 は、OS カーネルに関する物理アドレス 3 9 0 となるように、取得された物理ページインデックス 3 9 1 (すなわち、4 MB の物理ページインデックス 3 6 2 - 1) 及びページオフセット 3 9 3 (すなわち、ページオフセット 3 7 3) を組み合わせるように構成され得る。それに応じて、プロセッサ 1 2 0 は、変換された物理アドレス 3 9 0 でメモリ 1 1 0 にアクセスすることにより、OS カーネルに関する必要とされるデータ又は命令を取得するように構成され得る。

【 0 0 6 5 】

20

[074] 図 3 B を参照すると、ページテーブルエントリ 3 6 2 は、物理ページインデックス 3 6 2 - 1 及びページ属性 3 6 2 - 2 を含む。プロセッサ 1 2 0 は、上記のように仮想アドレスを物理アドレスに変換するために物理ページインデックス 3 6 2 - 1 を使用するように構成され得る。物理ページインデックス 3 6 2 - 1 は、例えば、32 ビットコンピュータシステムのカーネル空間において 4 MB のページをアドレス指定するためのページテーブルエントリ 3 6 2 の最上位ビット (MSB) の 10 ビットを含み得る。

【 0 0 6 6 】

30

[075] ページ属性 3 6 2 - 2 は、例えば、32 ビットコンピュータシステムにおけるページテーブルエントリ 3 6 2 の最下位ビット (LSB) の 12 ビットを含み得る。ページ属性 3 6 2 - 2 のこれらの 12 ビットは、変換された物理アドレスのページの属性を示すように構成され得る。例えば、ページ属性 3 6 2 - 2 のビット 7 は、カーネル空間のページサイズを示すように構成され得る。例えば、上記のように、ページ属性 3 6 2 - 2 のビット 7 が「0」である場合、カーネル空間のページサイズは、4 MB となるように構成され得る。ページ属性 3 6 2 - 2 のビット 7 が「0」である場合、カーネル空間のページサイズは、4 KB となるように構成され得る。プロセッサ 1 2 0 は、上記のように、ページ属性 3 6 2 - 2 のビット 7 に応じて、仮想アドレスを物理アドレスに変換するように構成され得る。

【 0 0 6 7 】

40

[076] 3 種類以上のページサイズが必要とされる場合、ページ属性 3 6 2 - 2 は、カーネル空間のページサイズを示す 2 ビット (例えば、ビット 9 及びビット 7) を含み得る。図面では、未使用と示されているが、このような使用のために、ページ属性 3 6 2 - 2 のビット 9 がさらに構成され得る。例えば、ページ属性 3 6 2 - 2 のビット 9 及びビット 7 が「11」である場合、カーネル空間のページサイズは、4 KB でもよい。ページ属性 3 6 2 - 2 のビット 9 及びビット 7 が「10」である場合、カーネル空間のページサイズは、4 MB でもよい。ページ属性 3 6 2 - 2 のビット 9 及びビット 7 が「01」である場合、カーネル空間のページサイズは、16 MB でもよい。ページ属性 3 6 2 - 2 のビット 9 及びビット 7 が「00」である場合、カーネル空間のページサイズは、64 MB でもよい。プロセッサ 1 2 0 は、上記に類似したページ属性 3 6 2 - 2 のビット 9 及びビット 7 に応じて仮想アドレスを物理アドレスに変換するように構成され得る。

【 0 0 6 8 】

50

[077] また、ページ属性 3 6 2 - 2 は、変換された物理アドレスのページが実行不可

であることを示すNXインジケータ（例えば、ビット2）を含み得る。例えば、ページ属性362-2のビット2が「1」である場合、変換された物理アドレス390のページは、実行可能でなくてもよい。プロセッサ120は、上記のようにページ属性362-2のビット2に応じて、変換された物理アドレス390のページを実行しないように、又はそのページにアクセスしないように構成され得る。

【0069】

[078] 代替的に、カーネル空間のページサイズを示すビットは、ページディレクトリ又はページテーブルエントリの1つに含まれてもよい。例えば、ページサイズビットが、図3Bのページディレクトリ323の予約フィールド323-2又はページテーブルエントリ362の予約フィールド362-2に含まれ得る。例えば、予約フィールド323-2のビット7が、ページサイズビットとして使用され得る。プロセッサ120は、予約フィールド323-2のビット7を読み出し、それに応じてカーネル空間のページサイズを取得するように構成され得る。

【0070】

[079] ページサイズを示すビットに類似して、NXインジケータは、ページディレクトリ又はページテーブルエントリの1つに含まれてもよい。例えば、NXビットは、図3Bのページディレクトリ323の予約フィールド323-2又はページテーブルエントリ362の予約フィールド362-2に含まれ得る。それに応じて、プロセッサ120は、変換された物理アドレスのページに関するNXビットを読み出すように構成され得る。実行不可インジケータは、OSカーネルが、変換された物理アドレスのページにおいて悪意のあるコード又はウイルスを実行することを阻止する助けとなり得る。

【0071】

[080] 幾つかの実施形態では、プロセッサ120は、仮想アドレスを物理アドレスに直接変換するように構成され得る。つまり、仮想アドレスは、物理アドレスと同一である。例えば、図3Aの仮想アドレス370は、図に示されるようにページディレクトリ320及びページテーブルエントリ360を通して変換されることなく、直接物理アドレス390となるように変換され得る。仮想アドレスから物理アドレスへの直接変換は、OSカーネルがメモリ110に素早く且つ効率的にアクセスする助けとなり得る。

【0072】

[081] ユーザプロセスに関して、プロセッサ120は、制御レジスタ124に保存されたルートページテーブルインデックスに従って、仮想アドレスを物理アドレスに変換するように構成され得る。例えば、プロセッサ120は、制御レジスタ124の保存されたルートページテーブルインデックスに従って、アプリケーションプログラムに関して、32ビットの仮想アドレスをメモリ110の32ビットの物理アドレスに変換するように構成され得る。

【0073】

[082] 図4A及び4Bは、本開示の幾つかの実施形態による、ユーザ空間のメモリ管理の例示的方法、及び例示的な対応するページマッピングの模式図である。ユーザプロセスに関する仮想アドレスは、ページディレクトリインデックス、ページテーブルインデックス、及びページオフセットを含み得る。例えば、図4Aに示すように、32ビットのユーザプロセスに関する仮想アドレス470は、ページディレクトリインデックス471、ページテーブルインデックス472、及びページオフセット473を含み得る。

【0074】

[083] ページディレクトリインデックス471は、ページディレクトリのインデックスでよく、ページテーブルのベースアドレスを含み得るページディレクトリを指すルートアドレスと共に使用され得る。図4Aでは、制御レジスタ124が、例えば、ユーザプロセスに関するルートページテーブルインデックスをルートアドレス410として保存し得る。ページディレクトリインデックス471は、ページテーブル440のベースアドレス430を含むページディレクトリ(PD)423を指すルートアドレス410と共に使用され得る。

10

20

30

40

50

## 【 0 0 7 5 】

[084] ページテーブルインデックス 4 7 2 は、ページテーブルのインデックスでよく、ページテーブルエントリのエントリアドレスを含み得るページテーブル ( P T ) を指すベースアドレス 4 3 0 と共に使用され得る。例えば、ページテーブルインデックス 4 7 2 は、図 4 A に示すように、ページテーブルエントリ 4 6 0 のエントリアドレス 4 5 0 を含み得るページテーブル 4 4 2 を指すベースアドレス 4 3 0 と共に使用され得る。

## 【 0 0 7 6 】

[085] ページオフセット 4 7 3 は、ページのあるオフセットでよく、変換された物理アドレスにおいてページオフセットとして使用され得る。例えば、仮想アドレス 4 7 0 のページオフセット 4 7 3 は、図 4 A に示すように、物理アドレス 4 9 0 のページオフセット 4 9 3 としてそのまま使用され得る。

10

## 【 0 0 7 7 】

[086] ユーザプロセスに関して仮想アドレスを物理アドレスに変換するために、プロセッサ 1 2 0 は、記憶ユニットにユーザプロセスのルートページテーブルインデックスを保存するように構成され得る。例えば、プロセッサ 1 2 0 は、Linux カーネルに関するルートページテーブルインデックスを制御レジスタ 1 2 4 に保存するように構成され得る。プロセッサ 1 2 0 は、図 4 A に示すように、ページディレクトリ 4 2 0 において、ある位置 (例えばページディレクトリ 4 2 1) を指すルートアドレス 4 1 0 として、保存されたルートページテーブルインデックスを使用するように構成され得る。プロセッサ 1 2 0 は、制御レジスタ 1 2 4 に保存されたルートページテーブルインデックスをページディレクトリインデックス 4 7 1 と組み合わせることによって、ページテーブルのベースアドレスを含み得るページディレクトリを見つけ出すようにも構成され得る。例えば、プロセッサ 1 2 0 は、図 4 B に示すように、ルートアドレス 4 1 0 を仮想アドレス 4 7 0 のページディレクトリインデックス 4 7 1 と組み合わせることによって、ページディレクトリ 4 2 3 を見つけ出すように構成され得る。つまり、プロセッサ 1 2 0 は、図 4 A 及び 4 B に示すように、ページディレクトリ 4 2 0 における開始位置としてページディレクトリ 4 2 1 を使用し、及びページディレクトリインデックス 4 7 1 をオフセットとして使用することによって、ページディレクトリ 4 2 3 を見つけ出すように構成される。

20

## 【 0 0 7 8 】

[087] ページディレクトリ 4 2 3 は、3 レベルページテーブル構造の次のレベルであるページテーブルのベースアドレスを指すベースアドレスを含み得る。図 4 B に示すように、ページディレクトリ 4 2 3 は、3 レベルページテーブルの次のレベルのページテーブルを指す 4 K B のベースアドレス 4 2 3 - 1 を含む。プロセッサ 1 2 0 は、図 4 A に示すように、ページディレクトリ 4 2 3 のコンテンツを読み出すことによって、ページテーブル 4 4 0 のページテーブル 4 4 1 を指すベースアドレス 4 3 0 を取得するようにさらに構成され得る。

30

## 【 0 0 7 9 】

[088] プロセッサ 1 2 0 は、見つけ出されたページディレクトリ 4 2 3 に保存されたベースアドレスをページテーブルインデックス 4 7 2 と組み合わせることによって、ページテーブルエントリのエントリアドレスを含むページテーブルを見つけ出すようにも構成され得る。例えば、プロセッサ 1 2 0 は、図 4 B に示すように、ベースアドレス 4 3 0 (すなわち、4 K B のベースアドレス 4 2 3 - 1) を仮想アドレス 4 7 0 のページテーブルインデックス 4 7 2 と組み合わせることによって、ページテーブル 4 4 2 を見つけ出すように構成され得る。つまり、プロセッサ 1 2 0 は、図 4 A 及び 4 B に示すように、ページテーブル 4 4 0 における開始位置としてページテーブル 4 4 1 を使用し、及びページテーブルインデックス 4 7 2 をオフセットとして使用することによって、ページテーブル 4 4 2 を見つけ出すように構成される。

40

## 【 0 0 8 0 】

[089] ページテーブル 4 4 2 は、3 レベルページテーブル構造における次のレベルであるページテーブルエントリ中のある位置を指すエントリアドレスを含み得る。図 4 B に

50

示すように、ページテーブル 442 は、3 レベルページテーブルの次のレベルのページテーブルエントリを指す 4 KB のエントリアドレス 442 - 1 を含む。プロセッサ 120 は、図 4 A に示すように、ページテーブル 442 のコンテンツを読み出すことによって、ページテーブルエントリ 460 のページテーブルエントリ 462 を指すエントリアドレス 450 を取得するように構成され得る。

【0081】

[090] プロセッサ 120 は、ページテーブルエントリ 462 のコンテンツを読み出すことによって、物理ページインデックスを取得するようにさらに構成され得る。図 4 B に示すように、ページテーブルエントリ 462 は、4 KB の物理ページインデックス 462 - 1 及びページ属性 462 - 2 を含む。エントリアドレス 450 に従って、プロセッサ 120 は、図 4 A 及び 4 B に示すように、物理ページインデックス 491 となるページテーブルエントリ 462 の 4 KB の物理ページインデックス 462 - 1 を読み出すように構成され得る。

10

【0082】

[091] プロセッサ 120 は、ユーザプロセスに関する物理アドレスとなるように、物理ページインデックス及びページオフセットを組み合わせるようにさらに構成され得る。例えば、プロセッサ 120 は、ユーザプロセスに関する物理アドレス 490 となるように、取得された物理ページインデックス 491 (すなわち、4 KB の物理ページインデックス 462 - 1) 及びページオフセット 493 (すなわち、ページオフセット 473) を組み合わせるように構成され得る。それに応じて、プロセッサ 120 は、変換された物理アドレス 490 でメモリ 110 にアクセスすることにより、ユーザプロセスに関する必要とされるデータ又は命令を取得するように構成され得る。

20

【0083】

[092] 幾つかの実施形態では、プロセッサ 120 は、ユーザ空間のページサイズ又はユーザ空間ページサイズを示すページサイズインジケータを取得するように構成され得る。図 4 B に示すように、ページテーブルエントリ 462 は、ページ属性 462 - 2 を含む。プロセッサ 120 は、ページテーブルエントリ 462 におけるページ属性 462 - 2 を読み出すように構成され得る。ページ属性 462 - 2 は、ユーザ空間のページサイズを示すビット (例えば、ビット 7) を含み得る。例えば、ページ属性 462 - 2 のビット 7 が「1」である場合、ユーザ空間のページサイズは、4 KB でもよい。ページ属性 462 - 2 のビット 7 が「0」である場合、ユーザ空間のページサイズは、4 MB でもよい。

30

【0084】

[093] ユーザ空間が第 1 のページサイズを含むことをページサイズインジケータが示す場合、プロセッサ 120 は、上記のように、仮想アドレスを物理アドレスに変換するように構成され得る。例えば、ページ属性 462 - 2 のビット 7 が、ユーザ空間の 4 KB のページサイズを示す「1」である場合、プロセッサ 120 は、上記のように、並びに図 4 A 及び 4 B に示すように、仮想アドレス 470 を物理アドレス 490 に変換するように構成され得る。ページ属性 462 - 2 のビット 7 が、ユーザ空間の 4 MB のページサイズを示す「0」である場合、プロセッサ 120 は、以下に記載するように、並びに図 5 A 及び 5 B に示すように、仮想アドレス 570 を物理アドレス 590 に変換するように構成され得る。

40

【0085】

[094] カーネル空間のページサイズを示すビットは、ページディレクトリ又はページテーブルの 1 つに含まれてもよい。例えば、ページサイズビットは、図 4 B のページディレクトリ 423 の予約フィールド 423 - 2 又はページテーブル 442 の予約フィールド 442 - 2 に含まれ得る。例えば、予約フィールド 442 - 2 のビット 7 が、ページサイズビットとして使用され得る。プロセッサ 120 は、予約フィールド 442 - 2 のビット 7 を読み出し、それに応じてユーザ空間のページサイズを取得するように構成され得る。

【0086】

[095] 図 4 B を参照すると、ページテーブルエントリ 462 は、物理ページインデッ

50

クス 4 6 2 - 1 及びページ属性 4 6 2 - 2 を含む。プロセッサ 1 2 0 は、上記のように仮想アドレスを物理アドレスに変換するために物理ページインデックス 4 6 2 - 1 を使用するように構成され得る。物理ページインデックス 4 6 2 - 1 は、例えば、3 2 ビットコンピュータシステムのユーザ空間において 4 K B のページをアドレス指定するためのページテーブルエントリ 4 6 2 の最上位ビット ( M S B ) の 2 0 ビットを含み得る。

【 0 0 8 7 】

[096] ページ属性 4 6 2 - 2 は、例えば、3 2 ビットコンピュータシステムにおけるページテーブルエントリ 4 6 2 の最下位ビット ( L S B ) の 1 2 ビットを含み得る。ページ属性 4 6 2 - 2 のこれらの 1 2 ビットは、変換された物理アドレスのページの属性を示すように構成され得る。例えば、ページ属性 4 6 2 - 2 のビット 7 は、ユーザ空間のページサイズを示すように構成され得る。例えば、上記のように、ページ属性 4 6 2 - 2 のビット 7 が「 1 」である場合、ユーザ空間のページサイズは、4 K B となるように構成され得る。ページ属性 4 6 2 - 2 のビット 7 が「 0 」である場合、ユーザ空間のページサイズは、4 M B となるように構成され得る。プロセッサ 1 2 0 は、上記のように、ページ属性 4 6 2 - 2 のビット 7 に応じて、仮想アドレスを物理アドレスに変換するように構成され得る。

【 0 0 8 8 】

[097] 3 種類以上のページサイズが必要とされる場合、ページ属性 4 6 2 - 2 は、ユーザ空間のページサイズを示す 2 ビット (例えば、ビット 9 及びビット 7 ) を含み得る。図面では、未使用と示されているが、このような使用のために、ページ属性 4 6 2 - 2 のビット 9 がさらに構成され得る。例えば、ページ属性 4 6 2 - 2 のビット 9 及びビット 7 が「 1 1 」である場合、ユーザ空間のページサイズは、4 K B でもよい。ページ属性 4 6 2 - 2 のビット 9 及びビット 7 が「 1 0 」である場合、ユーザ空間のページサイズは、4 M B でもよい。ページ属性 4 6 2 - 2 のビット 9 及びビット 7 が「 0 1 」である場合、ユーザ空間のページサイズは、1 6 M B でもよい。ページ属性 4 6 2 - 2 のビット 9 及びビット 7 が「 0 0 」である場合、ユーザ空間のページサイズは、6 4 M B でもよい。プロセッサ 1 2 0 は、上記に類似したページ属性 4 6 2 - 2 のビット 9 及びビット 7 に応じて仮想アドレスを物理アドレスに変換するように構成され得る。

【 0 0 8 9 】

[098] また、ページ属性 4 6 2 - 2 は、変換された物理アドレスのページが実行不可であることを示す N X インジケータ (例えば、ビット 2 ) を含み得る。例えば、ページ属性 4 6 2 - 2 のビット 2 が「 1 」である場合、変換された物理アドレス 4 9 0 のページは、実行可能でなくてもよい。プロセッサ 1 2 0 は、上記のようにページ属性 4 6 2 - 2 のビット 2 に応じて、変換された物理アドレス 4 9 0 のページを実行しないように、又はそのページにアクセスしないように構成され得る。

【 0 0 9 0 】

[099] ユーザ空間のページサイズを示すビットは、ページディレクトリ又はページテーブルの 1 つに含まれてもよい。例えば、ページサイズビットが、図 4 B のページディレクトリ 4 2 3 の予約フィールド 4 2 3 - 2 又はページテーブル 4 4 2 の予約フィールド 4 4 2 - 2 に含まれ得る。例えば、予約フィールド 4 4 2 - 2 のビット 7 が、ページサイズビットとして使用され得る。プロセッサ 1 2 0 は、予約フィールド 4 4 2 - 2 のビット 7 を読み出し、それに応じてユーザ空間のページサイズを取得するように構成され得る。

【 0 0 9 1 】

[0100] ページサイズを示すビットに類似して、N X インジケータは、ページディレクトリ又はページテーブルの 1 つに含まれてもよい。例えば、N X ビットは、図 4 B のページディレクトリ 4 2 3 の予約フィールド 4 2 3 - 2 又はページテーブル 4 4 2 の予約フィールド 4 4 2 - 2 に含まれ得る。プロセッサ 1 2 0 は、それに応じて変換された物理アドレスに関する N X ビットからページに関する N X ビットを読み出すように構成され得る。

【 0 0 9 2 】

[0101] 幾つかの実施形態では、プロセッサ 1 2 0 は、仮想アドレスを物理アドレスに



直接変換するように構成され得る。つまり、仮想アドレスは、物理アドレスと同一である。例えば、図 4 A の仮想アドレス 4 7 0 は、図に示されるようにページディレクトリ 4 2 0、ページテーブル 4 4 0、及びページテーブルエントリ 4 6 0 を通して変換されることなく、直接物理アドレス 4 9 0 となるように変換され得る。仮想アドレスから物理アドレスへの直接変換は、ユーザプロセスがメモリ 1 1 0 に素早く且つ効率的にアクセスする助けとなり得る。

【 0 0 9 3 】

[0102] ユーザ空間が第 2 のページサイズを含むことをページサイズインジケータが示す場合、プロセッサ 1 2 0 は、図 5 A 及び 5 B に示すように、仮想アドレスを物理アドレスに変換するように構成され得る。例えば、図 5 B のページ属性 5 6 2 - 2 のビット 7 が、カーネル空間の 4 M B のページサイズを示す「 0 」である場合に、プロセッサ 1 2 0 は、図 5 A に示すように、ページディレクトリ 5 2 0 及びページテーブルエントリ 5 6 0 を含む 2 レベルページテーブルを通して仮想アドレス 5 7 0 を物理アドレス 5 9 0 に変換するように構成され得る。

10

【 0 0 9 4 】

[0103] 図 5 A 及び 5 B は、本開示の幾つかの実施形態による、ユーザ空間のメモリ管理の例示的方法、及び例示的な対応するページマッピングの模式図である。ユーザプロセスに関する仮想アドレスは、ページディレクトリインデックス及びページオフセットを含み得る。例えば、図 5 A に示すように、3 2 ビットのユーザプロセスに関する仮想アドレス 5 7 0 は、ページディレクトリインデックス 5 7 1 及びページオフセット 5 7 3 を含み得る。

20

【 0 0 9 5 】

[0104] ページディレクトリインデックス 5 7 1 は、ページディレクトリのインデックスでよく、ページテーブルエントリのエントリアドレスを含み得るページディレクトリを指すルートアドレスと共に使用され得る。図 5 A では、制御レジスタ 1 2 4 は、例えば、ルートアドレス 5 1 0 としてユーザプロセスに関するルートページテーブルインデックスを保存するように構成され得る。ページディレクトリインデックス 5 7 1 は、ページテーブルエントリ 5 6 0 のエントリアドレス 5 5 0 を含むページディレクトリ ( P D ) 5 2 3 を指すルートアドレス 5 1 0 と共に使用され得る。

【 0 0 9 6 】

30

[0105] ページオフセット 5 7 3 は、ページのオフセットでよく、変換された物理アドレスにおいてページオフセットとして使用され得る。例えば、仮想アドレス 5 7 0 のページオフセット 5 7 3 は、図 5 A に示すように、物理アドレス 5 9 0 のページオフセット 5 9 3 としてそのまま使用され得る。

【 0 0 9 7 】

[0106] ユーザプロセスに関して仮想アドレスを物理アドレスに変換するために、プロセッサ 1 2 0 は、記憶ユニットにユーザプロセスのルートページテーブルインデックスを保存するように構成され得る。例えば、プロセッサ 1 2 0 は、図 5 A において、Linux カーネルに関するルートページテーブルインデックスを制御レジスタ 1 2 4 に保存するように構成され得る。プロセッサ 1 2 0 は、図 5 A に示すように、ページディレクトリ 5 2 0 において、ある位置 (例えばページディレクトリ 5 2 1) を指すルートアドレス 5 1 0 として、保存されたルートページテーブルインデックスを使用するように構成され得る。プロセッサ 1 2 0 は、制御レジスタ 1 2 4 に保存されたルートページテーブルインデックスをページディレクトリインデックス 5 7 1 と組み合わせることによって、ページテーブルのエントリアドレスを含むページディレクトリを見つけ出すようにも構成され得る。例えば、プロセッサ 1 2 0 は、図 5 B に示すように、ルートアドレス 5 1 0 を仮想アドレス 5 7 0 のページディレクトリインデックス 5 7 1 と組み合わせることによって、ページディレクトリ 5 2 3 をを見つけ出すように構成され得る。つまり、プロセッサ 1 2 0 は、図 5 A 及び 5 B に示すように、ページディレクトリ 5 2 0 における開始位置としてページディレクトリ 5 2 1 を使用し、及びページディレクトリインデックス 5 7 1 をオフセットとして

40

50

使用することによって、ページディレクトリ 5 2 3 を見つけ出すように構成される。

【 0 0 9 8 】

[0107] ページディレクトリ 5 2 3 は、2 レベルページテーブル構造の次のレベルであるページテーブルエントリ中のある位置を指すエントリアドレスを含み得る。図 5 B に示すように、ページディレクトリ 5 2 3 は、2 レベルページテーブルの次のレベルのページテーブルエントリを指す 4 M B のエントリアドレス 5 2 3 - 1 を含む。プロセッサ 1 2 0 は、図 5 A に示すように、ページディレクトリ 5 2 3 のコンテンツを読み出すことによって、ページテーブルエントリ 5 6 0 のページテーブルエントリ 5 6 2 を指すエントリアドレス 5 5 0 を取得するように構成され得る。

【 0 0 9 9 】

[0108] プロセッサ 1 2 0 は、ページテーブルエントリ 5 6 2 のコンテンツを読み出すことによって、物理ページインデックスを取得するようにも構成され得る。図 5 B に示すように、ページテーブルエントリ 5 6 2 は、4 M B の物理ページインデックス 5 6 2 - 1、ページ属性 5 6 2 - 2、及び予約フィールド 5 6 2 - 3 を含む。エントリアドレス 5 5 0 に従って、プロセッサ 1 2 0 は、図 5 A 及び 5 B に示すように、物理ページインデックス 5 9 1 となるページテーブルエントリ 5 6 2 の 4 M B の物理ページインデックス 5 6 2 - 1 を読み出すように構成され得る。

【 0 1 0 0 】

[0109] プロセッサ 1 2 0 は、ユーザプロセスに関する物理アドレスとなるように、物理ページインデックス及びページオフセットを組み合わせるようにさらに構成され得る。例えば、プロセッサ 1 2 0 は、ユーザプロセスに関する物理アドレス 5 9 0 となるように、取得された物理ページインデックス 5 9 1 (すなわち、4 M B の物理ページインデックス 5 6 2 - 1) 及びページオフセット 5 9 3 (すなわち、ページオフセット 5 7 3) を組み合わせるように構成され得る。それに応じて、プロセッサ 1 2 0 は、変換された物理アドレス 5 9 0 でメモリ 1 1 0 にアクセスすることにより、ユーザプロセスに関する必要とされるデータ又は命令を取得するように構成され得る。

【 0 1 0 1 】

[0110] 図 5 B を参照すると、ページテーブルエントリ 5 6 2 は、物理ページインデックス 5 6 2 - 1 及びページ属性 5 6 2 - 2 を含む。プロセッサ 1 2 0 は、上記のように仮想アドレスを物理アドレスに変換するために物理ページインデックス 5 6 2 - 1 を使用するように構成され得る。物理ページインデックス 5 6 2 - 1 は、例えば、3 2 ビットコンピュータシステムのユーザ空間において 4 M B のページをアドレス指定するためのページテーブルエントリ 5 6 2 の最上位ビット (MSB) の 1 0 ビットを含み得る。

【 0 1 0 2 】

[0111] ページ属性 5 6 2 - 2 は、例えば、3 2 ビットコンピュータシステムにおけるページテーブルエントリ 5 6 2 の最下位ビット (LSB) の 1 2 ビットを含み得る。ページ属性 5 6 2 - 2 のこれらの 1 2 ビットは、変換された物理アドレスのページの属性を示すように構成され得る。例えば、ページ属性 5 6 2 - 2 のビット 7 は、ユーザ空間のページサイズを示すように構成され得る。例えば、上記のように、ページ属性 5 6 2 - 2 のビット 7 が「0」である場合、ユーザ空間のページサイズは、4 M B となるように構成され得る。ページ属性 5 6 2 - 2 のビット 7 が「0」である場合、ユーザ空間のページサイズは、4 K B となるように構成され得る。プロセッサ 1 2 0 は、上記のように、ページ属性 5 6 2 - 2 のビット 7 に応じて、仮想アドレスを物理アドレスに変換するように構成され得る。

【 0 1 0 3 】

[0112] 3 種類以上のページサイズが必要とされる場合、ページ属性 5 6 2 - 2 は、ユーザ空間のページサイズを示す 2 ビット (例えば、ビット 9 及びビット 7) を含み得る。図面では、未使用と示されているが、このような使用のために、ページ属性 5 6 2 - 2 のビット 9 がさらに構成され得る。例えば、ページ属性 5 6 2 - 2 のビット 9 及びビット 7 が「1 1」である場合、ユーザ空間のページサイズは、4 K B でもよい。ページ属性 5 6

10

20

30

40

50

2 - 2 のビット 9 及びビット 7 が「10」である場合、ユーザ空間のページサイズは、4 MB でもよい。ページ属性 562 - 2 のビット 9 及びビット 7 が「01」である場合、ユーザ空間のページサイズは、16 MB でもよい。ページ属性 562 - 2 のビット 9 及びビット 7 が「00」である場合、ユーザ空間のページサイズは、64 MB でもよい。プロセッサ 120 は、上記に類似したページ属性 562 - 2 のビット 9 及びビット 7 に応じて仮想アドレスを物理アドレスに変換するように構成され得る。

【0104】

[0113] また、ページ属性 562 - 2 は、変換された物理アドレスのページが実行不可であることを示す NX インジケータ（例えば、ビット 2）を含み得る。例えば、ページ属性 562 - 2 のビット 2 が「1」である場合、変換された物理アドレス 590 のページは、実行可能でなくてもよい。プロセッサ 120 は、上記のようにページ属性 562 - 2 のビット 2 に応じて、変換された物理アドレス 590 のページを実行しないように、又はそのページにアクセスしないように構成され得る。

【0105】

[0114] 代替的に、ユーザ空間のページサイズを示すビットは、ページディレクトリ又はページテーブルエントリの 1 つに含まれてもよい。例えば、ページサイズビットが、図 5B のページディレクトリ 523 の予約フィールド 523 - 2 又はページテーブルエントリ 562 の予約フィールド 562 - 2 に含まれ得る。例えば、予約フィールド 523 - 2 のビット 7 が、ページサイズビットとして構成され得る。プロセッサ 120 は、予約フィールド 523 - 2 のビット 7 を読み出し、それに応じてユーザ空間のページサイズを取得するように構成され得る。

【0106】

[0115] ページサイズを示すビットに類似して、NX インジケータは、ページディレクトリ又はページテーブルエントリの 1 つに含まれてもよい。例えば、NX ビットは、図 5B のページディレクトリ 523 の予約フィールド 523 - 2 又はページテーブルエントリ 562 の予約フィールド 562 - 2 に含まれ得る。それに応じて、プロセッサ 120 は、変換された物理アドレスのページの NX インジケータを読み出すように構成され得る。実行不可インジケータは、OS カーネルが、変換された物理アドレスのページにおいて悪意のあるコード又はウイルスを実行することを阻止する助けとなり得る。

【0107】

[0116] 幾つかの実施形態では、プロセッサ 120 は、仮想アドレスを物理アドレスに直接変換するように構成され得る。つまり、仮想アドレスは、物理アドレスと同一である。例えば、図 5A の仮想アドレス 570 は、図に示されるようにページディレクトリ 520 及びページテーブルエントリ 560 を通して変換されることなく、直接物理アドレス 590 となるように変換され得る。仮想アドレスから物理アドレスへの直接変換は、ユーザプロセスがメモリ 110 に素早く且つ効率的にアクセスする助けとなり得る。

【0108】

[0117] 幾つかの実施形態では、プロセッサ 120 は、OS カーネルのユーザ空間へのアクセスの許容を示すユーザ空間アクセスインジケータを取得するように構成され得る。例えば、プロセッサ 120 は、その命令セットにおいて、ユーザ空間にアクセスするために使用される接頭語「US」を含む 1 つ又は複数の命令を含み得る。接頭語「US」を含む命令は、ユーザ空間アクセスインジケータとして使用され得る。「US」の接頭語を有する命令が実行され得る場合、プロセッサ 120 は、ユーザ空間アクセスインジケーションを取得するように構成され得る。図 2A を参照すると、プロセッサ 120 は、仮想アドレス 270 を使用してユーザ空間にアクセスするように構成され得る。

【0109】

[0118] プロセッサ 120 は、ユーザ空間に関するルートページテーブルインデックス及び仮想アドレスのページディレクトリインデックスに従って、ユーザ空間に関するベースアドレスを取得するように構成され得る。例えば、図 2A 及び 4A を参照すると、プロセッサ 120 は、制御レジスタ 124 に保存されたルートページテーブルインデックスを

ページディレクトリインデックス 271 と組み合わせることによって、ページテーブルのベースアドレスを含み得るページディレクトリを見つけ出すように構成され得る。プロセッサ 120 は、ルートアドレス 410 を仮想アドレス 270 のページディレクトリインデックス 271 と組み合わせることによって、ページディレクトリ 423 をを見つけ出すように構成され得る。つまり、プロセッサ 120 は、ページディレクトリ 420 における開始位置としてページディレクトリ 421 を使用し、及びページディレクトリインデックス 271 をオフセットとして使用することによって、ページディレクトリ 423 をを見つけ出すように構成される。プロセッサ 120 は、ページディレクトリ 423 のコンテンツを読み出すことによって、ページテーブル 440 のページテーブル 441 を指すベースアドレス 430 を取得するようにさらに構成され得る。

10

#### 【0110】

[0119] プロセッサ 120 は、見つけ出されたページディレクトリ 423 に保存されたベースアドレスをページテーブルインデックス 272 と組み合わせることによって、ページテーブルエントリのエントリアドレスを含むページテーブルを見つけ出すようにも構成され得る。例えば、プロセッサ 120 は、ベースアドレス 430 (すなわち、4KB のベースアドレス 423 - 1) を仮想アドレス 270 のページテーブルインデックス 272 と組み合わせることによって、ページテーブル 442 を見つけ出すように構成され得る。つまり、プロセッサ 120 は、ページテーブル 440 における開始位置としてページテーブル 441 を使用し、及びページテーブルインデックス 272 をオフセットとして使用することによって、ページテーブル 442 を見つけ出すように構成される。プロセッサ 120 は、ページテーブル 442 のコンテンツを読み出すことによって、ページテーブルエントリ 460 のページテーブルエントリ 462 を指すエントリアドレス 450 を取得するようにさらに構成され得る。

20

#### 【0111】

[0120] プロセッサ 120 は、ページテーブルエントリ 462 のコンテンツを読み出すことによって、物理ページインデックスを取得するようにも構成され得る。エントリアドレス 450 に従って、プロセッサ 120 は、物理ページインデックス 491 となるページテーブルエントリ 462 の 4KB の物理ページインデックス 462 - 1 を読み出すように構成され得る。

#### 【0112】

[0121] プロセッサ 120 は、ユーザプロセスに関する物理アドレスとなるように、物理ページインデックス及びページオフセットを組み合わせるようにさらに構成され得る。例えば、プロセッサ 120 は、OS カーネルがユーザ空間にアクセスする物理アドレスとなるように、取得された物理ページインデックス 491 (すなわち、4KB の物理ページインデックス 462 - 1) 及びページオフセット 273 を組み合わせるように構成され得る。それに応じて、プロセッサ 120 は、変換された物理アドレスでメモリ 110 にアクセスすることにより、ユーザ空間から OS カーネルに関する必要とされるデータ又は命令を取得するように構成され得る。

30

#### 【0113】

[0122] 幾つかの実施形態では、プロセッサ 120 は、ユーザプロセスのユーザ空間へのアクセスの許容を示すユーザ空間アクセスインジケータを取得するように構成され得る。例えば、プロセッサ 120 は、その命令セットにおいて、ユーザ空間にアクセスするために使用される接頭語「US」を含む 1 つ又は複数の命令を含み得る。接頭語「US」を含む命令は、ユーザ空間アクセスインジケータとして使用され得る。「US」の接頭語を有する命令が実行される場合、「US」の接頭語を有する命令が、OS カーネルだけのものであるので、プロセッサ 120 は、一般保護違反を設定するように構成され得る。例えば、プロセッサ 120 は、警告メッセージ又はフラグを設定し、ユーザプロセスに関して「US」の接頭語を有する命令を実行しないようにし得る。

40

#### 【0114】

[0123] 図 6 は、本開示の幾つかの実施形態による例示的メモリ管理方法 600 のフロ

50

ーチャートである。方法 6 0 0 は、仮想アドレスを獲得すること（ステップ 6 1 0）、カーネル空間又はユーザ空間にアクセスすることを決定すること（ステップ 6 2 0）、カーネル空間に対応する第 1 のルートページテーブルインデックスを取得すること（ステップ 6 3 1）、ユーザ空間に対応する第 2 のルートページテーブルインデックスを取得すること（ステップ 6 4 1）、オペレーティングシステムカーネルのために第 1 のルートページテーブルインデックスに従って第 1 の仮想アドレスを第 1 の物理アドレスに変換すること（ステップ 6 3 2）、及びユーザプロセスのために第 2 のルートページテーブルインデックスに従って第 2 の仮想アドレスを第 2 の物理アドレスに変換すること（ステップ 6 4 2）を含む。方法 6 0 0 は、カーネル空間又はユーザ空間のページサイズを示すページサイズインジケータを取得することも含み得る。方法 6 0 0 は、変換された物理アドレスのページに関する実行不可インジケータを取得することも含み得る。方法 6 0 0 は、OS カーネル又はユーザプロセスに関するユーザ空間アクセスインジケーションを取得することをさらに含み得る。

10

**【 0 1 1 5 】**

[0124] ステップ 6 1 0 は、仮想アドレスを獲得することを含む。例えば、ステップ 6 1 0 の仮想アドレスを獲得することは、命令の復号後に仮想アドレスを取得することを含み得る。別の例として、ステップ 6 1 0 の仮想アドレスを獲得することは、仮想 - 物理アドレスマッピングのために OS カーネル又はユーザプロセスから仮想アドレスを取得することを含み得る。

**【 0 1 1 6 】**

20

[0125] ステップ 6 2 0 は、カーネル空間又はユーザ空間にアクセスすることを決定することを含む。例えば、ステップ 6 1 0 において OS カーネルから仮想アドレスを取得した後に、ステップ 6 2 0 においてカーネル空間又はユーザ空間にアクセスすることを決定することは、デフォルト結果としてカーネル空間にアクセスすることを決定することを含み得る。別の例として、ステップ 6 1 0 においてユーザプロセスから仮想アドレスを取得した後に、ステップ 6 2 0 においてカーネル空間又はユーザ空間にアクセスすることを決定することは、デフォルト結果としてユーザ空間にアクセスすることを決定することを含み得る。

**【 0 1 1 7 】**

[0126] 幾つかの実施形態では、方法 6 0 0 は、OS カーネルに関するユーザ空間アクセスインジケーションを取得することを含み得る。OS カーネルからユーザ空間アクセスインジケーションを取得した後に、ステップ 6 2 0 においてカーネル空間又はユーザ空間にアクセスすることを決定することは、ユーザ空間にアクセスすることを決定することを含み得る。

30

**【 0 1 1 8 】**

[0127] ステップ 6 3 1 は、カーネル空間に対応するルートページテーブルインデックスを取得することを含む。例えば、ステップ 6 3 1 におけるカーネル空間に対応するルートページテーブルインデックスを取得することは、上記で例示したように、制御レジスタ 3（CR3）、OS カーネルに関する制御レジスタ 3（CR3K）などの記憶ユニット、又はキャッシュ、メインメモリ、若しくはストレージデバイスの記憶空間からルートページテーブルインデックスを取得することを含み得る。

40

**【 0 1 1 9 】**

[0128] 幾つかの実施形態では、ステップ 6 3 1 においてカーネル空間に対応するルートページテーブルインデックスを取得することは、記憶ユニットから OS カーネルに対応するルートページテーブルインデックスを読み出すこと、及びそれをプロセッサの専用制御レジスタに保存することも含み得る。例えば、ステップ 6 3 1 においてカーネル空間に対応するルートページテーブルインデックスを取得することは、OS カーネルに関するスタックから OS カーネルに対応するルートページテーブルインデックスをポップすること、及びそれを CR3K に保存することを含み得る。別の例として、ステップ 6 3 1 においてカーネル空間に対応するルートページテーブルインデックスを取得することは、ストレ

50

ージ 130 から OS カーネルのルートページテーブルインデックスを含み得るデータのスワッピングを行うこと、及びそれをプロセッサ 120 の制御レジスタ 122 に保存することを含み得る。

【0120】

[0129] ステップ 632 は、OS カーネルに関するルートページテーブルインデックスに従って仮想アドレスを物理アドレスに変換することを含む。例えば、ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、図 2A 及び 2B に示すように、並びに上記のように、制御レジスタ 122 のルートページテーブルインデックスに従って、OS カーネルに関して仮想アドレス 270 を物理アドレス 290 に変換することを含み得る。別の例として、ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、図 3A 及び 3B に示すように、並びに上記のように、OS カーネルに関する制御レジスタ 122 に保存されたルートページテーブルインデックスに従って、仮想アドレス 370 を物理アドレス 390 に変換することを含み得る。

10

【0121】

[0130] 幾つかの実施形態では、ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、OS カーネルのルートページテーブルインデックス及び仮想アドレスのページディレクトリインデックスに従ってベースアドレスを取得することを含み得る。例えば、ステップ 632 においてベースアドレスを取得することは、図 2A 及び 2B に示すように、ルートアドレス 210 を仮想アドレス 270 のページディレクトリインデックス 271 と組み合わせることによって、ページディレクトリ 223 を見つけ出すことを含み得る。ページディレクトリ 223 は、ページテーブル 240 のベースアドレスを含み得る。

20

【0122】

[0131] ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、ベースアドレス及びページテーブルインデックスに従ってエントリアドレスを取得することも含み得る。例えば、ステップ 632 においてエントリアドレスを取得することは、図 2A 及び 2B に示すように、ベースアドレス 230 を仮想アドレス 270 のページテーブルインデックス 272 と組み合わせることによって、ページテーブル 242 を見つけ出すことを含み得る。ページテーブル 242 は、ページテーブルエントリ中のある位置を指すエントリアドレスを含み得る。ステップ 632 においてエントリアドレスを取得することは、ページテーブル 242 のコンテンツを読み出すことによって、ページテーブルエントリ 260 のページテーブルエントリ 262 を指すエントリアドレス 250 を取得することも含み得る。

30

【0123】

[0132] ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、エントリアドレスに従って物理ページインデックスを取得することをさらに含み得る。例えば、ステップ 632 において物理ページインデックスを取得することは、ページテーブルエントリ 262 のコンテンツを読み出すことによって、物理ページインデックス 291 を取得することを含み得る。

40

【0124】

[0133] ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、物理アドレスとなるように物理ページインデックス及びページオフセットを組み合わせることをさらに含み得る。例えば、ステップ 632 において物理ページインデックス及びページオフセットを組み合わせることは、図 2A に示すように、物理アドレス 290 となるように、物理ページインデックス 291 及びページオフセット 293 を組み合わせることを含み得る。ページオフセット 293 は、仮想アドレス 270 のページオフセット 273 と同一になり得る。

【0125】

[0134] ステップ 632 において OS カーネルに関して仮想アドレスを物理アドレスに

50

変換することは、仮想アドレスを物理アドレスに直接変換することも含み得る。つまり、仮想アドレスは、物理アドレスと同一である。例えば、図 2 A の仮想アドレス 2 7 0 は、図に示されるようにページディレクトリ 2 2 0、ページテーブル 2 4 0、及びページテーブルエントリ 2 6 0 を通して変換されることなく、直接物理アドレス 2 9 0 となるように変換され得る。

【 0 1 2 6 】

[0135] 別の例として、ステップ 6 3 2 において OS カーネルに関して仮想アドレスを物理アドレスに変換することは、図 3 A 及び 3 B に示すように、並びに上記のように、仮想アドレス 3 7 0 を物理アドレス 3 9 0 に変換することを含み得る。

【 0 1 2 7 】

[0136] ステップ 6 4 1 は、ユーザ空間に対応するルートページテーブルインデックスを取得することを含む。例えば、ステップ 6 4 1 におけるユーザ空間に対応するルートページテーブルインデックスを取得することは、上記で例示したように、制御レジスタ 4 (CR 4)、ユーザプロセスに関する制御レジスタ 3 (CR 3 U) などの記憶ユニット、又はキャッシュ、メインメモリ、若しくはストレージデバイスの記憶空間からルートページテーブルインデックスを取得することを含み得る。

【 0 1 2 8 】

[0137] 幾つかの実施形態では、ステップ 6 4 1 におけるユーザ空間に対応するルートページテーブルインデックスを取得することは、記憶ユニットからユーザプロセスに対応するルートページテーブルインデックスを読み出すこと、及びそれをプロセッサの専用制御レジスタに保存することも含み得る。例えば、ステップ 6 4 1 においてユーザ空間に対応するルートページテーブルインデックスを取得することは、ユーザプロセスに関するスタックからユーザプロセスに対応するルートページテーブルインデックスをポップすること、及びそれを CR 3 U に保存することを含み得る。別の例として、ステップ 6 4 1 においてユーザ空間に対応するルートページテーブルインデックスを取得することは、ストレージ 1 3 0 からユーザプロセスのルートページテーブルインデックスを含み得るデータのスワッピングを行うこと、及びそれをプロセッサ 1 2 0 の制御レジスタ 1 2 4 に保存することを含み得る。

【 0 1 2 9 】

[0138] ステップ 6 4 2 は、ユーザプロセスに関するルートページテーブルインデックスに従って仮想アドレスを物理アドレスに変換することを含む。例えば、ステップ 6 4 1 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、図 4 A 及び 4 B に示すように、並びに上記のように、制御レジスタ 1 2 4 に保存されたユーザプロセスのルートページテーブルインデックスに従って、仮想アドレス 4 7 0 を物理アドレス 4 9 0 に変換することを含み得る。別の例として、ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、図 5 A 及び 5 B に示すように、並びに上記のように、ユーザプロセスに関する制御レジスタ 1 2 4 に保存されたルートページテーブルインデックスに従って、仮想アドレス 5 7 0 を物理アドレス 5 9 0 に変換することを含み得る。

【 0 1 3 0 】

[0139] ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、ユーザプロセスのルートページテーブルインデックス及び仮想アドレスのページディレクトリインデックスに従ってベースアドレスを取得することを含み得る。例えば、図 4 A 及び 4 B に示すように、ステップ 6 4 2 においてベースアドレスを取得することは、ルートアドレス 4 1 0 を仮想アドレス 4 7 0 のページディレクトリインデックス 4 7 1 と組み合わせることによって、ページディレクトリ 4 2 3 を見つけ出すことを含み得る。ページディレクトリ 4 2 3 は、ページテーブル 4 4 0 のベースアドレスを含み得る。

【 0 1 3 1 】

[0140] ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレス

10

20

30

40

50

に変換することは、ベースアドレス及びページテーブルインデックスに従ってエントリアドレスを取得することを含み得る。例えば、ステップ 6 4 2 においてエントリアドレスを取得することは、図 4 A 及び 4 B に示すように、ベースアドレス 4 3 0 を仮想アドレス 4 7 0 のページテーブルインデックス 4 7 2 と組み合わせることによって、ページテーブル 4 4 2 を見つけ出すことを含み得る。ページテーブル 4 4 2 は、ページテーブルエントリ中のある位置を指すエントリアドレスを含み得る。ステップ 6 4 2 においてエントリアドレスを取得することは、ページテーブル 4 4 2 のコンテンツを読み出すことによって、ページテーブルエントリ 4 6 0 中のページテーブルエントリ 4 6 2 を指すエントリアドレス 4 5 0 を取得することを含み得る。

【 0 1 3 2 】

[0141] 幾つかの実施形態では、ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、エントリアドレスに従って物理ページインデックスを取得することをさらに含み得る。例えば、ステップ 6 4 2 において物理ページインデックスを取得することは、ページテーブルエントリ 4 6 2 のコンテンツを読み出すことによって、物理ページインデックス 4 9 1 を取得することを含み得る。

【 0 1 3 3 】

[0142] 幾つかの実施形態では、ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、物理アドレスとなるように、物理ページインデックス及びページオフセットを組み合わせることをさらに含み得る。例えば、ステップ 6 4 2 において物理ページインデックス及びページオフセットを組み合わせることは、図 4 A に示すように、物理アドレス 4 9 0 となるように、物理ページインデックス 4 9 1 及びページオフセット 4 9 2 を組み合わせることを含み得る。ページオフセット 4 9 2 は、仮想アドレス 4 7 0 のページオフセット 4 7 1 と同一になり得る。

【 0 1 3 4 】

[0143] 代替的に、ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、仮想アドレスを物理アドレスに直接変換することを含み得る。つまり、仮想アドレスは、物理アドレスと同一である。例えば、図 4 A の仮想アドレス 4 7 0 は、図に示されるようにページディレクトリ 4 2 0、ページテーブル 4 4 0、及びページテーブルエントリ 4 6 0 を通して変換されることなく、直接物理アドレス 4 9 0 となるように変換され得る。

【 0 1 3 5 】

[0144] 別の例として、ステップ 6 4 2 においてユーザプロセスに関して仮想アドレスを物理アドレスに変換することは、図 5 A 及び 5 B に示すように、並びに上記のように、仮想アドレス 5 7 0 を物理アドレス 5 9 0 に変換することを含み得る。

【 0 1 3 6 】

[0145] 方法 6 0 0 は、カーネル空間又はユーザ空間のページサイズを示すページサイズインジケータを取得することを含み得る。例えば、方法 6 0 0 のページサイズインジケータを取得することは、図 2 B に示すように、及び上記のように、ページテーブルエントリ 2 6 2 のページ属性 2 6 2 - 2 のビット 7 を読み出すことを含み得る。ページ属性 2 6 2 - 2 のビット 7 は、カーネル空間のページサイズを示す。例えば、ページ属性 2 6 2 - 2 のビット 7 が「 1 」である場合、カーネル空間のページサイズは、4 K B でもよい。ページ属性 2 6 2 - 2 のビット 7 が「 0 」である場合、カーネル空間のページサイズは、4 M B でもよい。

【 0 1 3 7 】

[0146] カーネル空間が第 1 のページサイズを含み得ることを取得されたページサイズインジケータが示し得る場合、ステップ 6 3 2 における仮想アドレスを物理アドレスに変換することは、図 2 A 及び 2 B に示すように、並びに上記のように、仮想アドレスを物理アドレスに変換することを含み得る。例えば、ページ属性 2 6 2 - 2 のビット 7 が、カーネル空間の 4 K B のページサイズを示す「 1 」である場合、ステップ 6 3 2 における仮想アドレスを物理アドレスに変換することは、上記のように、及び図 2 A に示すように、仮



想アドレス 270 を物理アドレス 290 に変換することを含み得る。

【0138】

[0147] カーネル空間が第 2 のページサイズを含み得ることを取得されたページサイズインジケータが示し得る場合、ステップ 632 における仮想アドレスを物理アドレスに変換することは、図 3A に示すように、及び上記のように、仮想アドレスを物理アドレスに変換することを含み得る。例えば、ページ属性 262 - 2 のビット 7 が、カーネル空間の 4 MB のページサイズを示す「0」である場合、ステップ 632 における仮想アドレスを物理アドレスに変換することは、図 3A 及び 3B に示すように、並びに上記のように、2 レベルページテーブル（ページディレクトリ 320 及びページテーブルエントリ 360）を通して仮想アドレス 370 を物理アドレス 390 に変換することを含み得る。

10

【0139】

[0148] 別の例として、方法 600 のページサイズインジケータを取得することは、図 4B に示すように、及び上記のように、ページテーブルエントリ 462 のページ属性 462 - 2 のビット 7 を読み出すことを含み得る。ページ属性 462 - 2 のビット 7 は、ユーザ空間のページサイズを示す。例えば、ページ属性 462 - 2 のビット 7 が「1」である場合、ユーザ空間のページサイズは、4 KB でもよい。ページ属性 462 - 2 のビット 7 が「0」である場合、ユーザ空間のページサイズは、4 MB でもよい。

【0140】

[0149] ユーザ空間が第 1 のページサイズを含み得ることを取得されたページサイズインジケータが示し得る場合、ステップ 642 における仮想アドレスを物理アドレスに変換することは、図 4A 及び 4B に示すように、並びに上記のように、仮想アドレスを物理アドレスに変換することを含み得る。例えば、ページ属性 462 - 2 のビット 7 が、ユーザ空間の 4 KB のページサイズを示す「1」である場合、ステップ 642 における仮想アドレスを物理アドレスに変換することは、上記のように、及び図 4A に示すように、仮想アドレス 470 を物理アドレス 490 に変換することを含み得る。

20

【0141】

[0150] ユーザ空間が第 2 のページサイズを含み得ることを取得されたページサイズインジケータが示し得る場合、ステップ 642 における仮想アドレスを物理アドレスに変換することは、図 5A 及び 5B に示すように、並びに上記のように、仮想アドレスを物理アドレスに変換することを含み得る。例えば、ページ属性 562 - 2 のビット 7 が、ユーザ空間の 4 MB のページサイズを示す「0」である場合、ステップ 642 における仮想アドレスを物理アドレスに変換することは、図 5A に示すように、及び上記のように、2 レベルページテーブル（ページディレクトリ 520 及びページテーブルエントリ 560）を通して仮想アドレス 570 を物理アドレス 590 に変換することを含み得る。

30

【0142】

[0151] 方法 600 は、変換された物理アドレスのページに関する実行不可インジケータを取得することを含み得る。例えば、方法 600 の実行不可インジケータを取得することは、図 2B に示すように、及び上記のように、ページテーブルエントリ 262 のページ属性 262 - 2 のビット 2 を読み出すことを含み得る。ページ属性 262 - 2 のビット 2 が「1」である場合、変換された物理アドレスのページは、実行可能でなくてもよい。方法 600 は、ページのコンテンツにアクセスしないことを含むか、又はページのコンテンツにアクセスするが、それを実行しないことを含む。ページ属性 262 - 2 のビット 2 が「0」である場合、変換された物理アドレスのページは、実行可能でもよい。方法 600 は、カーネル空間のページのコンテンツにアクセスすることを含んでもよく、及び / 又はそれに応じてそれを実行し得る。

40

【0143】

[0152] 別の例として、方法 600 の実行不可インジケータを取得することは、ユーザ空間に関する実行不可インジケータを取得することを含み得る。例えば、方法 600 の実行不可インジケータを取得することは、上記のように、及び図 4B に示すように、ページテーブルエントリ 462 のページ属性 462 - 2 のビット 2 を読み出すことを含み得る。

50

ページ属性 4 6 2 - 2 のビット 2 が「1」である場合、変換された物理アドレスのページは、実行可能でなくてもよい。方法 6 0 0 は、ページのコンテンツにアクセスしないことを含むか、又はページのコンテンツにアクセスするが、それを実行しないことを含む。ページ属性 4 6 2 - 2 のビット 2 が「0」である場合、変換された物理アドレスのページは、実行可能でもよい。方法 6 0 0 は、ユーザ空間のページのコンテンツにアクセスすることを含んでもよく、及び / 又はそれに応じてそれを実行し得る。

【0144】

[0153] 方法 6 0 0 は、OS カーネル又はユーザプロセスに関するユーザ空間アクセスインジケーションを取得することをさらに含み得る。例えば、プロセッサ 1 2 0 は、その命令セットにおいて、ユーザ空間にアクセスするために使用される接頭語「US」を含む 1 つ又は複数の命令を含み得る。接頭語「US」を含む命令は、OS カーネルにおけるユーザ空間アクセスインジケータとして使用され得る。「US」の接頭語を有する命令が実行され得る場合、ステップ 6 3 1 においてルートページテーブルインデックスを取得することは、仮想アドレスを物理アドレスに変換するために制御レジスタ 1 2 4 のユーザ空間に対応するルートページテーブルインデックスを読み出すことを含み得る。例えば、ステップ 6 3 1 においてルートページテーブルインデックスを取得することは、制御レジスタ 1 2 4 からユーザ空間に対応するルートアドレス 4 1 0 を読み出すことを含み得る。そして、ステップ 6 3 2 において仮想アドレスを物理アドレスに変換することは、上記のように、ルートアドレス 4 1 0 及び図 4 A の 3 レベルページテーブル構造を使用して仮想アドレス 2 7 0 を物理アドレスに変換することを含み得る。

【0145】

[0154] 別の例として、プロセッサ 1 2 0 は、その命令セットにおいて、ユーザ空間にアクセスするために使用される接頭語「US」を含む 1 つ又は複数の命令を含み得る。接頭語「US」を含む命令は、ユーザプロセスにおいてユーザ空間アクセスインジケータとして使用され得る。ユーザプロセスに対して「US」の接頭語を有する命令が実行され得る場合、「US」の接頭語を有する命令が、OS カーネルだけのものであり得るので、方法 6 0 0 は、一般保護違反を設定することをさらに含み得る。例えば、方法 6 0 0 は、警告メッセージ又はフラグを設定すること、及びユーザプロセスに対して「US」の接頭語を有する命令を実行しないことを含み得る。

【0146】

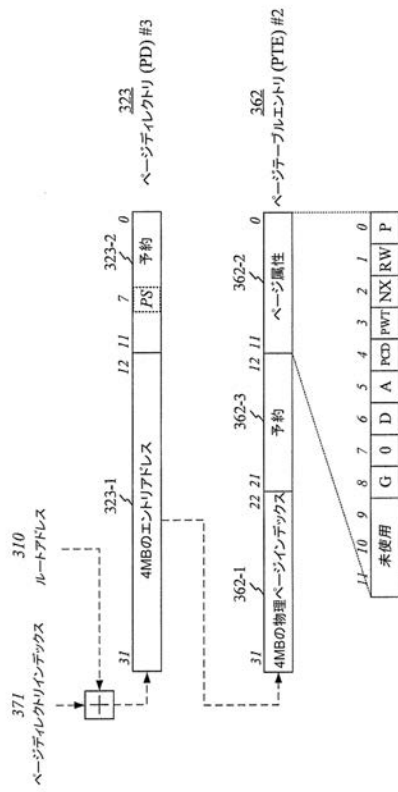
[0155] 本開示の別の態様は、上述のように、カーネル空間及びユーザ空間のメモリ管理のための方法を装置に行わせるための、装置の 1 つ又は複数のプロセッサによって実行可能な命令のセットを保存する非一時的コンピュータ可読媒体に向けられる。コンピュータ可読媒体は、揮発性若しくは不揮発性、磁気、半導体、テープ、光学式、取り外し可能、固定型、又は他のタイプのコンピュータ可読媒体又はコンピュータ可読ストレージデバイスを含み得る。例えば、コンピュータ可読媒体は、開示したような、保存されたコンピュータ命令を有するストレージデバイス又はメモリモジュールでもよい。幾つかの実施形態では、コンピュータ可読媒体は、保存されたコンピュータ命令を有するディスク又はフラッシュドライブでもよい。

【0147】

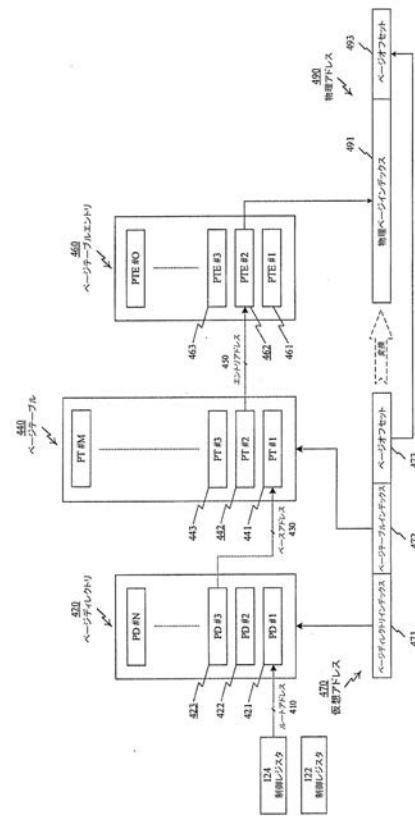
[0156] 本開示は、上記に記載した及び添付の図面に示した正確な構造に限定されないこと、並びに様々な修正及び変更がその範囲から逸脱することなく行われ得ることが理解されるだろう。本出願の範囲は、添付の特許請求の範囲によってのみ限定されることが意図される。



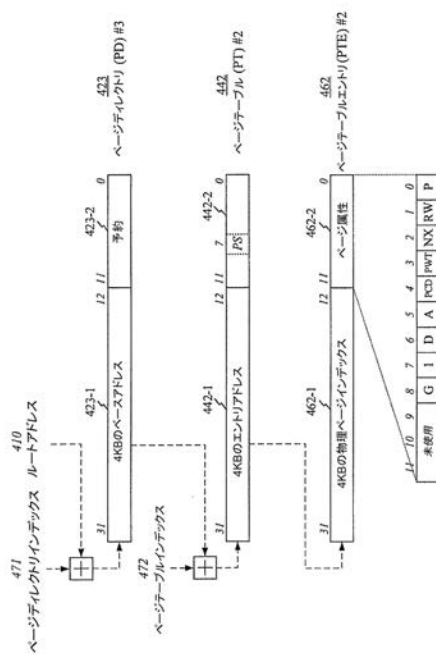
【図 3 B】



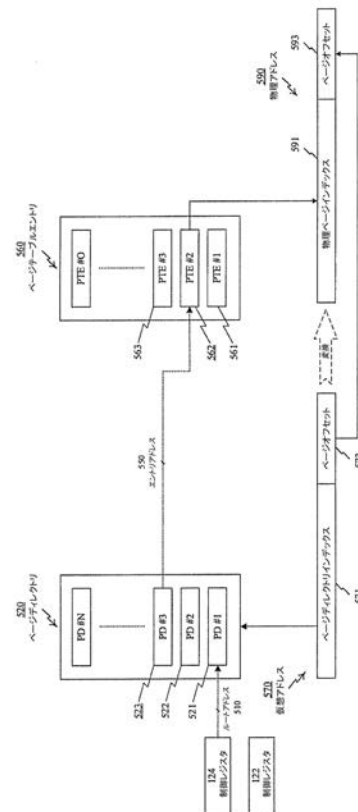
【図 4 A】



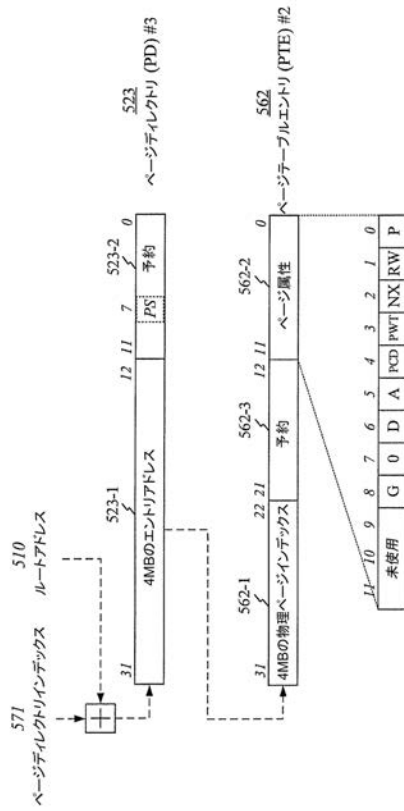
【図 4 B】



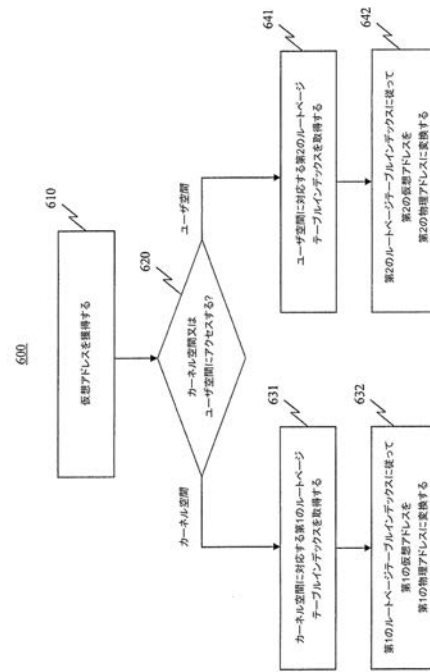
【図 5 A】



【図5B】



【図6】



## 【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US 18/47201
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(8) - G06F 13/00 (2018.01) CPC - G06F 3/0601, G06F 3/0659, G11B 27/034, G06F 21/79, G06F 12/0866		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols)		
See Search History Document		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
See Search History Document		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
See Search History Document		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/0246381 A1 (KEGEL et al.), 27 September 2012 (27.09.2012), entire document, especially Abstract, para [0015], [0043], [0086], [0094]-[0097], [0099], [0102]-[0103], [0106]	1-6, 12, 16-21, 33
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 24 October 2018		Date of mailing of the international search report <b>07 NOV 2018</b>
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300		Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 18/47201

**Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
2. ☐ Claims Nos.:  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
  
3. ☒ Claims Nos.: 7-11, 13-15, 22-32  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
  
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

**Remark on Protest**

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

## フロントページの続き

(81)指定国・地域 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT

(特許庁注：以下のものは登録商標)

1 . S O L A R I S

(72)発明者 ジャン , シャオウェイ

アメリカ合衆国 , カリフォルニア州 9 4 4 0 2 , サン マテオ , エス . エル カミーノ レアル  
4 0 0 , スイート 4 0 0 , アリババ グループ リーガル デパートメント

(72)発明者 リ , シュー

アメリカ合衆国 , カリフォルニア州 9 4 4 0 2 , サン マテオ , エス . エル カミーノ レアル  
4 0 0 , スイート 4 0 0 , アリババ グループ リーガル デパートメント

F ターム(参考) 5B017 AA07 BA04 BB08 CA01

5B205 RR04