

(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2020-87293

(P2020-87293A)

(43) 公開日 令和2年6月4日(2020.6.4)

(51) Int.Cl.

G06F 9/4401 (2018.01)

F I

G06F 9/4401

テーマコード (参考)

5B376

審査請求 未請求 請求項の数 15 O L (全 13 頁)

(21) 出願番号 特願2018-225249 (P2018-225249)  
 (22) 出願日 平成30年11月30日 (2018.11.30)

(71) 出願人 000001007  
 キヤノン株式会社  
 東京都大田区下丸子3丁目30番2号  
 (74) 代理人 100126240  
 弁理士 阿部 琢磨  
 (74) 代理人 100124442  
 弁理士 黒岩 創吾  
 (72) 発明者 伊藤 公一  
 東京都大田区下丸子3丁目30番2号キヤ  
 ノン株式会社内  
 Fターム(参考) 5B376 AE21 AE42 AE62 AE63 FA01  
 GA01

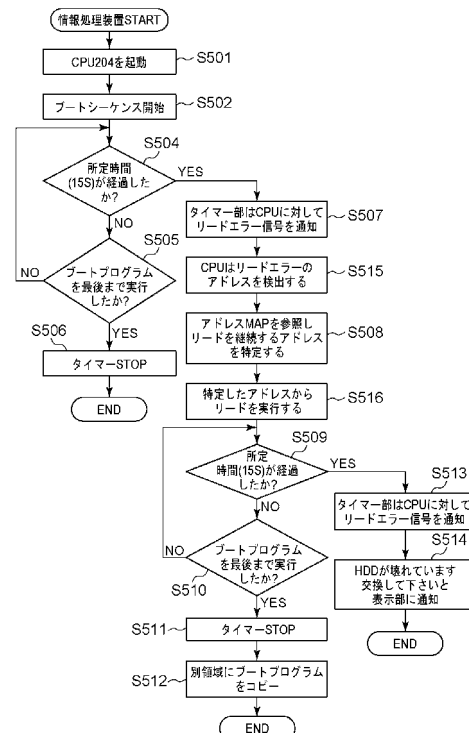
(54) 【発明の名称】 情報処理装置および情報処理装置の制御方法

(57) 【要約】 (修正有)

【課題】ブートプログラムのリードエラーが発生した場合において、情報処理装置を起動するまでにかかる時間を短縮する情報処理装置および情報処理装置の制御方法を提供する。

【解決手段】情報処理装置は、第1ブートプログラムおよび対応する第2ブートプログラムを少なくとも記憶する記憶手段と、プログラムを読み出して実行する実行手段と、実行手段によって第1ブートプログラムを読み出す際に読み出しエラーが生じたことに従って、第1ブートプログラムのうち読み出しエラーが生じたプログラムが記憶された記憶領域のアドレスを検出する検出手段と、第2ブートプログラムが記憶された記憶領域のアドレスのうち、検出手段によって検出したアドレスに対応するアドレスを特定する特定手段と、を有する。実行手段は、特定手段によって特定されたアドレスに記憶された第2ブートプログラムを読み出して実行する。

【選択図】図4



**【特許請求の範囲】****【請求項 1】**

少なくとも 1 つのプロセッサによってブートプログラムが実行されることでオペレーティングシステムが起動する情報処理装置であって、

第 1 ブートプログラムおよび前記第 1 ブートプログラムに対応する第 2 ブートプログラムを少なくとも記憶する記憶手段と、

プログラムを読み出して実行する実行手段と、

前記実行手段によって前記第 1 ブートプログラムを読み出す際に読み出しエラーが生じたことに従って、前記第 1 ブートプログラムのうち読み出しエラーが生じたプログラムが記憶された記憶領域のアドレスを検出する検出手段と、

前記第 2 ブートプログラムが記憶された記憶領域のアドレスのうち、前記検出手段によって検出したアドレスに対応するアドレスを特定する特定手段と、を有し、

前記実行手段は、前記特定手段によって特定されたアドレスに記憶された第 2 ブートプログラムを読み出して実行することを特徴とする情報処理装置。

**【請求項 2】**

前記記憶手段は、

前記第 1 ブートプログラムおよび前記第 2 ブートプログラムは、複数の記憶領域に分けて記憶し、

前記第 1 ブートプログラムおよび前記第 2 ブートプログラムの各々を、プログラムの起動順番とプログラムの格納先であるアドレスとを対応させて管理する管理テーブルを記憶することを特徴とする請求項 1 に記載の情報処理装置。

**【請求項 3】**

前記管理テーブルにおいて、プログラムの起動順番に対応するアドレスは、所定の順番に実行するプログラムの先頭アドレスであることを特徴とする請求項 2 に記載の情報処理装置。

**【請求項 4】**

前記読み出しエラーが生じた際に、前記検出手段で検出されるアドレスに記憶された前記第 1 ブートプログラムの途中までを前記実行手段が実行した状態であった場合には、前記実行手段は前記特定手段によって特定されたアドレスに記憶された第 2 ブートプログラムの先頭から実行することを特徴とする請求項 3 に記載の情報処理装置。

**【請求項 5】**

前記実行手段は、前記特定されたアドレスに記憶された第 2 ブートプログラムを読み出して実行した後は、前記特定されたアドレスの次の起動順番に対応するアドレスに記憶された前記第 2 ブートプログラムを順次読み出して実行することを特徴とする請求項 2 乃至 4 のいずれか 1 項に記載の情報処理装置。

**【請求項 6】**

前記実行手段は、前記特定されたアドレスに記憶された第 2 ブートプログラムを読み出して実行した後は、前記特定されたアドレスの次の起動順に対応するアドレスに記憶された前記第 1 ブートプログラムを順次読み出して実行することを特徴とする請求項 2 乃至 4 のいずれか 1 項に記載の情報処理装置。

**【請求項 7】**

前記ブートプログラムを複製して前記記憶手段に記憶する複製手段を有し、

前記実行手段がブートプログラムの実行を完了したことに基づいて、前記複製手段が前記特定手段によって特定されたアドレスに格納された前記第 2 ブートプログラムを複製した第 3 ブートプログラムを前記記憶手段に記憶させることを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の情報処理装置。

**【請求項 8】**

前記ブートプログラムを複製して前記記憶手段に記憶する複製手段を有し、

前記実行手段がブートプログラムの実行を完了したことに基づいて、前記複製手段が前記特定手段によって特定されたアドレスおよび前記特定されたアドレス以降の起動順のア

10

20

30

40

50

ドレスに格納された前記第 2 ブートプログラムを複製した第 3 ブートプログラムを前記記憶手段に記憶させることを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載の情報処理装置。

【請求項 9】

前記ブートプログラムを複製して前記記憶手段に記憶する複製手段を有し、

前記実行手段がブートプログラムの実行を完了したことに基づいて、前記複製手段が前記第 2 ブートプログラムを複製して第 3 ブートプログラムを前記記憶手段に記憶させることを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の情報処理装置。

【請求項 10】

前記プロセッサを起動するための異なるブートプログラムを記憶する異なる記憶手段と

10

、

前記異なるブートプログラムを起動するための前記プロセッサと異なるプロセッサと、を更に有し、

前記情報処理装置の電源がオンされることで、前記異なるブートプログラムが前記異なるプロセッサに実行され、前記プロセッサを起動することを特徴とする請求項 1 乃至 9 のいずれか 1 項に記載の情報処理装置。

【請求項 11】

時間をカウントするカウント手段を有し、

前記プロセッサが起動することに従って、前記カウント手段が時間のカウントを開始し

20

、

前記実行手段によって前記第 1 ブートプログラムの実行が完了するまでに前記カウントが閾値を超えたら、前記読み出しエラーが生じることを特徴とする請求項 10 に記載の情報処理装置。

【請求項 12】

前記第 1 ブートプログラムと前記第 2 ブートプログラムは同一のプログラムであることを特徴とする請求項 1 乃至 11 のいずれか 1 項に記載の情報処理装置。

【請求項 13】

第 1 ブートプログラムおよび前記第 1 ブートプログラムに対応する第 2 ブートプログラムを少なくとも記憶する不揮発性メモリを有し、少なくとも 1 つのプロセッサによってブートプログラムが実行されることでオペレーティングシステムが起動する情報処理装置の制御方法であって、

30

前記第 1 ブートプログラムを読み出す際に読み出しエラーが生じたことに従って、前記第 1 ブートプログラムのうち読み出しエラーが生じたプログラムが記憶された記憶領域のアドレスを検出するステップと、

前記第 2 ブートプログラムが記憶された記憶領域のアドレスのうち、前記検出したアドレスに対応するアドレスを特定するステップと、

前記特定されたアドレスに記憶された第 2 ブートプログラムを読み出して実行するステップと、を有することを特徴とする情報処理装置の制御方法。

【請求項 14】

請求項 13 に記載の制御方法を、コンピュータに実行させるためのプログラム。

40

【請求項 15】

請求項 14 に記載のプログラムを格納したコンピュータで読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、HDD（ハードディスク）等の不揮発性メモリに記憶されたブートプログラムを読み出して実行する情報処理装置および情報処理装置の制御方法。

【背景技術】

【0002】

50

従来、HDD等の不揮発性メモリは、記憶手段として様々な情報処理装置（PCやサーバ、画像形成装置等）に搭載されている。例えば、画像形成装置ではOSを起動させるためのブートプログラム、印刷ジョブ処理において、画像データや装置のログなどの格納先としてHDD等の不揮発性メモリを使用している。

【0003】

不揮発性メモリの突発的な故障や寿命によって、情報処理装置を起動させる際に、OSを起動させるためのブートプログラムのリードエラーが起きた場合には、情報処理装置が起動できなくなるおそれがある。

【0004】

特許文献1では、同一のブートプログラムを不揮発性メモリの異なる記憶領域に複数記憶している。起動時に用いたブートプログラムにリードエラーが発生した場合には、CPU、不揮発性メモリ、表示部を再起動させ、不揮発性メモリから異なるブートプログラムを読み出すことで情報処理装置を起動させる方法が開示されている。

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特開2009-70224号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

しかしながら、OSを起動するブートプログラムを不揮発性メモリから読み出す際にリードエラーが発生した場合であって、異なる記憶領域に記憶されたブートプログラムを初めから読み出して実行する場合には、装置の起動に時間がかかるおそれがある。

【0007】

本発明の情報処理装置はこのような課題に鑑み、実行することでOSが起動するブートプログラムのリードエラーが発生した場合でも、ブートプログラムを最初から読み出して実行することなく情報処理装置を起動することを目的とする。

【課題を解決するための手段】

【0008】

本発明は、少なくとも1つのプロセッサによってブートプログラムが実行されることでオペレーティングシステムが起動する情報処理装置であって、第1ブートプログラムおよび前記第1ブートプログラムに対応する第2ブートプログラムを少なくとも記憶する記憶手段と、プログラムを読み出して実行する実行手段と、実行手段によって第1ブートプログラムを読み出す際に読み出しエラーが生じたことに従って、第1ブートプログラムのうち読み出しエラーが生じたプログラムが記憶された記憶領域のアドレスを検出する検出手段と、第2ブートプログラムが記憶された記憶領域のアドレスのうち、検出手段によって検出したアドレスに対応するアドレスを特定する特定手段と、を有し、実行手段は、特定手段によって特定されたアドレスに記憶された第2ブートプログラムを読み出して実行することを特徴とする。

【発明の効果】

【0009】

本発明は、ブートプログラムのリードエラーが発生した場合において、情報処理装置を起動するまでにかかる時間を短縮することが可能である。

【図面の簡単な説明】

【0010】

【図1】情報処理装置のシステム制御構成を示すブロック図

【図2】プログラムの起動順の説明図

【図3】ブートプログラムの構成及びプログラム管理テーブルの構成を示す図

【図4】ブートプログラムのリードエラー時のフロー図

【図5】ブートプログラムの構成及びプログラム管理テーブルの構成を示す図

10

20

30

40

50

【図 6】ブートプログラムの構成及びプログラム管理テーブルの構成を示す図

【図 7】プログラム管理テーブルの構成を示す図

【図 8】ブートプログラムのリードエラー時のフロー図

【発明を実施するための形態】

【0011】

添付図面を参照して本発明の各実施例を詳しく説明する。なお、以下の実施例は特許請求の範囲に係る発明を限定するものではなく、また各実施例で説明されている特徴の組み合わせのすべてが本発明の解決手段に必須のものとは限らない。

【0012】

(実施例 1)

10

図 1 は、本実施例における情報処理装置 10 の構成の一例を示すブロック図である。図 1 の情報処理装置 10 は、操作部ユニット 150、FAX ユニット 160、コントローラユニット 100、プリンタユニット 120、スキャナユニット 130、電源ユニット 140、スイッチ 145、スイッチ 148 を有する。

【0013】

CPU システムであるコントローラユニット 100 は、CPU 204、ROM 103、RAM 104、HDD 206、ネットワーク I/F 106、EC 部 209 (Embedded Controller) を有する。

【0014】

CPU 204 は、RAM 104 に格納されたソフトウェアプログラムを実行し、装置全体の制御を行う。

20

【0015】

ROM 103 は、例えばコントローラユニット 100 の起動に係る形成を実行させるプログラムや固定パラメータ等を格納している。

【0016】

RAM 104 は、CPU 204 が情報処理装置 10 を制御する際に、プログラムや一時的なデータの格納などに使用される。なお、RAM 104 に格納されるプログラムや一時的なデータは、ROM 103 や後述の HDD 206 から読み出されたものである。

【0017】

HDD 206 は、CPU 204 が実行するプログラムや後述のプログラム管理テーブルや各種データをメインストレージとして格納する。実行するプログラムとは、例えば、情報処理装置の起動時に OS (オペレーティングシステム) を起動させるために CPU 204 が実行するブートプログラム (BOOT Loader 302、Kernel 303) である。ブートプログラムおよびプログラム管理テーブルの詳細な説明は後述する。

30

【0018】

なお、ここでは、ストレージとして HDD を用いたが、eMMC、NAND フラッシュメモリまたは NOR フラッシュメモリ等でもよい。を用いたが、HDD 等でもよい。

【0019】

ネットワーク I/F 106 は、ネットワーク 118 とのデータの送受信を行う。

【0020】

40

次に電源ユニット 140 は、情報処理装置 10 に電源を供給する。電源が OFF のときは、AC 電源はスイッチ 148 によって絶縁され、スイッチ 148 がオンになることで AC - DC コンバータ 141 に AC 電源が供給され、DC 電源が作られる。

【0021】

AC 電源 (電源装置) は、CPU 204 の指示によって装置全体を 3 つの独立した電源に制御することが可能である。スイッチ 142 によって、コントローラユニット 100 への電力の供給を制御することが可能である。スイッチ 143 において、プリンタユニット 120 への電力の供給を制御することが可能である。スイッチ 144 において、スキャナユニット 130 への電力の供給を制御することが可能である。

【0022】

50

次にその他のユニット及びスイッチを説明する。

【0023】

FAXユニット160は、電話回線などにデジタル画像を送受信することが可能である。

【0024】

操作部ユニット150は、タッチ式の操作パネルや、表示画面及びキー等によって構成される。なお、表示画面には、情報処理装置の各種状態（例えば、コピー状況や設定、エラー発生時のエラー表示など）を表示する。

【0025】

スキャナユニット130は、原稿を読み取り、白黒2値やカラー多値の画像データを生成する。スキャナユニット130は、不図示のスキャナ制御I/FによってCPU204に接続される。CPU204は、スキャナ制御I/Fを介してスキャナユニット130から入力される画像信号を制御する。

10

【0026】

プリンタユニット120は、ネットワークI/Fが受け付けたPDLデータを変換した画像データまたはスキャナユニット130によって生成した画像データ等を用紙（シート）に印刷する。

【0027】

プリンタユニット120には、例えば、CPU161と定着器162を含む。定着器162は、用紙に転写されたトナー像を熱と圧力により用紙に融着させるものである。図1において定着器162には、スイッチ145を介してAC電源から電力が供給され、電力が供給されることで熱が生じる。なお、AC-DCコンバータ141を介して電力が供給されてもよい。CPU161は、不図示のRAM等を用いてプリンタコントローラとして機能する。CPU161にはAC-DCコンバータ141を介して電力が供給される。またCPU161は、定着器162への電力の供給をスイッチ145によって制御する。

20

【0028】

スイッチ148は、情報処理装置10に電源を供給するか否かを、スイッチをオンまたはオフにすることによって切り替える。

【0029】

EC部209は、ブートプログラム（BIOS）を格納している不揮発性メモリである。ブートプログラムの詳細な説明は後述する。

30

【0030】

画像処理部208は、CPU204、プリンタユニット120、スキャナユニット130に接続されている。画像処理部208は、スキャナユニット130から出力されたデジタル画像を色空間変換などの画像処理を行い、画像処理後のデータをCPU204に出力する。また、画像処理部208は、スキャナユニット130で読み取られた画像データを基に色空間変換などの画像処理を行い、ビットマップデータに変換し、プリンタユニット120に出力する。

【0031】

なお、CPU204は、HDD206に対してデータの書き込み、データの読み出しを行わせる機能（HDD制御部234）を有する。ここでは、HDDの制御を行う機能としたが、HDDの代わりに異なるメモリを用いた場合には、異なるメモリの制御を行う機能となる。さらに、CPU204は、時間をカウントするタイマー機能を有する（タイマー部235）。CPUは、タイマー機能において、情報処理装置10の起動に従ってカウントを開始する。

40

【0032】

図2は、情報処理装置10を起動する際に実行するプログラムの順番を示した図である。なお、起動する際は、情報処理装置10をオンする指示を受け付けた状態である。

【0033】

BIOS301は、情報処理装置10の電源を入れることによってEC部209に含ま

50

れるCPU210で実行されるプログラムであり、EC部209に含まれる不図ROM211に記憶されている。

【0034】

BIOS301が実行されると、例えば電源が供給されたCPU204のリセット状態を解除する等の装置の起動に係わる処理を行う。つまり、BIOS301は、CPU204を起動するためのプログラムである。CPU204が起動すると、CPU204のタイマー部235が時間のカウントを開始する。

【0035】

BOOTLoader302は、BIOS301の実行後に、CPU204で実行されるプログラムであり、ROM103に格納されている。BOOTLoader302が実行されると、HDD206に格納されているLinux（登録商標）などのOSが起動する。そして、BOOTLoader302が実行されると、例えば、スイッチ146がオンし、プリンタユニット120に電力が供給される。さらに、操作部ユニット150の初期化をし、タッチパネル又は表示画面に描画可能な状態とする。

【0036】

Kernel303は、BOOTLoader302の実行が完了した後にCPU204で実行されるプログラムであり、OSの核となるプログラムである。例えば、他のプログラムの動作やメモリの管理などを行う。具体的には、スキャナのI/Fを制御するプログラムやプリンタのI/Fを制御するプログラムや起動プログラムなどである。

【0037】

Kernel303の実行が終了すると、CPU204は、OSの起動が完了したと判断する。CPU204のタイマー部235は、CPU204がOSの起動を完了すると時間のカウントを終了する。

【0038】

上述したように図3において、BOOTLoader302とKernel303は、OSを起動させるためのブートプログラムとなる。

【0039】

BIOS301の実行によってCPU204が起動し、タイマー部235が時間のカウントを開始してから、所定の時間が経過した時点で、ブートプログラムの実行が完了していない場合には、ブートプログラムのリードエラーとなる。所定の時間は、ブートプログラムの実行に必要な時間よりも長く設定され、例えば、起動に必要な時間が10秒であれば、15秒を閾値として設定する。

【0040】

リードエラーが発生した場合には、タイマー部235は、CPU204に対してリードエラー信号を通知する。

【0041】

次に図3は、本発明のHDD206のデータ構成およびHDD206に格納されるプログラム管理テーブルを示す図である。

【0042】

図3(a)においてHDD206は、ブートプログラム304、ブートプログラム305、その他保存データ306を格納している。ブートプログラム304は、OSを起動させるブートプログラム（BOOTLoader302とKernel303）である。ブートプログラム305は、ブートプログラム304と同じプログラムである。その他保存データ306は、例えばユーザの設定データやスキャナユニット130によって生成された画像データや図3(b)に示すプログラム管理テーブルである。

【0043】

なお、図3では、同一のブートプログラムが2つ格納される例を示したが、2つ以上でもよい。また、OSを起動させるためのブートプログラムの例として、BOOTLoader302とKernel303を挙げているが、これに限定するものではない。CPU204が起動した後に実行するOSを起動させるためのブートプログラムにあたるプログ

10

20

30

40

50

ラムはすべて複製してもよい。

【0044】

また、本実施例では、同一のブートプログラムを2つ持つ構成を示すが、設定データなどを2つ持っていてよい。

【0045】

さらに、ここで同一とは、実質的に同一であればよい。例えば、一方のブートプログラムに対して、他方のプログラムに改編があっても、同様の機能を実現する場合には同一とみなす。

【0046】

次に図3(b)を用いてプログラム管理テーブル307について説明する。プログラム管理テーブル307は、HDD206に保存されている。

10

【0047】

プログラム管理テーブル307は、ブートプログラム304およびブートプログラム305のBOOT Loader 302を3つのブロック、Kernel 303を2つのブロックに分けて、それぞれ異なる記憶領域に記憶されている。

【0048】

各記憶領域(格納先)はアドレスによって管理され、先頭アドレスが管理テーブルに記載されている。そして、プログラムの起動順番に対応させて先頭アドレスが管理されており、プログラムを起動する際には、起動順番に合わせて順次読み出し、実行する。なお、プログラム管理テーブル307において起動順に対応する各ブートプログラムのアドレスに格納されているプログラムの内容は同一である。

20

【0049】

例えば、起動順3に当たるBOOT Loaderのアドレスはブートプログラム304だと「0x1236」になり、ブートプログラム305だと「0xFF03」となる。

【0050】

本実施例では、BOOT Loaderを3つのブロック、Kernelを2つのブロックに分けているが、これに限定するものではない、また、アドレスが連続であるが、不連続でも良い。

【0051】

次に、図4を用いて本実施例に係る情報処理装置10の電源がオンすることによる起動処理手順を説明する。

30

【0052】

情報処理装置10の電源がオンされるとフローチャートが開始する。

【0053】

ステップS501において、情報処理装置10の電源がオンされるとEC部209がCPU204を起動する。

【0054】

ステップS502において、CPU204は、タイマー部235による時間のカウントを開始し、且つ、HDD206からブートプログラム304を読み出して実行する。

【0055】

ステップS504において、CPU204は、タイマー部235によってカウントしている時間が所定の時間(例えば15s)を経過したか否かを判断する。経過していない場合にはステップS505に遷移する。

40

【0056】

ステップS505において、CPU204は、ブートプログラムの実行が完了されたか否かを判断する。CPU204が、入力されていないと判断するとステップS504に戻る。一方、CPU204が、入力されたと判断するとステップS506に遷移する。

【0057】

ステップS506において、CPU204は、タイマー部235による時間のカウントを停止する。ステップS506を終えると、フローチャートを終了し、情報処理装置10

50



が使用可能になる。ステップの S 5 0 4 の説明に戻る。

【 0 0 5 8 】

ステップ S 5 0 4 において、タイマー部 2 3 5 によってカウントしている時間が所定の時間を経過した場合にはステップ S 5 0 7 に遷移する。

【 0 0 5 9 】

ステップ S 5 0 7 において、所定の時間を経過してもブートプログラムの実行が完了されない場合には、C P U 2 0 4 にリードエラー信号が生じる。そして、ステップ S 5 1 5 において、リードエラー信号を受け付けた C P U 2 0 4 は、リードエラーが生じたアドレスを検出する。なお、C P U 2 0 4 は、エラー信号が生じたらタイマーをストップする。

【 0 0 6 0 】

そして、ステップ S 5 0 8 においてプログラム管理テーブルを参照して、リードエラーが生じたアドレスに対応する、他のブートプログラムのアドレスを特定する。

【 0 0 6 1 】

ステップ S 5 1 6 において、C P U 2 0 4 は、ステップ S 5 0 8 で特定したアドレスからプログラムの読み出し、実行を継続する。なお、例えば、リードエラーが生じたアドレスに記憶されているプログラムの途中まで実行していたとしても、特定したアドレスに記憶されたプログラムの先頭から読み出しを行う。

【 0 0 6 2 】

ステップ S 5 1 6 を終わると、ステップ S 5 1 7 で、C P U 2 0 4 は、H D D 2 0 6 にリードエラーが発生しなかったブートプログラムを複製させ、リードエラーが生じたブートプログラムを削除させる。なお、ステップ S 5 1 7 は、ステップ S 5 1 6 の後であればどのタイミングでもよい。

【 0 0 6 3 】

ここで、ステップ S 5 0 8、ステップ S 5 1 5、S 5 0 8、ステップ S 5 1 6 について図 5 を用いて具体的に説明する。図 5 ( a ) は、C P U 2 0 4 がブートプログラム 3 0 4 を実行している際に、エラー信号を受け付けたことを示している。ステップ S 5 1 5 において、C P U 2 0 4 は、アドレス 0 x 1 2 3 6 に格納されたプログラムの読み出しでエラーが発生したことを検出する。

【 0 0 6 4 】

ステップ S 5 0 8 において、C P U 2 0 4 は、ブートプログラム 3 0 4 のアドレス 0 x 1 2 3 6 に対応するアドレスとして、ブートプログラム 3 0 5 のアドレス 0 x F F 0 3 を特定する。

【 0 0 6 5 】

ステップ S 5 1 6 において、C P U 2 0 4 は、ブートプログラム 3 0 5 のアドレス 0 x F F 0 3 から起動順にプログラムを読み出し実行する。具体的には、アドレス 0 x F F 0 3 の後、アドレス 0 x F F 0 4、アドレス 0 x F F 0 5 の順にブートプログラム 3 0 5 を読み出して実行する。なお、アドレス 0 x F F 0 3 の後、アドレス 0 x 1 2 3 7、アドレス 0 x 1 2 3 8 の順にブートプログラム 3 0 4 を読み出して実行してもよい。ステップ S 5 1 6 を終わるとステップ S 5 0 9 に遷移する。

【 0 0 6 6 】

次にステップ S 5 0 9 において、C P U 2 0 4 のタイマー部 2 3 5 によってカウントされた時間を確認する。カウントされた時間が所定時間を超えていない場合には、ステップ S 5 1 0 に遷移する。

【 0 0 6 7 】

ステップ S 5 1 0 において、C P U 2 0 4 は、ブートプログラムを最後まで実行したか否かを判断する。C P U 2 0 4 がブートプログラムを最後まで実行していない場合には、ステップ S 5 0 9 に戻る。C P U 2 0 4 がブートプログラムを最後まで実行した場合には、ステップ S 5 1 1 に遷移する。ステップ S 5 1 1 において、C P U 2 0 4 のタイマー部 2 3 5 は、時間のカウントを停止する。ステップ S 5 1 1 を終わると、ステップ S 5 1 2 に遷移する。

10

20

30

40

50

## 【 0 0 6 8 】

ステップ S 5 1 2 において、C P U 2 0 4 は、図 5 ( b ) および図 5 ( c ) に示すように、ブートプログラム 4 0 1 を H D D 2 0 6 に記憶させる。ブートプログラム 4 0 1 は、ブートプログラム 3 0 5、ブートプログラム 3 0 4 と同じ内容のプログラムである。ここではブートプログラム 4 0 1 は、ブートプログラム 3 0 5 を複製することで作成する。

## 【 0 0 6 9 】

また、例えば、ブートプログラム 3 0 4 を読みだす際にエラーが生じた起動順のアドレスに対応するアドレスおよび対応するアドレス以降に格納されたブートプログラム 3 0 5 を複製し、ブートプログラム 4 0 4 を作成してもよい。(図 6 ( a )、図 6 ( b ))

この場合には、ブートプログラム 3 0 4 は、エラーが生じたアドレス以降のプログラムを削除する。なお、ブートプログラム 3 0 4 を読みだす際にエラーが生じた起動順のアドレスに対応するアドレスおよび対応するアドレス以降に格納されたブートプログラム 3 0 5 のみを複製し、ブートプログラム 4 0 4 を作成してもよい。この場合には、C P U 2 0 4 は、ブートプログラム 3 0 4 を読みだす際にエラーが生じた起動順のアドレスに格納されたプログラムのみを削除する。なお、削除されたプログラムが格納されていたアドレスは、未使用領域となる。

## 【 0 0 7 0 】

このように、複製することにより、常に同一ブートプログラムを 2 つ持つことが出来る。このような構成によれば、仮に一方のブートプログラムがリードエラーでも、他方のブートプログラムから起動することが可能である。

## 【 0 0 7 1 】

ステップ S 5 1 2 を終わると、フローが終了し、情報処理装置 1 0 が使用可能になる。

## 【 0 0 7 2 】

ステップ S 5 0 9 の説明に戻る。ステップ S 5 0 9 において、確認した時間が所定時間を超えた場合には、ステップ S 5 1 3 に遷移する。

## 【 0 0 7 3 】

ステップ S 5 1 3 において、C P U 2 0 4 のタイマー部は、C P U 2 0 4 に対してリードエラー信号を通知する。

## 【 0 0 7 4 】

ステップ S 5 1 4 において、C P U 2 0 4 は操作部ユニット 1 5 0 の表示画面に、例えば「H D D が壊れています。交換してください。」等の、H D D に関する通知を表示させる。そして、フローが終了する。

## 【 0 0 7 5 】

なお、本実施例では、H D D 2 0 6 が同一のブートプログラムを 2 つ記憶する構成を例として示したが、複数(例えば 3 つ以上)でもよい。例えば、H D D 2 0 6 に記憶されたブートプログラムが 3 つの場合には、ステップ S 5 0 9 で Y E S となると、S 5 0 7 ~ S 5 1 6 を実行してもよい。つまり、同一のブートプログラムの数だけ、S 5 0 4、S 5 0 7、S 5 1 5、S 5 0 8、S 5 1 6、S 5 0 9 を実行してもよい。

## 【 0 0 7 6 】

本実施例の構成によれば、H D D 等の不揮発性メモリの突発な故障や寿命による、O S を起動するためのブートプログラムのリードエラーが起きても、ブートプログラムを初めから読み出す必要がないため起動にかかる時間を少なくすることが可能である。

## 【 0 0 7 7 】

(実施例 2)

本実施例は、実施例 1 とリードエラーが起こったアドレスに対応する、他のブートプログラムのアドレスの特定の仕方が異なる。ここでは、実施例 1 との差異についてのみ説明する。

## 【 0 0 7 8 】

図 7 に本実施例におけるプログラム管理テーブル 7 0 1 を示す。

## 【 0 0 7 9 】

ブートプログラム 304 の起動順の各プログラムのアドレスの先頭と、ブートプログラム 305 の起動順の各プログラムのアドレスの先頭とが、所定のアドレスを加算した構成となっている。具体的には、例えば、起動順 1 におけるブートプログラム 304 のアドレスは、アドレス  $0 \times FF01$  であり、起動順 1 におけるブートプログラム 305 のアドレスは、アドレス  $0 \times FFF1$  である。つまり、ブートプログラム 304 のアドレス  $0 \times FF01$  に、アドレス  $0 \times 00F0$  を加算した構成となっている。

【0080】

図 8 を用いて本実施例に係る情報処理装置 10 の電源がオンすることによる起動処理手順を説明する。ここでは、図 4 との差異である、ステップ S701、S702 の前後についてのみ説明する。

10

【0081】

ステップ S515 において、リードエラー信号を受け付けた CPU 204 は、リードエラーが生じたアドレスを検出する。

【0082】

ステップ S701 において、検出したアドレスに対して、所定のアドレスを加算する。ステップ S701 を終わると、ステップ S702 に遷移する。ステップ S702 では、加算したアドレスからプログラムのリードを継続する。ステップ S702 を終わると、ステップ S509 に遷移する。

【0083】

本実施例の構成においても、HDD 等の不揮発性メモリの突発な故障や寿命による、OS を起動するためのブートプログラムのリードエラーが起きても、ブートプログラムを初めから読み出す必要がないため起動にかかる時間を少なくすることが可能である。

20

【0084】

(その他の実施形態)

以上、本発明の様々な例と実施形態を示して説明したが、本発明の趣旨と範囲は、本明細書内の特定の説明に限定されるものではない。

【0085】

本発明は、上述の実施形態の 1 以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける 1 つ以上のプロセッサがプログラムを読み出し実行する処理でも実現可能である。また、1 以上の機能を実現する回路 (例えば、ASIC) によっても実現可能である。

30

【符号の説明】

【0086】

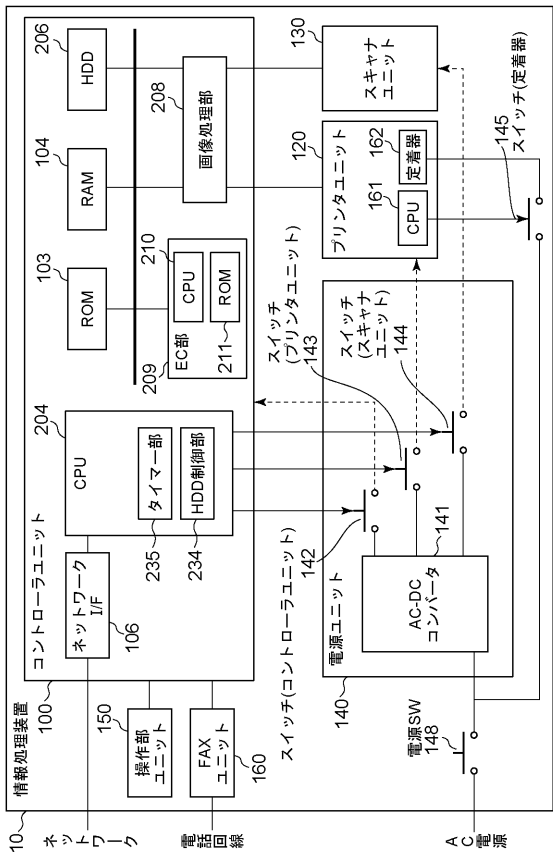
204 CPU

206 HDD

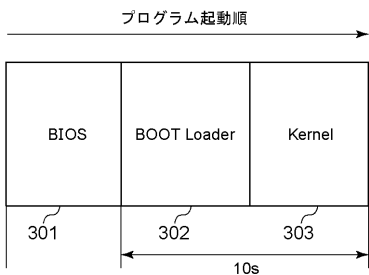
304 ブートプログラム

305 ブートプログラム

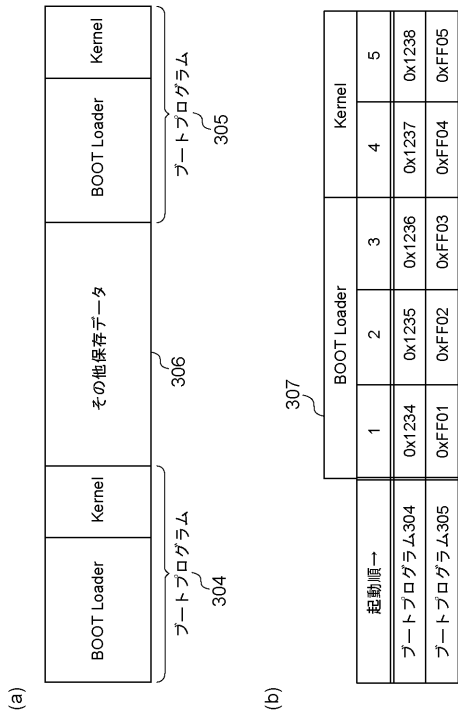
【図 1】



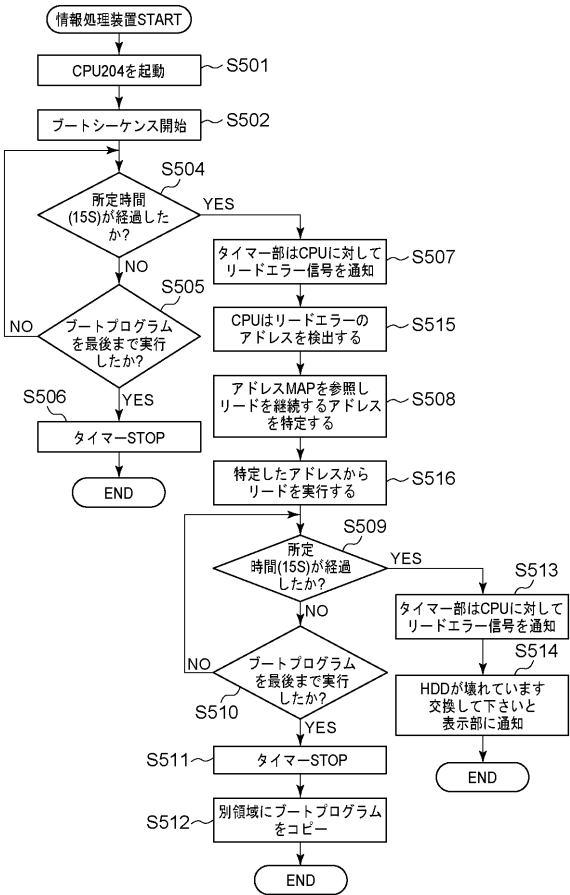
【図 2】



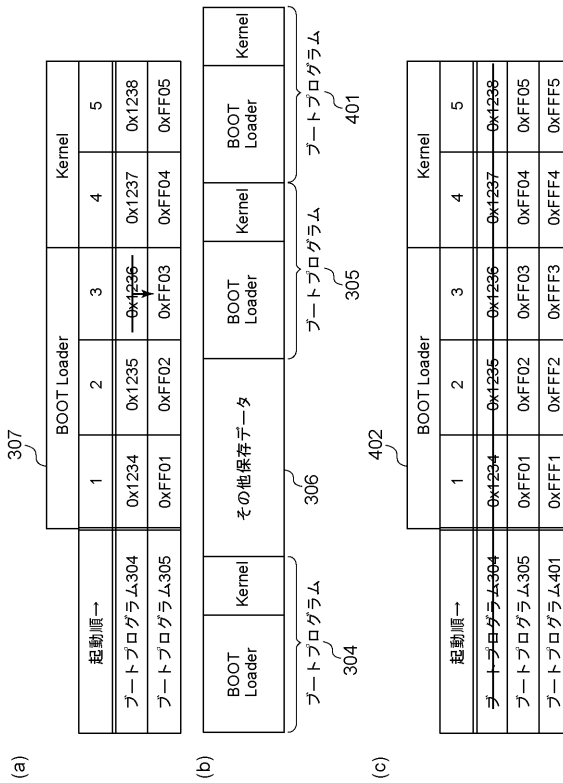
【図 3】



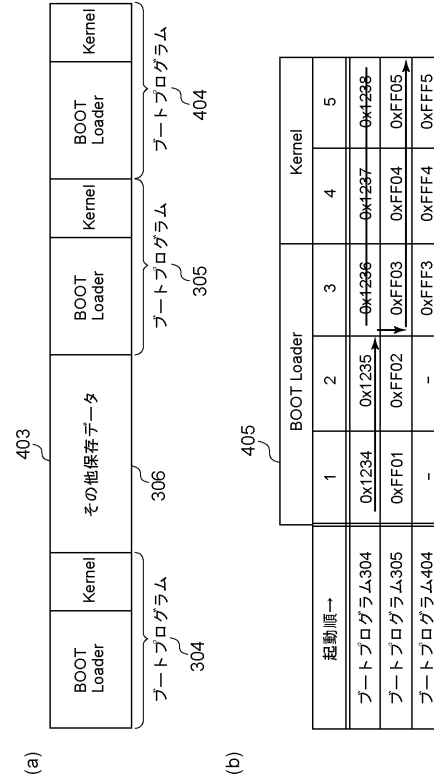
【図 4】



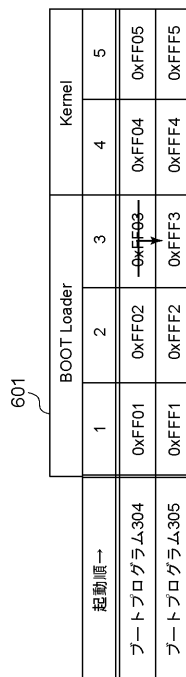
【図 5】



【図 6】



【図 7】



【図 8】

