

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5264250号  
(P5264250)

(45) 発行日 平成25年8月14日(2013.8.14)

(24) 登録日 平成25年5月10日(2013.5.10)

(51) Int.Cl.

F I

G 0 6 F 11/34 (2006.01)

G 0 6 F 11/34

H

請求項の数 15 (全 28 頁)

(21) 出願番号 特願2008-96352 (P2008-96352)  
 (22) 出願日 平成20年4月2日(2008.4.2)  
 (65) 公開番号 特開2009-251747 (P2009-251747A)  
 (43) 公開日 平成21年10月29日(2009.10.29)  
 審査請求日 平成23年4月4日(2011.4.4)

(73) 特許権者 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 100125254  
 弁理士 別役 重尚  
 (72) 発明者 水野 貴史  
 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

審査官 大塚 俊範

最終頁に続く

(54) 【発明の名称】 処理装置及びその制御方法、並びに制御プログラム

(57) 【特許請求の範囲】

【請求項 1】

稼動状況に関する情報を含む第1のログ、及び、エラーに関する情報を含む第2のログを送信する外部装置と通信可能な処理装置において、

前記外部装置から送信された前記第1のログを受信する受信手段と、

前記受信手段によって受信された第1のログから求められた前記外部装置の稼動状況に基づいて、当該外部装置から送信される前記第2のログを格納するか否かを決定するための優先度を判定する判定手段と、

前記判定手段によって判定された優先度に基づいて、前記外部装置が送信した前記第2のログを格納するか否かを決定する決定手段とを備えたことを特徴とする処理装置。

10

【請求項 2】

前記処理装置自身の動作状況を判断する判断手段と、

前記判断手段によって判断された動作状況に応じて、前記優先度を補正する補正手段とをさらに備え、

前記決定手段は、前記補正手段によって補正された優先度に基づいて、前記第2のログを格納するか否かを決定することを特徴とする請求項1に記載の処理装置。

【請求項 3】

操作者が情報を入力するための操作手段を備え、

前記補正手段は、前記操作手段が操作されている場合に優先度を補正することを特徴とする請求項2に記載の処理装置。

20

**【請求項 4】**

前記補正手段は、前記処理装置で処理中のジョブの種類に応じて優先度を補正することを特徴とする請求項 2 に記載の処理装置。

**【請求項 5】**

前記処理装置自身の前記第 1 のログを格納する格納手段と、

前記判定手段が、前記格納手段に格納された前記処理装置自身の第 1 のログを解析し、前記処理装置自身の動作状況として現在の時刻における使用頻度が一定の閾値以上であると判断した場合に、前記外部装置が前記処理装置に対して前記第 2 のログの格納を要求するためのログ格納要求を拒否する手段とを備えたことを特徴とする請求項 2 に記載の処理装置。

10

**【請求項 6】**

前記ログ格納要求を拒否した場合に、前記処理装置自身の動作状況としての使用頻度が一定の閾値未満の時間帯において前記第 2 のログを送信するように前記外部装置に指示する手段を有することを特徴とする請求項 5 に記載の処理装置。

**【請求項 7】**

前記ログ格納要求内に、優先して前記第 2 のログを格納することを指示するフラグを有することを特徴とする請求項 5 または 6 に記載の処理装置。

**【請求項 8】**

前記ログ格納要求を拒否した場合に、前記第 2 のログの格納を延期した外部装置が記載されたログ書き込み待ちリストに、前記ログ格納要求が拒否された外部装置を追加する更新手段と、

20

前記ログ書き込み待ちリストから外部装置を選択する選択手段と、

前記選択手段で選択された外部装置の第 2 のログを取得する取得手段とを備え、

前記取得手段で取得された第 2 のログを前記格納手段に格納することを特徴とする請求項 5 に記載の処理装置。

**【請求項 9】**

前記処理装置自身の第 1 のログを解析した結果に基づいて、前記取得手段の実行の可否を決めることを特徴とする請求項 8 に記載の処理装置。

**【請求項 10】**

前記選択手段は、前記判定手段で判定された優先度に基づいて外部装置を選択することを特徴とする請求項 8 に記載の処理装置。

30

**【請求項 11】**

前記選択手段が前記ログ書き込み待ちリストから外部装置を選択するときに用いる前記優先度を補正する第 2 の補正手段を備えたことを特徴とする請求項 10 に記載の処理装置。

**【請求項 12】**

前記第 2 の補正手段は、前記ログ格納要求の回数に基づいて前記優先度を補正することを特徴とする請求項 11 に記載の処理装置。

**【請求項 13】**

前記第 2 の補正手段は、前記ログ格納要求があった時間に基づいて前記優先度を補正することを特徴とする請求項 11 に記載の処理装置。

40

**【請求項 14】**

稼動状況に関する情報を含む第 1 のログ、及び、エラーに関する情報を含む第 2 のログを送信する外部装置と通信可能な処理装置の制御方法であって、

前記外部装置から送信された前記第 1 のログを受信する受信工程と、

前記受信工程によって受信された第 1 のログから求められた前記外部装置の稼動状況に基づいて、当該外部装置から送信される前記第 2 のログを格納するか否かを決定するための優先度を判定する判定工程と、

前記判定工程によって判定された優先度に基づいて、前記外部装置が送信した前記第 2 のログを格納するか否かを決定する決定工程とを備えたことを特徴とする処理装置の制御方法。

50

## 【請求項 15】

稼動状況に関する情報を含む第1のログ、及び、エラーに関する情報を含む第2のログを送信する外部装置と通信可能な処理装置の制御方法を実行するための、コンピュータで読み取り可能な制御プログラムであって、

受信手段が、前記外部装置から送信された前記第1のログを受信する受信ステップと、

判定手段が、前記受信ステップによって受信された第1のログから求められた前記外部装置の稼動状況に基づいて当該外部装置から送信される前記第2のログを格納するか否かを決定するための優先度を判定する判定ステップと、

決定手段が、前記判定ステップによって判定された優先度に基づいて、前記外部装置が送信した前記第2のログを格納するか否かを決定する決定ステップとを、前記処理装置に実行させることを特徴とする制御プログラム。

10

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、ログを収集する機能を備えた複写機等の処理装置及びその制御方法、並びに前記制御方法を実現するための制御プログラムに関するものである。

## 【背景技術】

## 【0002】

オフィス内で、専用のサーバ機器を置かずに、画像形成装置をサーバとして利用して、専用サーバ機器レスで運用できる環境が広まりつつある。このような状況において、C P Uの速度が速かったり、メモリを多く搭載しているなど性能が高い画像形成装置（親機）が、サーバの機能を有し、前述した複数のネットワーク上の画像形成装置（子機）を管理するデバイス管理の機能を持つようになってきている。

20

## 【0003】

このような親機による子機のデバイス管理においては、子機からのログの収集が行われる。ログには、印刷枚数など通常のログと、エラーが発生したときに障害を特定するためのエラーログがある。特にエラーログは、障害特定のために多くの情報を含むのでデータのサイズが大きい。

## 【0004】

子機は、ログを記憶する記憶領域のサイズが小さいため、記憶領域がメモリフルになると、ログが古いものから順番に削除され、エラーログを多く格納することができない。そこで、子機は、格納しきれないログが消えてしまうことを回避するために、エラーが発生したときにはエラーログを逐一親機へ送る必要がある。

30

## 【0005】

しかし、親機が複数の子機からエラーログのような大きなサイズのログを無条件に受け付けてしまうと、親機自身のパフォーマンスが劣化する可能性がある。この問題に対応するために子機側のメモリ容量を増やすことは、コスト面から難しい。そこで、一定容量のエラーログ用のメモリしかなくても効率よくエラーログの収集を行える技術が求められていた。

## 【0006】

40

このようなログ収集を行う従来の技術として、次のようなものがあった（例えば特許文献1参照）。即ち、複数の子機と、ネットワークを介してそれらの子機のログを収集する親機から成る。子機の優先度が定義された優先度定義テーブルがあり、子機に格納されたログの量が所定量に達した場合に、当該子機の優先度が予め設定された優先度より高い場合にのみログを収集する。また、情報に優先度を付けて、子機に格納されたログの量が所定量に達した場合に、情報の優先度が予め設定された優先度より高い場合にのみログを収集する。

【特許文献1】特開2001-147839号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

50

## 【 0 0 0 7 】

しかしながら、上述した従来の技術は、個々の子機或いは情報に対して予め設定された固定的な優先度を基にして、指定した優先度以上の場合のみにログ収集を行うものである。そのため、子機の稼動状況に応じて自動的に優先度を設定することができない。また、親機自身がパフォーマンスを落とさずに機能を実行しなければならないような状況において、親機自身の状態に応じて動的にログの受け入れの可否を決定することができない。

## 【 0 0 0 8 】

このように上記従来技術では、未だ十分効率よくエラーログの収集を行うことができなかった。

## 【 0 0 0 9 】

本発明は上記従来の問題点に鑑み、処理装置自身のパフォーマンスを低下させることなく、外部処理装置から効率的にエラーログの受け付けを行うことができる処理装置及びその制御方法、並びに制御プログラムを提供することを目的とする。

## 【課題を解決するための手段】

## 【 0 0 1 0 】

上記目的を達成するため、本発明の処理装置は、稼動状況に関する情報を含む第1のログ、及び、エラーに関する情報を含む第2のログを送信する外部装置と通信可能な処理装置において、前記外部装置から送信された前記第1のログを受信する受信手段と、前記受信手段によって受信された第1のログから求められた前記外部装置の稼動状況に基づいて、当該外部装置から送信される前記第2のログを格納するか否かを決定するための優先度を判定する判定手段と、前記判定手段によって判定された優先度に基づいて、前記外部装置が送信した前記第2のログを格納するか否かを決定する決定手段とを備えたことを特徴とする。

## 【 0 0 1 1 】

本発明の処理装置の制御方法は、稼動状況に関する情報を含む第1のログ、及び、エラーに関する情報を含む第2のログを送信する外部装置と通信可能な処理装置の制御方法であって、前記外部装置から送信された前記第1のログを受信する受信工程と、前記受信工程によって受信された第1のログから求められた前記外部装置の稼動状況に基づいて、当該外部装置から送信される前記第2のログを格納するか否かを決定するための優先度を判定する判定工程と、前記判定工程によって判定された優先度に基づいて、前記外部装置が送信した前記第2のログを格納するか否かを決定する決定工程とを備えたことを特徴とする。

## 【 0 0 1 2 】

本発明の制御プログラムは、稼動状況に関する情報を含む第1のログ、及び、エラーに関する情報を含む第2のログを送信する外部装置と通信可能な処理装置の制御方法を実行するための、コンピュータで読み取り可能な制御プログラムであって、受信手段が、前記外部装置から送信された前記第1のログを受信する受信ステップと、判定手段が、前記受信ステップによって受信された第1のログから求められた前記外部装置の稼動状況に基づいて当該外部装置から送信される前記第2のログを格納するか否かを決定するための優先度を判定する判定ステップと、決定手段が、前記判定ステップによって判定された優先度に基づいて、前記外部装置が送信した前記第2のログを格納するか否かを決定する決定ステップとを、前記処理装置に実行させることを特徴とする。

## 【発明の効果】

## 【 0 0 1 3 】

本発明によれば、処理装置自身のパフォーマンスを低下させることなく、外部装置から効率的にエラーログの受け付けを行うことが可能になる。

## 【発明を実施するための最良の形態】

## 【 0 0 1 4 】

以下、本発明の実施の形態について、図面を参照しながら説明する。

## 【 0 0 1 5 】

## 〔第１の実施の形態〕

## &lt; 画像形成システムの全体構成 &gt;

図１は、本発明の実施の形態に係る処理装置を含む画像形成システムの全体の構成を示すブロック図である。

## 【００１６】

この画像形成システムは、ＬＡＮ ２０１１に接続された画像形成装置６０１を有している。画像形成装置６０１は処理装置の一例であり、画像入力デバイスであるスキャナ部２０７０、画像出力デバイスであるプリンタ部２０９５、コントロールユニット２０００、及びユーザインターフェースである操作部２０１２から構成される。

## 【００１７】

スキャナ部２０７０、プリンタ部２０９５、及び操作部２０１２は、それぞれコントロールユニット２０００に接続され、コントロールユニット２０００は、ＬＡＮ ２０１１などのネットワーク伝送手段、公衆回線等に接続されている。公衆回線からはカラー画像送信を含むＧ３、Ｇ４ファックスによる送受信が可能である。

## 【００１８】

また、ＬＡＮ ２０１１には、外部処理装置の一例である他の画像形成装置６０２～６０５のほか、パーソナルコンピュータ（以下、単にＰＣと記す）１４０も接続されている。他の画像形成装置６０２～６０５も、画像形成装置６０１と同様の機器構成を持つ。

## 【００１９】

なお、図１に示す画像形成装置は、プリンタ機能、スキャナ機能の両方を有するように記載しているが、例えば、プリンタ機能だけを持つような画像形成装置やスキャナ機能やファックス機能だけを持つ画像形成装置でも構わない。

## 【００２０】

## &lt; 画像形成装置の構成 &gt;

図２は、図１中の画像形成装置の構成を示すブロック図である。

## 【００２１】

コントロールユニット２０００は、画像入力デバイスであるスキャナ２０７０や画像出力デバイスであるプリンタ２０９５と接続しこれらの動作を制御する。また、一方ではＬＡＮ ２０１１や公衆回線（ＷＡＮ）２０５１と接続することで、画像情報やデバイス情報の入出力を行うためのコントローラである。

## 【００２２】

コントロールユニット２０００は、システムバス２００７上に、ＣＰＵ ２００１、ＲＡＭ ２００２、ＲＯＭ ２００３、及びＨＤＤ ２００４を配置している。

## 【００２３】

ＣＰＵ ２００１は、装置全体の動作を制御する。ＲＡＭ ２００２は、ＣＰＵ ２００１が動作するためのシステムワークメモリであり、画像データを一時記憶するための画像メモリでもある。また、オペレーティングシステムやシステムソフトウェア、アプリケーションソフトウェアなどのプログラムを動作させるために配置されることもある。ＲＯＭ ２００３はブートＲＯＭであり、システムのブートプログラムが格納されている。また、システムプログラムやアプリケーションプログラムが格納されている場合もある。ＨＤＤ ２００４はハードディスクドライブで、システムソフトウェア、アプリケーションソフトウェア、画像データ等を格納する。また、図示していないが、小型の画像形成装置では、ＨＤＤ ２００４を搭載せずにシステムソフトウェア、アプリケーションソフトウェア等をＲＯＭ ２００３に格納して、ハードディスクレスの構成もある。

## 【００２４】

さらに、コントロールユニット２０００は、システムバス２００７上に、ログ用メモリ２１００、操作部Ｉ／Ｆ ２００６、ネットワーク部２０１０、及びモデム２０５０を配置している。

## 【００２５】

ログ用メモリ２１００は、ログを格納するためのメモリであり、エラーログや稼動状況

10

20

30

40

50

などの各種ログが格納される。なお、ログの内容については、後述する。また、ログ用メモリ 2100 を用意せずに、RAM 2002 の一部をログ用メモリとして利用するような仕組みとしてもよい。操作部 I/F 2006 は、操作部 (UI) 2012 とのインターフェース部であり、操作部 2012 に表示する画像データを操作部 2012 に対して出力する。また、操作部 2012 から操作者が入力した情報を CPU 2001 に伝える役割をする。ネットワーク部 2010 は、LAN 2011 に接続して情報の入出力を行う。モデム 2050 は、公衆回線 2051 に接続して画像情報の入出力を行う。

#### 【0026】

イメージバス I/F 2005 は、システムバス 2007 と画像データを高速で転送する画像バス 2008 を接続し、データ構造を変換するバスブリッジである。画像バス 2008 は、PCI バスなどで構成される。

10

#### 【0027】

画像バス 2008 上には、ラスタイメージプロセッサ (RIP) 2060、デバイス I/F 部 2020、スキャナ画像処理部 2080、及びプリンタ画像処理部 2090 が配置されている。

#### 【0028】

ラスタイメージプロセッサ (RIP) 2060 は、PDL コードをビットマップイメージに展開する。デバイス I/F 部 2020 は、スキャナ 2070 やプリンタ 2095 とコントローラ 2000 とを接続し、画像データの同期系 / 非同期系の変換を行う。スキャナ画像処理部 2080 は、入力画像データに対し補正、加工、編集を行う。また入力された画像がカラー原稿であるか或いは白黒原稿かを画像の彩度信号から判断しその結果を保持する機能を有する。プリンタ画像処理部 2090 は、出力画像データに対し補正、加工、編集を行う。

20

#### 【0029】

また、画像バス 2008 上には、画像回転部 2030、解像度変換部 2031、色空間変換部 2032、及び階調変換部 2033 も配置されている。

#### 【0030】

画像回転部 2030 は、スキャナ画像処理部 2080 と連携して、スキャナ 2070 からの画像読み込みを行う。解像度変換部 2031 は、画像の解像度を変換する処理を行い、色空間変換部 2032 はマトリクス演算により、画像の色空間を変換する処理を行う。階調変換部 2033 は、画像の階調を変換する処理を行い、画像圧縮部 2040 は、画像の圧縮伸長の処理を行う。

30

#### 【0031】

上記の説明においては、画像回転などの画像処理は、画像バス 2008 に接続されたデバイスで行っているが、HDD 2004 や ROM 2003 に画像処理のソフトウェアを格納しておき、RAM 2002 に読み込んで、CPU 2001 で実行するようにしてもよい。なお、実行するソフトウェアは、画像処理に限らず、画像形成装置で必要となる他の処理でも構わないのは、いうまでもない。また、ROM 2003 に格納されたプログラムは、RAM 2002 に読み込まずに実行してもよい。

#### 【0032】

また、図 2 の HDD 2004 において、ハードディスク内には、システムソフトウェアや、アプリケーションソフトウェア、画像データを格納すると記載した。しかし、これらに限らず、画像管理のためのデータなどの画像以外の一般データファイルや画像形成装置で管理するデータを格納することができる。画像形成装置で管理するデータとしては、例えば、どの利用者が、いつ、どのような機能を利用したかを記録するジョブログなどがある。

40

#### 【0033】

上記構成の画像形成システムにおいて、画像形成装置 601 が、ログ収集のサーバ機能を有する親機の画像形成装置である。また、画像形成装置 602、603、604、605 は、画像形成装置 601 にログが収集される管理対象となる子機の画像形成装置である

50

ものとする。そして、以下、親機である画像形成装置を単に親機と記し、子機である画像形成装置を単に子機と記す。なお、子機 6 0 2、6 0 3、6 0 4、6 0 5 は、H D D 2 0 0 4 を搭載していなかったり、メモリ容量が少ない場合がある。

#### 【 0 0 3 4 】

< 本実施の形態におけるログ >

次に、本実施の形態におけるログについて説明する。

#### 【 0 0 3 5 】

ログは 2 種類あり、1 つは、どの利用者が、いつ、どれだけ（使用頻度）、どの機能を利用したかを表すジョブログである。もう一つは、エラーが発生したときに障害を特定するためのエラーログである。エラーログは、エラーが発生したときにそのエラーを解析するために必要となる内部モジュールの呼び出しシーケンス情報や、エラー発生時点でのメモリのダンプ結果情報などを含む。エラーログのサイズは、このような情報を含むため大きいものとなる。ジョブログは第 1 のログの一例であり、エラーログは第 2 のログの一例である。

#### 【 0 0 3 6 】

( A ) ログの一例

( i ) ジョブログの一例

ジョブログの一例を図 3 を参照して説明する。図 3 は、ジョブログの一例を示す表形式図である。

#### 【 0 0 3 7 】

図 3 に示すように、ジョブログには、ジョブの処理が行われた日付の情報 5 0 1、時間の情報 5 0 2、誰が処理を実行したかを表すユーザ識別子 5 0 3 が記録されている。更に、どのような処理を実行したのかを示すタイプ情報 5 0 4 が記載される。タイプ情報 5 0 4 には、例えば、P C 1 4 0 などからアプリケーションの印刷を実行した P r i n t や、画像形成装置の複写機能（以下、コピー機能と記す）を実行した C o p y、スキャンして画像を読み取る S c a n、ファクシミリ送信を行う F A X などがある。

#### 【 0 0 3 8 】

また、ジョブログには、処理に要したページ数情報 5 0 5 が記録されている。さらに、そのページ数の中で、白黒で処理したページ数情報 5 0 6、カラーで処理したページ数情報 5 0 7、及び両面処理されたページ数情報 5 0 8 などが記録されている。

#### 【 0 0 3 9 】

例えば、図 3 における 5 0 9 の事項では、ユーザ「 5 0 2 5 1 」が、「 2 0 0 7 年 1 0 月 5 日 8 時 2 5 分 4 1 秒 」に、コピー機能を実行したことを表している。そのときに 5 4 枚コピーを行い、そのうち 3 7 枚が白黒、1 7 枚がカラーであり、2 7 枚が両面処理されていることが記録されている。その他も同様である。

#### 【 0 0 4 0 】

本実施の形態では、ジョブログに含まれる情報として 9 種類の項目（ 5 0 1 ~ 5 0 9 ）を挙げているが、この限りでなく他の項目を含んでよいのは勿論のこと、この中のいくつかの項目だけを選択してジョブログとしても構わない。

#### 【 0 0 4 1 】

( i i ) エラーログの一例

エラーログの一例を図 4 を参照して説明する。図 4 は、エラー時の画像形成装置における動作状況を記録したエラーログの一例を示す図である。

#### 【 0 0 4 2 】

エラーログのヘッダー部 2 4 0 2 は、画像形成装置の機種名「 I R X X X X」、システムソフトウェアのバージョン「 C O N T v e r 0 . 4 2」、及びエラーログ情報取得時の日付などで構成される。

#### 【 0 0 4 3 】

動作手順は、図 4 における 2 4 0 3、2 4 0 5 などに示されるように、時間、ブロックの場所、動作内容、などが読み取れる。これらの情報を基に、エラー発生の原因などを解

10

20

30

40

50

析することができる。また、エラーレベルによって、例えば、簡易的な情報（２４０１）や、詳細な情報（２４０４）など情報量の異なるエラーログの生成が行える。

#### 【００４４】

##### （Ｂ）ログの収集

ログ収集について、具体的に図５を参照して説明する。図５は、親機６０１によって行われるログ収集を示す概念図である。

#### 【００４５】

子機６０２～６０５は、ログを親機６０１へ送付する。なお、このときにログ自体を親機６０１へ送付するのではなく、ログ送付依頼を先へ送付して、送付が許可されたときにログ本体を送るような仕組みとしてもよい。図５の例では、子機６０２～６０５が、それぞれログ６０６、６０７、６０８、６０９を親機６０１へ送付している。この例では、子機を４台としたが、子機が４台に限定されるものでない。

10

#### 【００４６】

また、親機６０１自体も画像形成装置として機能するため、ログを持っているので、親機６０１も自身のログ６１０を収集することになる。

#### 【００４７】

なお、図５の例では、各子機６０２～６０５から親機６０１へログを送信しているが、親機６０１が、子機６０２～６０５からログを収集するような仕組みとしても構わない。例えば、前述したジョブログの場合には、親機６０１が定期的に子機６０２～６０５に対してログを収集するようにしてもよい。

20

#### 【００４８】

子機を親機の管理対象にする設定の方法は、例えば、親機６０１に予め登録しておく方法や、子機が新規にネットワークに接続された場合に、ブロードキャストで自身の存在を親機に通知するなどの公知の方法を用いる。

#### 【００４９】

##### < 親機６０１の動作 >

以下、親機６０１の動作について説明する。

#### 【００５０】

親機６０１は、１台以上の子機を管理するための、管理機能を持ち、ログを収集するサーバとして動作する。なお、本実施の形態では、管理機能として、ログ収集機能のみしか記載していないが、その他の管理機能を有していても構わない。

30

#### 【００５１】

ログとしては、前述したように、ジョブログとエラーログの収集を行う。ジョブログは、定期的に収集されており、定められた時間になると、親機６０１が、親機６０１のネットワーク部２０１０を用いてＬＡＮ２０１１を経由して管理対象の子機６０２～６０５に接続する。

#### 【００５２】

子機６０２～６０５では、ログ用メモリ２１００に格納されているジョブログをネットワーク部２０１０を用いてＬＡＮ２０１１を経由して親機６０１に送信する。親機６０１は、受信したジョブログをＨＤＤ２００４の所定の格納場所に書込む。親機６０１は、管理対象の全ての子機６０２～６０５に対して、同様にジョブログの収集を行い、ＨＤＤ２００４に書込む。更には、親機６０１自身のジョブログも、親機６０１のログ用メモリ２１００から読み出し、ＨＤＤ２００４の所定の格納場所に書込む。

40

#### 【００５３】

このようにして、定期的に収集された全ての管理対象の子機６０２～６０５と親機６０１自身のジョブログが、親機６０１のＨＤＤ２００４に保存されている。

#### 【００５４】

##### （Ａ）ジョブログを収集したときの動作

##### （ｉ）各子機の稼動状況を求める

子機６０２～６０５からジョブログを収集したときに、親機６０１は、各子機６０２～

50



605の稼動状況を求める。以下、図5及び図6を参照して、収集されたジョブログから稼動状況を求める一例について説明する。4台の子機602～605のジョブログの一例を示す表形式図である。

【0055】

図6における701、702、703、704に示すものは、それぞれ4台の子機602、603、604、605から取得された親機601のHDD2004に格納されているジョブログである。ここで、本実施の形態では、説明を簡単にするために、1日分のジョブログのみを記載している。当然であるが、親機601のHDD2004に格納されるジョブログは、1日分だけでなく、複数日分格納されていても構わない。また、そのような場合は、稼動状況を求めるために、格納される子機のジョブログを1日分（或いは、所定の日数分）だけ用いるようになっていても構わない。

10

【0056】

子機の稼動状況の求め方は、例えば、ユーザが子機を利用した回数で判断する。701のジョブログから、子機602は、「2007年10月5日の8時25分41秒」、「11時21分43秒」、「14時11分52秒」、「15時12分32秒」、「17時45分53秒」、「19時36分25秒」の合計6回、利用されていることが分かる。同様に、702のジョブログから子機603は2回、703のジョブログから子機604は2回、704のジョブログから子機605は4回の利用がそれぞれあったことが分かる。

【0057】

(iii) 稼動状況から優先度を算出

20

親機は、ジョブログから求められた稼動状況に基づいて、子機の優先度を算出する。優先度の算出として、例えば、各利用回数を最大の利用回数で割って、100を掛けるようなものがある。子機602の場合は、

【0058】

【数1】

$$a_{602} = \frac{6}{\max\{6,2,2,4\}} \times 100 = 100$$

(maxは、集合の中の最大値を表す)

【0059】

30

となり、優先度は100となる。同様に、子機603、604は、利用回数が2であるので、優先度が33.3、子機605は、利用回数が4であるので、優先度が66.7となる。

【0060】

優先度の求め方は、このような算定式だけでなく、他の算定式でも構わないことはいうまでもない。また、利用回数で判定するだけでなく、例えば、合計処理枚数で判定するようにしてもよい。合計処理枚数は、子機602、603、604、605それぞれ215、27、376、148となるので、先の算出式を用いれば、子機602の優先度は、

【0061】

【数2】

40

$$a_{602} = \frac{215}{\max\{215,27,376,148\}} \times 100 = 57.2$$

【0062】

となる。同様に、子機603、604、605の優先度は、それぞれ7.2、100、39.4となる。

【0063】

また、処理の種別毎に重み付けを変えてもよい。例えば、PC140からアプリケーションの印刷を実行したPrintの重み付けを1、画像形成装置のコピー機能を実行したCopyの重み付けを1.3とする。また、スキャンして画像を読み取るScanの重み

50

付けを 0.9、ファクシミリ送信を行う FAX の重み付けを 1.1 とする。その結果、子機 602、603、604、604 の重み付け後の枚数は、それぞれ 262.5、28.2、421、149.8 となるので、先の算出式を用いれば、子機 602 の優先度は、  
 【0064】  
 【数3】

$$a_{602} = \frac{262.5}{\max\{262.5, 28.2, 421, 149.8\}} \times 100 = 62.4$$

【0065】

となる。同様に、子機 603、604、605 の優先度は、それぞれ 6.7、100、35.6 となる。 10

【0066】

本実施の形態では、ジョブログの情報だけから優先度を求めているが、子機のスペック（例えば、単位時間処理枚数）やログ用メモリ容量などを基準として、優先度に重み付けをするようにしてもよい。例えば、単位時間処理枚数が予め定めた基準より少ない場合や、ログ用メモリ容量が予め定めた基準より少ない場合に、優先度を高くする重み付けを与えるようなことを行ってもよい。

【0067】

(iii) 優先度テーブルを更新

このようにして、求めた優先度で優先度テーブルを更新する。ここで、優先度テーブルの一例について、図7を参照して説明する。図7は、優先度テーブルの一例を示す図である。 20

【0068】

図7に示すように、優先度テーブルは、子機を特定するフィールド801と、子機の優先度を表す優先度フィールド802から成る。

【0069】

図7の例では、フィールド801には、子機を特定する名称が記載されている。なお、子機を特定する情報は、名称に限らず、例えばIPアドレスなどで、子機を一意に特定できる情報であればよい。ここで、図7中の803のデバイス名称Aは、図5中の子機602を表す。同様に、804のデバイス名称Bは子機603、805のデバイス名称Cは子機604、806のデバイス名称Dは子機605をそれぞれ表すものとする。 30

【0070】

図7の優先度フィールド802には、先に説明した利用回数による優先度がデバイス名称A～Dに対応してそれぞれ記載されている。

【0071】

なお、優先度テーブル自体の新規作成や、子機のフィールド追加や削除については、本発明の本質的なものでない。しかし、例えば、ジョブログとして親機601のHDD2004に記録があるものを優先度テーブルのフィールド802に追加して、一定期間存在していないものは削除するような仕組みとしてもよい。

【0072】

(iv) 処理の流れ

次に、上述した、親機601が子機602～605からジョブログを収集して優先度テーブルを更新する処理の流れについて、図8のフローチャートを参照して説明する。

【0073】

図8は、ジョブログを収集して優先度テーブルを更新する処理の流れを示すフローチャートである。図8の各ステップは、親機601が備えるCPU2001が、ROM2003に格納されたプログラムをRAM2002に展開して実行される。

【0074】

まずS901において、CPU2001は、各子機602～605からジョブログを収集する。次のS902では、CPU2001は、S901で収集した各子機602～60 50

5 のジョブログから各子機 6 0 2 ~ 6 0 5 の稼動状況を求める。稼動状況とは、例えば子機 6 0 2 において利用回数であれば、処理が行われた回数である「6」がジョブログ 7 0 1 から求められる。また、合計処理枚数を稼動状況とするのであれば、ジョブログ 7 0 1 の各処理の処理枚数を合計して、子機 6 0 2 の稼動状況は、「2 1 5」とする。

【0 0 7 5】

S 9 0 3 において、CPU 2 0 0 1 は、各子機 6 0 2 ~ 6 0 5 の稼動状況から優先度を求める。これは、例えば、先の説明では、各稼動状況の値を全ての子機 6 0 2 ~ 6 0 5 の稼動状況の最大値で割ったものに 1 0 0 を掛けることで優先度を求めている。本実施の形態で説明した、S 9 0 2、S 9 0 3 での稼動状況、及び優先度の求め方は一例であり、他の算出の仕方でも構わない。

10

【0 0 7 6】

S 9 0 4 において、CPU 2 0 0 1 は、優先度テーブルの各子機 6 0 2 ~ 6 0 5 の優先度を S 9 0 3 で求められた優先度に置き換える。

【0 0 7 7】

(B) エラーログが送付されたときの動作

次に、子機 6 0 2 ~ 6 0 5 から親機 6 0 1 に対してエラーログが送付されたときの親機 6 0 1 の動作について説明する。

【0 0 7 8】

(i) エラーログ受け付けの可否判定の詳細

以下、子機 6 0 2 ~ 6 0 5 からエラーログが送付されたときに、親機 6 0 1 がエラーログを受け付けるかどうかを判定する動作について、図 9 を参照して具体的に説明する。図 9 は、親機 6 0 1 のエラーログ受け付け可否判定動作を説明する概念図であり、図 5 と共通する要素には同一符号を付与する。

20

【0 0 7 9】

・親機 6 0 1 が処理を行っていない場合

まず、親機 6 0 1 が処理を行っていない場合について説明する。この場合、エラーログの格納要求 1 3 0 1 が子機 6 0 2 から親機 6 0 1 へ送付されると、親機 6 0 1 は、自身のジョブログ 6 1 0 から、使用の集中が予想される時間であるかどうかを判断する。

【0 0 8 0】

図 1 0 は、親機 6 0 1 自身のジョブログ 6 1 0 の一例を示す表形式図である。

30

【0 0 8 1】

このジョブログ 6 1 0 は、1 時間毎に 8 時台 (図 1 0 の 1 0 0 1)、9 時台 (1 0 0 2)、1 0 時台 (1 0 0 3)、1 1 時台 (1 1 0 4)、1 5 時台 (1 0 0 5)、1 7 時台 (1 0 0 6)、及び 1 9 時台 (1 0 0 7) と分割されている。

【0 0 8 2】

この分割された時間の中で、稼動状況を求める。稼動状況の求め方は、先に示した利用回数であったり、合計処理枚数などである。この求めた稼動状況に対して閾値を設けて、それ以上の場合は、使用の集中が予想される時間と判断する。

【0 0 8 3】

例えば、稼動状況を利用回数として、閾値を 3 回とすると、図 1 0 の例では、8 時台、9 時台にエラーログの格納要求が子機 6 0 2 からあった場合は、エラーログの書込みを延期する。

40

【0 0 8 4】

本実施の形態では、1 日のジョブログとしているが、親機 6 0 1 自身のジョブログ 6 1 0 の格納がより多くあった場合には、1 日だけに限らず、日にち毎に各時間の稼動状況を求めて、更に各時間の平均の稼動状況を算出して、それを利用するようにしてもよい。

【0 0 8 5】

また、本実施の形態のように 1 時間単位とは限らず、例えば、午前、午後で分けるとか、9 時以前、9 時から 1 2 時、1 2 時から 1 3 時、1 3 時から 1 7 時、1 7 時以降のようにオフィスの勤務状況に合わせたりしてもよい。更には、時間で区切るのではなく、エラー

50

ログの格納要求時点から１時間以内などのようにしてもよい。

【００８６】

親機６０１の使用が予想されるときに、子機６０２からのエラーログ格納要求を延期すると記載した。しかし、エラーログ自体が致命的なエラーであったり、子機６０２内のエラーログ格納用のログ用メモリ２１００がメモリフルの状態になっている（或いは、メモリフルになりかけている）ときなど緊急にエラーログを格納する必要がある。

【００８７】

そのような場合には、次の処理を行う。図示しないが、エラーログ格納要求内に設けられた緊急用のフラグが立つことで、親機６０１は、エラーログの書込みが必要なことであると判断する。親機６０１は、エラーログの送信を子機６０２へ要求して、子機６０２のエラーログを親機６０１のＨＤＤ２００４へ書込む。

10

【００８８】

また、例えば、子機６０２で紙詰まり（ジャム）が頻発する場合などは、エラーログもすぐに埋まってしまうので、次の処理を行う。予め定めた回数以上のジャムが発生すると、子機６０２で、先の緊急用フラグを立ててエラーログの格納要求を送信する。これにより、エラーログによってログ用メモリ２１００のメモリフルが発生する前に親機６０１にエラーログを書込むことができる。

【００８９】

エラーログを親機６０１６０２へ書込むことができた子機６０２は、ログ用メモリ２１００をクリアする。なお、ログ用メモリ２１００にエラーログ以外のログが格納されている場合は、格納したエラーログだけを削除する。

20

【００９０】

なお、本実施の形態の説明では、エラーログの格納要求をまず親機６０１に送信しているが、子機６０２からダイレクトにエラーログを親機６０１へ送信してもよい。また、親機６０１において機能が実行されていない状態で使用の集中が予想されるときに、上記の緊急用フラグが立っていない場合であっても、次のような子機からのエラーログの格納要求を受け付けるようにしても構わない。即ち、子機の優先度に基づいて、一定の閾値以上の優先度を持つ子機からのエラーログ格納要求を受け付けるようにしても構わない。

【００９１】

・親機６０１が機能を実行している場合

30

次に、親機６０１が機能を実行している場合について具体的に説明する。この場合、先に説明した優先度テーブルから取得した子機の優先度を親機６０１の状態によって補正し、補正された優先度が予め定めた閾値以上であれば、親機６０１は、エラーログの書込みを行う。

【００９２】

ここで、説明のために、図１１の優先度テーブルと、図１２の閾値及び各種補正值を用いることにする。図１１は、優先度テーブルの一例を示す図であり、図１２（ａ）、（ｂ）は、優先度の閾値と各種補正值の一例を示す図である。

【００９３】

図１１中の１１０１に示すデバイス名称Ａは、図５中の子機６０２を表す。同様に、１１０２に示すデバイス名称Ｂは子機６０３、１１０３に示すデバイス名称Ｃは子機６０４、１１０４に示すデバイス名称Ｄは子機６０５をそれぞれ表すものとする。

40

【００９４】

図１２（ａ）において、閾値は、補正された優先度がこの値以上であれば、エラーログの書込みを認めるという境界値である。再送用補正值については後で説明する。パネル操作補正值は、親機６０１の操作パネル（操作部２０１２）を使用しているときに優先度を補正するための値である。即ち、操作パネルが利用されているということは、親機６０１の前で利用者が使っているため、親機６０１の処理をより優先させるために優先度を補正する。

【００９５】

50

図 1 2 ( b ) に示すジョブ種類別補正テーブルは、親機 6 0 1 が処理中のジョブの種類によって優先度の補正を行うためのものである。これは、例えば、カラーデータの処理を行うときには白黒データに比べてデータ量が多いため、親機 6 0 1 の C P U 2 0 0 1 に負担が大きいなど、ジョブの種類に応じて優先度を変えるためのテーブルである。

【 0 0 9 6 】

このジョブ種類別補正テーブルは、ジョブ種類を表すフィールド 1 2 0 5 と、その補正値を表すフィールド 1 2 0 6 から成る。図 1 2 ( b ) の例では、カラージョブに対する補正値が - 1 5、白黒ジョブの補正値が - 5、F A X ジョブの補正値が - 2 0、プリントジョブの補正値が - 1 0 である。また、1 枚の用紙に複数ページを面付けして印刷する N i n 1 のジョブに対する補正値は - 5 である。

10

【 0 0 9 7 】

なお、図 1 2 ( b ) に示すジョブ種類別補正テーブルは、一例であり、ジョブ種類はこれらに限定されることはないし、これらを含まなくてもよい。

【 0 0 9 8 】

以下、J C T ( X ) は、ジョブ種類 X に対する補正値を表すものとする。例えば、J C T ( F A X ) は、ジョブ種類別補正テーブルから - 2 0 となる。なお、X の部分は 1 つに限らず複数併記することができ、その場合は、補正値の合計となる。例えば、J C T ( C o l o r , N i n 1 ) は、C o l o r の補正値 - 1 5 と N i n 1 の補正値 - 5 の合計 - 2 0 となる。

【 0 0 9 9 】

20

子機の優先度を とすると、補正された優先度 ' が、閾値 以上であればエラーログの書込みを親機 6 0 1 は行う。具体的に例を挙げて説明する。

【 0 1 0 0 】

子機 6 0 2 ( 図 1 1 中のデバイス名称 A ) からエラーログの格納要求が親機 6 0 1 に対して送られると、親機 6 0 1 は、優先度テーブルから当該子機 6 0 2 の優先度 6 0 2 ( = 1 0 0 ) を取得する。この例では、1 0 0 である。

【 0 1 0 1 】

このとき、親機 6 0 1 で操作パネルを利用してカラーコピーを実行している場合は、パネル操作補正値 = - 2 0、カラーのジョブ補正値 J C T ( C o l o r ) = - 1 5 であるので、

30

$$\alpha'_{602} = \alpha_{602} + p + JCT(Color) = 100 - 20 - 15 = 65 < \theta$$

となり、補正された優先度

$$\alpha'_{602}$$

は 6 5 となり、閾値 ( = 7 0 ) を下回っているため親機 6 0 1 にエラーログの格納は行われない。

【 0 1 0 2 】

なお、先に説明したように、緊急時のエラーログについては、優先的に書込まれる。これは、以下の説明においても明記はしないが同様である。一方、同じく操作パネルを利用しても、白黒コピーを実行している場合は、白黒のジョブ補正値 J C T ( B W ) = - 5 であるので、

40

$$\alpha'_{602} = \alpha_{602} + p + JCT(BW) = 100 - 20 - 5 = 75 > \theta$$

【 0 1 0 3 】

であり、閾値 を上回るので、親機 6 0 1 にエラーログは書込まれる。また、親機 6 0 1 の操作パネルを使わずに P C 1 4 0 などから F A X 送信されている場合は、F A X のジョブ補正値 J C T ( F A X ) = - 2 0 であるので、

$$\alpha'_{602} = \alpha_{602} + JCT(FAX) = 100 - 20 = 80 > \theta$$

【 0 1 0 4 】

となり、エラーログの格納要求を親機 6 0 1 は受け付けるが、親機 6 0 1 の操作パネルから F A X 送信した場合では、パネル操作補正値 が追加されるので、

$$\alpha'_{602} = \alpha_{602} + p + JCT(FAX) = 100 - 20 - 20 = 60 < \theta$$

50

## 【 0 1 0 5 】

となり、エラーログの格納要求を親機 6 0 1 は受け付けない。

## 【 0 1 0 6 】

更には、例えば、子機 6 0 5 ( 図 1 1 中のデバイス名称 D ) は、優先度が 8 0 である。パネル操作補正値が - 2 0 であることから、親機 6 0 1 のパネル操作が行われている場合には、それだけで優先度が 6 0 となり、エラーログの格納要求が受け付けられない。

## 【 0 1 0 7 】

しかしながら、親機 6 0 1 に P C 1 4 0 などからプリントジョブが送信されてきて、それを処理しているだけであれば、P r i n t のジョブ補正値 J C T ( P r i n t ) = - 1 0 であるので、

$$\alpha'_{605} = \alpha_{605} + JCT(Print) = 80 - 10 = 70 \geq \theta$$

10

## 【 0 1 0 8 】

であり、閾値と同じ 7 0 でエラーログ格納要求を親機 6 0 1 は受け付ける。

## 【 0 1 0 9 】

なお、これまでの説明で用いた例は、各子機 6 0 2 ~ 6 0 5 からの初回のエラーログ格納依頼とする。エラーログ格納依頼が再送された場合の処理については、後述する。

## 【 0 1 1 0 】

このように、各子機 6 0 2 ~ 6 0 5 の優先度が異なるので、親機 6 0 1 が同一の状態であっても、エラーログの格納要求の受け付けの可否が動的に変わる。また、同一の子機からのエラーログ格納要求であっても、親機 6 0 1 の状態によって、エラーログの格納要求の受け付けの可否が動的に変わる。

20

## 【 0 1 1 1 】

・エラーログの格納を受け付けなかった場合

次に、子機 6 0 2 ~ 6 0 5 からのエラーログの格納を親機 6 0 1 が受け付けなかった場合について説明する。

## 【 0 1 1 2 】

親機 6 0 1 は、先に図 1 0 を参照して例示したように、自身のジョブログ 6 1 0 から使用の集中が予想される時間について算出することができる。親機 6 0 1 が、その時点でのエラーログの格納を受け付けず、延期させる子機に対して、使用の集中が無いと予想される時間での再送を指示する。

30

## 【 0 1 1 3 】

延期の指示を受けた子機は、指定された時間になると、不図示であるエラーログ格納要求に再送であることを示す再送フラグを立てて、再度、親機 6 0 1 へエラーログ格納要求を送付する。

## 【 0 1 1 4 】

親機 6 0 1 では、エラーログ格納要求を受け取り、再送フラグが立っていることを確認すると、優先度を補正するときに、先の図 1 2 ( a ) の再送用補正値 を補正値として加える。例えば、先の例で、親機 6 0 1 で操作パネルを利用して、カラーコピーを実行している場合の初回のエラーログ格納要求における補正後の優先度は、6 5 であったが、同じ状況で再送の場合は、再送用補正値 が 1 4 5 であるので、

40

$$\alpha'_{602} = \alpha_{602} + \pi + \rho + JCT(Color) = 100 + 145 - 20 - 15 = 210 > \theta$$

## 【 0 1 1 5 】

となり、エラーログの格納要求は、親機 6 0 1 で受け付けられる。

## 【 0 1 1 6 】

図 1 2 の例で、再送用補正値 を 1 4 5 としたのは、どのような場合でも再送されたエラーログ格納要求を受け付けさせるためである。つまり、補正されていない優先度がゼロ以上であるとする、

$$\begin{aligned}
 \alpha' &= \alpha + \pi + \rho + JCT(\text{Color}, BW, FAX, \text{Print}, \text{Nml}) \\
 &> \pi + \rho + JCT(\text{Color}, BW, FAX, \text{Print}, \text{Nml}) \\
 &= \pi - 20 + (-15 - 5 - 20 - 10 - 5) \\
 &= \pi - 75 \geq \theta = 70 \\
 \pi &\geq 70 + 75 = 145
 \end{aligned}$$

【 0 1 1 7 】

から、再送用補正值 が 1 4 5 以上であれば、常に閾値 ( 7 0 ) を上回る。

【 0 1 1 8 】

この例では、常にエラーログ格納要求が再送時に受け入れられるようにしたが、その限りでなくてもよい。より低い値で、一定以上の(補正された)優先度を持つものでないと、再送された場合でも受け付けなくしてもよい。或いは、再送用補正值 を固定値でなく、例えば再送回数によって値を変えるような仕組みでもよい。

10

【 0 1 1 9 】

一例として、再送用補正值 を

$$\pi = 10 \times n$$

( $n$ は再送回数)

【 0 1 2 0 】

として、子機 6 0 4 ( 図 1 1 中のデバイス名称 C ) からエラーログの格納要求が、親機 6 0 1 へ送付され、その格納要求が 1 度目の再送であったとする。このとき、親機 6 0 1 では、P C 1 4 0 などから F A X 送信が実行されていると、子機 6 0 4 の優先度が 6 5 であるから、

20

$$\alpha'_{604} = \alpha_{604} + \pi + JCT(\text{Print}) = 65 + (15 \times 1) - 20 = 60 < \theta$$

【 0 1 2 1 】

となり、エラーログ格納要求は受け付けられず、親機 6 0 1 から再度、子機 6 0 4 に対してエラーログ格納要求の延期が要求される。2 回目の再送を子機 6 0 4 が行ったときに、親機 6 0 1 で操作パネル上で白黒コピーが行われていたとすると、

$$\alpha'_{604} = \alpha_{604} + \pi + \rho + JCT(BW) = 65 + (15 \times 2) - 20 - 5 = 70 = \theta$$

【 0 1 2 2 】

となり、エラーログ格納要求は受け入れられて、親機 6 0 1 は、子機 6 0 5 のエラーログを書込む。

【 0 1 2 3 】

30

2 回目の親機 6 0 1 は、操作パネルが使われており、再送用補正值を除いた補正值が、1 度目の再送時の - 2 0 と比べて - 2 5 と大きく、エラーログ格納要求を受け入れづらい状態である。それに関わらず、且つ元々の優先度が閾値を下回るような子機 6 0 5 のエラーログ格納要求であっても、再送回数が多いことで受け入れることが可能となる。なお、この場合は、前述したエラーログ格納要求に再送であることを示す再送フラグを設けるだけでなく、再送回数を示す値をエラーログ格納要求に入れることになる。

【 0 1 2 4 】

本実施の形態では、親機 6 0 1 自身のジョブログ 6 1 0 を参照して、使用の集中が予想されない時間の再送を子機に対して行うように説明しているが、そのような時間を指示しないで、予め定められた一定時間後に再送するように指示してもよい。或いは、親機 6 0 1 で時間を指定しないで、子機が一定時間経過したら再送するような仕組みでも構わない。更には、再送用補正值 を無くして ( = 0 ) として、子機の優先度と親機 6 0 1 のエラーログ格納要求時の状態だけで補正された優先度を求めて判定するようにしても構わない。

40

【 0 1 2 5 】

( i i ) エラーログ受け付けの可否判定処理の流れ

次に、親機 6 0 1 が子機のエラーログ格納要求を受け付けるか否かの決定を行う際の処理の流れについて、図 1 3 を参照して説明する。図 1 3 は、第 1 の実施の形態におけるエラーログ受け付けの可否判定処理の流れを示すフローチャートである。図 1 3 の各ステップは、親機 6 0 1 が備える C P U 2 0 0 1 が、R O M 2 0 0 3 に格納されたプログラムを

50

R A M 2 0 0 2 に展開して実行される。

【 0 1 2 6 】

まず S 1 4 0 1 において、C P U 2 0 0 1 は、優先度テーブルからエラーログ格納要求のあった子機の優先度を取得して、S 1 4 0 2 へ進む。S 1 4 0 2 では、C P U 2 0 0 1 は、エラーログ格納要求を確認して、再送されたエラーログを示すフラグが立っている場合には、S 1 4 0 3 へ進み、そうでなければ S 1 4 0 4 へ進む。

【 0 1 2 7 】

S 1 4 0 3 では、C P U 2 0 0 1 は、子機からのエラーログ格納要求が、再送されたエラーログ格納要求であったので、S 1 4 0 1 で取得した優先度を再送用補正值を用いて補正して、S 1 4 0 4 へ進む。

10

【 0 1 2 8 】

S 1 4 0 4 では、C P U 2 0 0 1 は、親機 6 0 1 自身が複合機としての機能を実行しているかを判定する。実行していれば S 1 4 0 5 へ進み、実行していなければ S 1 4 1 3 へ進む。S 1 4 0 5 において、C P U 2 0 0 1 は、親機 6 0 1 が自身の操作パネルから機能を実行しているかどうかを判定する。操作パネルから機能を実行していれば、S 1 4 0 6 へ進む。

【 0 1 2 9 】

S 1 4 0 6 において、C P U 2 0 0 1 は、子機の優先度を、操作パネルが操作されているとして補正、つまりパネル操作用に補正を行う。S 1 4 0 5 で操作パネルから実行されていないのであれば S 1 4 0 7 へ進む。これは、例えば、P C 1 4 0 などからジョブが投入された場合などである。S 1 4 0 7 では、C P U 2 0 0 1 は、親機 6 0 1 が画像形成装置で実行中のジョブの種類に応じて、子機の優先度の補正を行う。

20

【 0 1 3 0 】

一方、S 1 4 1 3 は、S 1 4 0 4 で親機 6 0 1 が複合機としての機能を実行していない場合に実行される。S 1 4 1 3 では、C P U 2 0 0 1 は、親機 6 0 1 が自分自身のジョブログを確認して、現在の時間が親機 6 0 1 の使用が集中する時間であるかどうかを判定する。使用の集中が予想される場合は、S 1 4 0 8 へ進み、使用の集中が予想されない場合は、S 1 4 1 4 へ進んで、親機 6 0 1 にエラーログを書込んで終了する。

【 0 1 3 1 】

S 1 4 0 8 において、C P U 2 0 0 1 は、エラーログ格納要求がハード的に動作しないなどの致命的なエラーかどうかを判定する。これは、エラーログ格納要求中にフラグなどを準備して判断することができるようにする。致命的なエラーログの場合は、S 1 4 1 4 へ進む。致命的なものでない場合は、S 1 4 0 9 へ進む。

30

【 0 1 3 2 】

S 1 4 0 9 において、C P U 2 0 0 1 は、子機のログ用メモリ 2 1 0 0 がメモリフルになっているかを判定する。これも、エラーログ格納要求中にフラグなどを準備して判断できるようにする。メモリフルの場合は、S 1 4 1 4 へ進み、そうでない場合は、S 1 4 1 0 へ進む。

【 0 1 3 3 】

S 1 4 1 0 において、C P U 2 0 0 1 は、エラーログ格納要求が、ジャムが一定回数発生した後のエラーログであるかどうかを判定する。これも、エラーログ格納要求中にフラグなどを準備して判断することができるようにする。一定回数以上のジャム発生のエラーログの場合は、S 1 4 1 4 へ進み、そうでなければ、S 1 4 1 1 へ進む。

40

【 0 1 3 4 】

S 1 4 1 1 において、C P U 2 0 0 1 は、補正された優先度が、予め定められた閾値より大きいかどうかを判定し、閾値よりも大きければ S 1 4 1 4 へ進む。閾値未満であれば、S 1 4 1 2 へ進む。

【 0 1 3 5 】

S 1 4 1 2 において、C P U 2 0 0 1 は、エラーログ格納要求を送信した子機に対して、親機 6 0 1 のジョブログを用いて稼動状況を確認する。そして、親機 6 0 1 の稼動状況

50



が低下すると予想される時間での再送を子機に指示して終了する。S 1 4 1 4において、C P U 2 0 0 1は、当該子機のエラーログを書込んで終了する。

【 0 1 3 6 】

なお、本実施の形態においては、エラーログとして説明したが、突発的に発生してサイズが大きいことなどが親機 6 0 1 の処理のパフォーマンスを劣化させるようなログであれば、エラーログでなくても構わない。

【 0 1 3 7 】

< 本実施の形態に係る利点 >

第 1 の実施の形態によれば、次のような利点を有する。

【 0 1 3 8 】

親機 6 0 1 は、通常のジョブログを定期的に子機 6 0 2 ~ 6 0 5 から取得している。そこで、ジョブログから子機の稼動状況を確認してその稼動率を算出し、稼動率の高い子機は頻繁に使われるため子機のログ用メモリ 2 1 0 0 がメモリフルになり易いと判断して、このような子機からのエラーログ格納要求を優先的に受け付ける。逆に、稼動率の低い子機は、ログ用メモリ 2 1 0 0 がメモリフルになり難いと判断して、エラーログ格納要求の受け付けの優先度を下げる。このように、管理対象の子機が多数あっても、稼動率の高い機器からのエラーログを優先的に受け付けるようにして、稼動率の低い機器からの受け付けを制限する。これにより、親機自身の画像形成装置としてのパフォーマンスを低下させることなく、効率的にエラーログの受け付けを行うことができる。このとき、単純に子機側の稼動率だけで優先度を決定するのではなく、親機 6 0 1 自身の稼動率や稼動状況（実際にユーザが機器の操作をしているかなど）を基にして優先度を決めるので、子機に対して自動的に動的な優先度の設定を行うことが可能になる。

【 0 1 3 9 】

即ち、本実施の形態では、逐次変化するジョブログを用いることにより、子機のジョブログから算出された優先度と、逐次変化する親機 6 0 1 の稼動状態とによって、子機からのエラーログ格納要求の可否を動的に判定することができる。

【 0 1 4 0 】

また、親機 6 0 1 のジョブログ 6 1 0 から親機 6 0 1 自身の稼動状況の予測を行って、子機からのエラーログ格納要求の可否を決めたり、エラーログ格納要求が拒否された子機への再送時間を決めたりすることができる。

【 0 1 4 1 】

更には、親機 6 0 1 で実行中のジョブの種類や、操作パネル上の実行かどうかによって、子機の優先度の補正を動的に行うことができる。また、エラーログ格納要求で、緊急エラーログを優先的に格納できる仕組みも実現することができる。

【 0 1 4 2 】

[ 第 2 の実施の形態 ]

第 1 の実施の形態では、エラーログ格納要求を延期する場合には、親機 6 0 1 が子機に対して再送を指示して、エラーログ格納要求を再送してもらうようにした形態について説明した。これに対して、第 2 の実施の形態では、エラーログ格納要求を延期した場合には、親機 6 0 1 の都合が良いときに、子機のエラーログを取得するようにした形態について説明する。

【 0 1 4 3 】

< エラーログ格納要求を延期した場合の動作 >

第 2 の実施の形態に係る、親機 6 0 1 がエラーログ格納要求を延期した場合の動作について、図 1 4 を参照して説明する。図 1 4 は、エラーログ格納要求を延期した場合の動作を説明するための概念図であり、図 5 と共通の要素には同一符号を付与する。

【 0 1 4 4 】

子機 6 0 2 からエラーログ格納要求 1 6 0 1 があったが、親機 6 0 1 は、このエラーログ格納要求 1 6 0 1 を拒否したとする。このとき、親機 6 0 1 は、エラーログ書込み待ちリストに子機 6 0 2 を追加する。同様に、子機 6 0 4 からエラーログ格納要求 1 6 0 2 が

10

20

30

40

50

あったが、親機 6 0 1 がこのエラーログ格納要求 1 6 0 2 を拒否すると、親機 6 0 1 はエラーログ書込み待ちリストに子機 6 0 4 を追加する。なお、エラーログ格納要求の受け付けの可否の方法は、第 1 の実施の形態で示したものとする。

【 0 1 4 5 】

エラーログ書込み待ちリストは、例えば、図 1 5 に示すようなものである。図 1 5 は、エラーログ書込み待ちリストの一例を示す図である。

【 0 1 4 6 】

図 1 5 において、1 7 0 1 は、エラーログ書込み待ちリストを特定するデバイスの名称であり、1 7 0 2、1 7 0 3 はそれぞれ、エラーログ格納要求を送付した日付と時間である。なお、第 1 の実施の形態の図 1 1 で示したものと同様に、デバイス名称 A は、図 5 中の子機 6 0 2 を表す。同様に、デバイス名称 B は子機 6 0 3、デバイス名称 C は子機 6 0 4、デバイス名称 D は子機 6 0 5 をそれぞれ表すものとする。

【 0 1 4 7 】

図 1 5 の 1 7 0 4 に示すものは、子機 6 0 4 が「2 0 0 7 年 1 0 月 5 日」の「1 1 時 2 4 分 5 5 秒」に送ったエラーログ格納要求は、親機 6 0 1 に拒否されたことを示す。1 7 0 5 に示すものは、子機 6 0 2 が「2 0 0 7 年 1 0 月 5 日」の「1 3 時 1 0 分 0 0 秒」に送ったエラーログ格納要求は、親機 6 0 1 に拒否されたことを示す。

【 0 1 4 8 】

親機 6 0 1 では、自身がコピーなど画像形成装置としての機能を実行していないとき、または自身のジョブログ 6 1 0 から機能の集中使用が予想されないときは、次の処理を行う。即ち、エラーログ書込み待ちリストを確認して、エラーログ格納要求があったが、まだエラーログを書込んでない子機からエラーログを取得する。図 1 5 の例では、子機 6 0 4 と 6 0 2 がエラーログの書込み待ちになっているが、親機 6 0 1 は、優先度テーブルを参照して優先度が高い子機からエラーログを取得する。

【 0 1 4 9 】

例えば、図 7 の優先度テーブルであったとすると、子機 6 0 4 の優先度は 6 5、子機 6 0 2 の優先度は 1 0 0 である。時間的には子機 6 0 4 よりも後にエラーログ格納要求を行った子機 6 0 2 のエラーログが先に取得される。当然ながら、単純にエラーログ格納要求した時間が早いものから順にエラーログを取得するような仕組みでも構わない。

【 0 1 5 0 】

また、複数回のエラーログ格納要求がある場合に、例えば、図 1 6 のようなエラーログ書込み待ちリストであったとする。優先度は、子機 6 0 5 (デバイス名称 D) が 8 0 で、子機 6 0 3 (デバイス名称 B) が 3 5 である。しかし、エラーログ格納要求の回数が、子機 6 0 3 からの方が多いため、親機 6 0 1 は、子機 6 0 3 のエラーログを先に取得してもよい。

【 0 1 5 1 】

また、一定回数以上のエラーログ格納要求が延期された子機がエラーログ書込み待ちリストにある場合には、その子機を優先として親機 6 0 1 が当該子機のエラーログを取得してもよい。

【 0 1 5 2 】

< エラーログ取得決定処理の流れ >

次に、親機 6 0 1 が、エラーログ書込み待ちリスト中のどの子機のエラーログを取得するのかを決定する処理の流れについて、図 1 7 及び図 1 8 を参照して説明する。

【 0 1 5 3 】

図 1 7 及び図 1 8 は、本発明の第 2 の実施の形態におけるエラーログ取得決定処理の流れを示すフローチャートであり、図 1 3 と同一処理の部分については、図 1 3 と同一符号を付与して、その説明は割愛する。なお、図 1 7 及び図 1 8 の各ステップは、親機 6 0 1 が備える CPU 2 0 0 1 が、ROM 2 0 0 3 に格納されたプログラムを RAM 2 0 0 2 に展開して実行される。

【 0 1 5 4 】

まず S 1 5 0 1 において、C P U 2 0 0 1 は、子機からのエラーログ格納要求を受信したかを判定する。受信した場合は、S 1 4 0 1 へ進み、図 1 3 のフローチャートで説明した処理を実施する。エラーログ格納要求を延期する場合に、図 1 3 のフローチャートでは、S 1 4 1 2 で子機へ再送要求を行っていたが、本フローチャートでは、S 1 5 0 2 へ進む。S 1 5 0 2 において、C P U 2 0 0 1 は、エラーログ書込み待ちリストに当該子機の情報を追加する。

【 0 1 5 5 】

S 1 5 0 1 で、親機 6 0 1 が、エラーログ格納要求を受信していない場合は、S 1 5 0 3 へ進む。S 1 5 0 3 において、C P U 2 0 0 1 は、親機 6 0 1 自身が複合機としての機能を実行しておらず、自身のジョブログ 6 1 0 から使用の集中が予想される時間帯でないかを判定する。N o の場合、即ち親機 6 0 1 が機能を実行しているか、或いは使用の集中が予想される時間帯であれば、終了する。Y e s の場合、即ち親機 6 0 1 で機能も実行されていないし、使用が集中されることも予想されないときは、S 1 5 0 4 へ進む。

【 0 1 5 6 】

S 1 5 0 4 において、C P U 2 0 0 1 は、エラーリスト書込み待ちリストを確認して、空であれば終了する。空でなければ、S 1 5 0 5 へ進む。S 1 5 0 5 において、C P U 2 0 0 1 は、エラーログ書込み待ちリストに予め定められた一定回数以上延期された子機があるかを判定する。あれば S 1 5 0 7 へ進み、無ければ S 1 5 0 6 へ進む。

【 0 1 5 7 】

S 1 5 0 6 において、C P U 2 0 0 1 は、エラーログ書込み待ちリスト中で優先度が一番高い子機を選択して、S 1 5 1 0 へ進む。なお、このとき同一優先度の複数の子機があった場合は、1 つを選択する。選択の仕方は、例えば、最初に見つかったものであっても構わない。或いは、該当する子機の中で一番最初にエラーログ格納要求があったものでも構わない。或いは、延期回数が一番多いものでも構わないし、ここに挙げた以外の方法でも 1 つを選択する方法であれば構わない。

【 0 1 5 8 】

S 1 5 0 7 において、C P U 2 0 0 1 は、エラーログ書込みリスト中で、エラーログ格納要求の延期回数が一番多い子機を選択する。次の S 1 5 0 8 において、C P U 2 0 0 1 は、S 1 5 0 7 で選択した子機が複数あるか判定する。1 つだけなら、S 1 5 1 0 へ進み、複数あれば、S 1 5 0 9 へ進む。

【 0 1 5 9 】

S 1 5 0 9 において、C P U 2 0 0 1 は、S 1 5 0 7 で選択された複数の子機の優先度が一番高いものを選択する。このとき、優先度が同一のものが複数あった場合は、先に S 1 5 0 6 で説明したような方法で 1 つの子機を選択する。S 1 5 1 0 において、C P U 2 0 0 1 は、親機 6 0 1 が選択した子機のエラーログを取得して、S 1 4 1 4 へ進む。

【 0 1 6 0 】

S 1 4 1 4 において、C P U 2 0 0 1 が、当該子機のエラーログを親機 6 0 1 に書込んだ後、S 1 5 1 1 へ進む。S 1 5 1 1 において、C P U 2 0 0 1 は、エラーログ書込み待ちリストから当該子機の項目を削除して、終了する。なお、S 1 5 0 1 で Y e s の場合は、エラーログ書込み待ちリストでなく、子機から直接要求のあったエラーログを書込む場合となる。このときに、当該子機がエラーログ書込み待ちリストになれば、当然ながら S 1 5 1 1 では、何も処理しない。

【 0 1 6 1 】

< 第 2 の実施の形態に係る利点 >

第 2 の実施の形態では、エラーログ格納要求を延期した場合には、親機 6 0 1 の都合が良いときに子機のエラーログを取得することができる。即ち、親機 6 0 1 は、エラーログ格納要求が延期された回数の多い子機のエラーログ、或いは高い優先度の子機のエラーログを、親機 6 0 1 自身の処理が行われていない合間に取得しに行くことができ、効率的なエラーログ書込み処理を行うことができる。

【 0 1 6 2 】

10

20

30

40

50

### 〔第3の実施の形態〕

第3の実施の形態においても、エラーログ格納要求を延期した場合には親機601が都合が良いときに子機のエラーログを取得する点において第2の実施の形態と同様であるが、その取得の方法が異なる。

#### 【0163】

即ち、第3の実施の形態では、親機601自身が複合機としての機能を実行しておらず、自身のジョブログ610から使用の集中が予想される時間帯でない場合に、エラーログ書込み待ちリスト中の子機の優先度を補正する。そして、補正された優先度が最も高いの子機のエラーログを取得するような方法を採用する。

#### 【0164】

<優先度補正方法>

以下、本実施の形態における優先度補正方法について説明する。

#### 【0165】

例えば、エラーログ格納要求の回数による補正値を

$$\gamma = 15 \times n$$

( $n$ は要求回数)

#### 【0166】

とすると、子機603の補正値は、 $\gamma = 15 \times 3 = 45$ となり、補正値を優先度35に加えて、80となる。一方、子機605の補正値は、 $\gamma = 15 \times 1 = 15$ となり、補正値を優先度65に加えて、95となる。子機605の方が、エラーログ格納要求の回数が少ないが、補正された優先度が大きいため、親機601は、子機605のエラーログを取得する。

#### 【0167】

また、待ち時間に応じて優先度を変えるように補正してもよい。例えば、現時点から待ち時間の分単位の経過時間を求めて、それを

$$\varepsilon = \text{time} \times 0.5$$

( $\text{time}$ は、エラーログ格納要求からの分単位での経過時間)

#### 【0168】

として、子機602～605の優先度を補正する。例えば、現在の時刻を14時丁度とすると、図15のエラーログ書込み待ちリストにおいて、子機604は、156分待ち、子機602は、50分待ちとなる。但し、秒単位は切り捨てとする。すると、子機604の補正値は、 $\gamma = 156 \times 0.5 = 78$ 、子機602の補正値は、 $\gamma = 50 \times 0.5 = 25$ となる。したがって子機604、602の優先度が65、100であるので、補正後の優先度は、それぞれ143と125となっており、親機601は、子機604のエラーログを取得する。

#### 【0169】

また、複数の同一子機のエラーログ書込み待ちがあるときは、例えば、一番長い間待っているものにするなどの方法がある。

#### 【0170】

いくつかの優先度の補正方法の例を説明したが、これらの補正方法を組み合わせて優先度を補正するようにしても構わない。また、ある一定の時間内に特定の子機からのエラーログ格納要求が一定回数以上あったときは、当該子機の優先度を上げるようにしてもよい。

#### 【0171】

<エラーログ取得決定処理の流れ>

次に、親機601が、補正された優先度を用いて、エラーログ書込み待ちリスト中のどの子機のエラーログを取得するのかを決定する処理の流れについて、図19及び図20を参照して説明する。

#### 【0172】

図19及び図20は、本発明の第3の実施の形態におけるエラーログ取得決定処理の流

10

20

30

40

50

れを示すフローチャートであり、図 13、図 17 及び図 18 と同一処理の部分については、図 13、図 17 及び図 18 と同一符号を付与して、その説明は割愛する。図 19 及び図 20 の各ステップは、親機 601 が備える CPU 2001 が、ROM 2003 に格納されたプログラムを RAM 2002 に展開して実行される。

【0173】

S1901 は、親機 601 が処理を実行しておらず、使用の集中も予想されないときに、子機からエラーログ格納要求がなく、且つエラーログ書込み待ちリストが空で無いときに実行されるステップである。S1901 において、CPU 2001 は、エラーログ書込み待ちリスト中の子機に対して、上述した優先度補正方法を用いて優先度の補正を行い、S1902 へ進む。

10

【0174】

S1902 において、CPU 2001 は、補正された優先度の中で、優先度が最も高い子機を選択して、S1510 へ進む。同一の優先度を持つ子機が複数あった場合には、先に S1506 で説明したような方法で 1 つを選択する。

【0175】

< 第 3 の実施の形態に係る利点 >

第 2 の実施の形態によれば、エラーログ格納要求を延期した場合には、親機 601 の都合が良いときに子機のエラーログを取得することができる。即ち、親機 601 は、エラーログ格納要求が延期された子機の中で高い優先度の子機のエラーログを、親機 601 自身の処理が行われていない合間に取得しに行くことができ、効率的なエラーログ書込み処理を行うことができる。

20

【0176】

[ 変形例 ]

本発明は、上記した各実施の形態に限定されず、種々の変形が可能であり、その変形例としては例えば次のようなものがある。

【0177】

(1) 上記した第 2 の実施の形態または第 3 の実施の形態は、第 1 の実施の形態で説明した再送指示を行う方式との複合でも構わない。つまり、エラーログ格納要求を延期する場合に、当該子機に対して再送指示をすると同時にエラーログ書込み待ちリストにも当該子機の情報を追加する。子機が再送もしてくるし、再送が行われる前に親機 601 が処理を実行していない且つ使用の集中が予想されない場合は、親機 601 が、エラーログ書込み待ちリスト中の子機にエラーログを取りに行くようにする。このような形態は、図 13、図 17、図 18、図 19 及び図 20 のフローチャートの組み合わせで実現できることは容易に分かる。

30

【0178】

(2) 子機のエラーログ格納要求が親機 601 に拒否されたときに、子機側では、ログ用メモリ 2100 がメモリフルである場合に、緊急用フラグを立てて、親機 601 に対して、優先的にエラーログを書込むよう依頼することを第 1 の実施の形態で説明した。このように緊急用フラグを設けずに、子機で、エラーログが取得されるまで、子機の動作できないように操作パネルなどをロックしてしまう仕組みとしてもよい。

40

【0179】

(3) また、子機側の処理として、次のような処理を行ってもよい。即ち、同一のエラーが発生している場合に、親機 601 がエラーログ格納要求を何度も拒否していると、同じようなエラーログが何度も子機側のログ用メモリ 2100 に記録されて、すぐにログ用メモリ 2100 がフルになってしまう可能性がある。そこで、同じエラーが発生して、既にログ用メモリ 2100 に記録されている場合には、同一のエラーログとして、エラーログのレベルを下げて、簡略化したエラーログをログ用メモリ 2100 に書込む。これによって、ログ用メモリ 2100 がメモリフルになるのをなるべく抑えるようにしてもよい。

【0180】

或いは、同じエラーが発生している場合は、通常よりもエラーログのレベルを上げて、

50

詳細なエラーログを取得し、ログ用メモリ 2100 に書込んで、エラーの原因を究明し易くする。同時に、ログ用メモリ 2100 のメモリフルを早期に発生させて、緊急用フラグを立てたエラーログ格納要求を送出し易くして、エラーログをなるべく早く親機 601 に書込ませるようにしてもよい。

【0181】

(4) 本実施の形態において画像形成装置として説明しているが、サービスを提供する機能を有する装置であれば、本発明の適用は画像形成装置に限定されない。例えば、サーバ機器にスキャナ、プリンタ等を接続して、画像形成装置と同等の機能を有するようにした構成を画像形成装置と読み換えても構わない。

【0182】

(5) 本発明の目的は、以下の処理を実行することによっても達成される。即ち、上述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（または CPU や MPU 等）が記憶媒体に格納されたプログラムコードを読み出す処理である。

【0183】

この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施の形態の機能を実現することになり、そのプログラムコード及び該プログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0184】

また、プログラムコードを供給するための記憶媒体としては、次のものを用いることができる。例えば、フロッピー（登録商標）ディスク、ハードディスク、光磁気ディスク、CD-ROM、CD-R、CD-RW、DVD-ROM、DVD-RAM、DVD-RW、DVD+RW、磁気テープ、不揮発性のメモリカード、ROM 等である。または、プログラムコードをネットワークを介してダウンロードしてもよい。

【0185】

また、コンピュータが読み出したプログラムコードを実行することにより、上記実施の形態の機能が実現される場合も本発明に含まれる。加えて、そのプログラムコードの指示に基づき、コンピュータ上で稼動している OS（オペレーティングシステム）等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0186】

更に、前述した実施形態の機能が以下の処理によって実現される場合も本発明に含まれる。即ち、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれる。その後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わる CPU 等が実際の処理の一部または全部を行う場合である。

【図面の簡単な説明】

【0187】

【図 1】実施の形態に係る処理装置を含む画像形成システムの全体の構成を示すブロック図である。

【図 2】図 1 中の画像形成装置の構成を示すブロック図である。

【図 3】ジョブログの一例を示す表形式図である。

【図 4】エラー時の画像形成装置における動作状況を記録したエラーログの一例を示す図である。

【図 5】親機のログ収集を示す概念図である。

【図 6】子機のジョブログの一例を示す表形式図である。

【図 7】優先度テーブルの一例を示す図である。

【図 8】ジョブログを収集して優先度テーブルを更新する処理の流れを示すフローチャートである。

【図 9】親機のエラーログ受け付け可否判定動作を説明する概念図である。

【図 10】親機自身のジョブログの一例を示す表形式図である。

【図 11】優先度テーブルの一例を示す図である。

【図 12】優先度の閾値と各種補正值の一例を示す図である。

【図 13】第 1 の実施の形態におけるエラーログ受け付けの可否判定処理の流れを示すフローチャートである。

【図 14】エラーログ格納要求を延期した場合の動作を説明するための概念図である。

【図 15】エラーログ書込み待ちリストの一例を示す図である。

【図 16】エラーログ書込み待ちリストの一例を示す図である。

【図 17】第 2 の実施の形態におけるエラーログ取得決定処理の流れを示すフローチャートである。

【図 18】図 17 の続きのフローチャートである。

【図 19】第 3 の実施の形態におけるエラーログ取得決定処理の流れを示すフローチャートである。

【図 20】図 19 の続きのフローチャートである。

【符号の説明】

【0188】

601 親機（画像形成装置）

602～605 子機（画像形成装置）

2001 CPU

2002 RAM

2003 ROM

2004 HDD

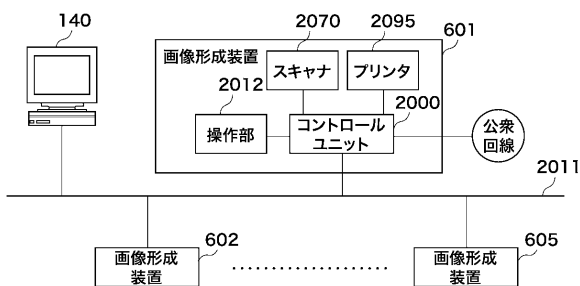
2012 操作部

2100 ログ用メモリ

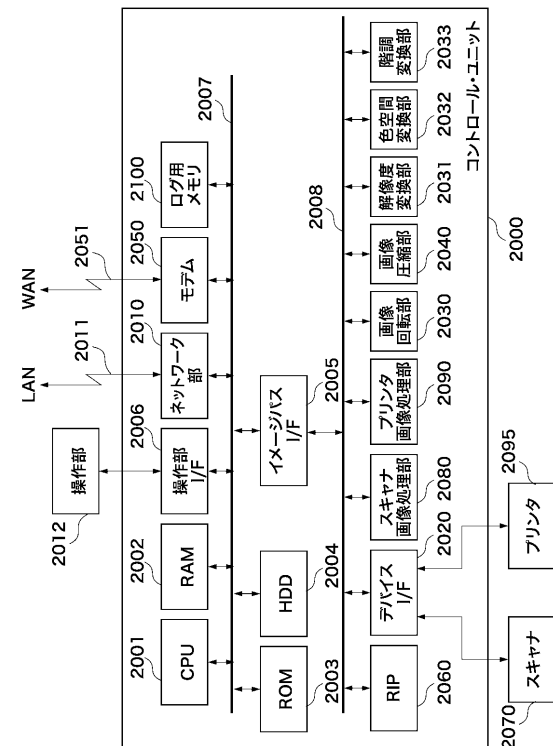
10

20

【図 1】



【図 2】



【図 3】

501	502	503	504	505	506	507	508
日付	時間	ユーザ識別子	タイプ	ページ数	白黒	カラー	画面
2007.10.5	08:25.41	50251	Copy	54	37	17	27
2007.10.5	09:33.52	10047	Print	14	0	14	0
2007.10.5	09:51.27	50251	FAX	4	4	0	0
2007.10.5	10:10.39	00844	Print	22	22	0	0
2007.10.5	10:24.16	42941	Scan	15	15	0	27
2007.10.5	10:54.52	50251	Copy	118	10	108	59
.....	.....	.....	.....	.....	.....	.....	.....

【図 4】

(a)

2402

====Log====  
・機種名 IRXXXX  
・CONT ver 0.42  
200X/06/01  
  
====Log====  
  
Time : ID  
0.000.001 OS initialize  
  
0.000.521 DE SRAM initialize  
0.003.256 DE NVRAM initialize  
  
.....  
0.007.443 FAX CCC initialize  
  
.....  
0.009.294 SCAN DDD initialize  
  
.....  
0.130.382 PRINT EEE initialize

(b)

2405

====Log====  
・機種名 IRXXXX  
・CONT ver 0.42  
200X/06/01  
  
====Log====  
  
Time : ID  
0.000.001 OS initialize  
  
0.000.521 DE SRAM initialize  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
00000000 00000000 00000000 00000000  
  
0.003.256 DE NVRAM initialize  
000000FF 0000001C 00012340 00000000  
00000000 00000000 00000000 00000000  
007771300 0BF 0000 00000000 00000000  
00000000 00000000 00000000 00000000

【図 5】

【図 6】

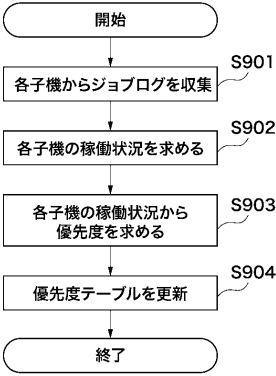
701							
日付	時間	ユーザ識別子	タイプ	ページ数	白黒	カラー	画面
2007.10.5	08:25.41	50251	Copy	54	37	17	27
2007.10.5	11:21.43	10047	Print	14	0	14	0
2007.10.5	14:11.52	50251	FAX	4	4	0	0
2007.10.5	15:12.32	00844	Print	20	20	0	0
2007.10.5	17:45.53	42941	Scan	15	15	0	0
2007.10.5	19:36.25	50251	Copy	108	10	98	54
702							
日付	時間	ユーザ識別子	タイプ	ページ数	白黒	カラー	画面
2007.10.5	10:24.33	09204	Print	15	0	15	0
2007.10.5	14:18.37	33620	FAX	12	12	0	0
703							
日付	時間	ユーザ識別子	タイプ	ページ数	白黒	カラー	画面
2007.10.5	09:39.28	26721	Copy	150	40	110	0
2007.10.5	10:19.57	00673	Print	226	226	0	113
704							
日付	時間	ユーザ識別子	タイプ	ページ数	白黒	カラー	画面
2007.10.5	09:45.41	24078	FAX	11	11	0	0
2007.10.5	12:00.05	41787	Print	46	6	40	23
2007.10.5	13:36.37	41787	FAX	7	7	0	0
2007.10.5	17:51.21	10047	Print	84	22	62	42



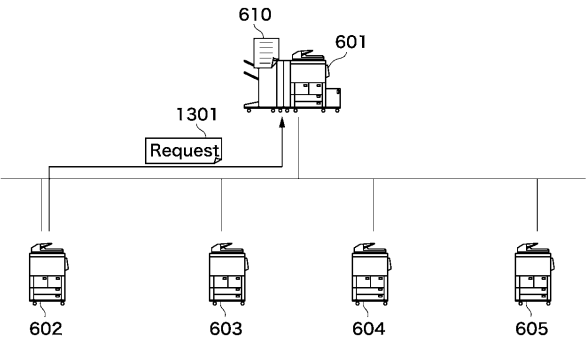
【図 7】

	801 Device	802 優先度
803	A	100
804	B	33.3
805	C	33.3
806	D	66.7

【図 8】



【図 9】



【図 10】

日付	時間	ユーザ識別子	タイプ	ページ数	白黒	カラー	両面
2007.10.5	08:25:41	50251	Copy	36	36	0	0
2007.10.5	08:43:22	50251	Print	112	0	112	66
2007.10.5	08:51:53	50251	FAX	4	4	0	0
2007.10.5	09:04:31	50251	FAX	5	5	0	0
2007.10.5	09:08:15	50251	Copy	16	0	16	0
2007.10.5	09:15:55	50251	Copy	33	33	0	0
2007.10.5	09:31:35	50251	Print	108	8	100	54
2007.10.5	09:45:12	50251	FAX	7	7	0	0
2007.10.5	10:21:43	10047	Print	78	0	78	0
2007.10.5	11:51:53	50251	FAX	4	4	0	0
2007.10.5	15:12:32	00844	Scan	10	10	0	0
2007.10.5	17:45:53	42941	Print	19	15	4	0
2007.10.5	19:36:25	50251	Copy	61	1	60	0

【図 11】

	Device	優先度
1101	A	100
1102	B	35
1103	C	65
1104	D	80

【図 12】

(a)

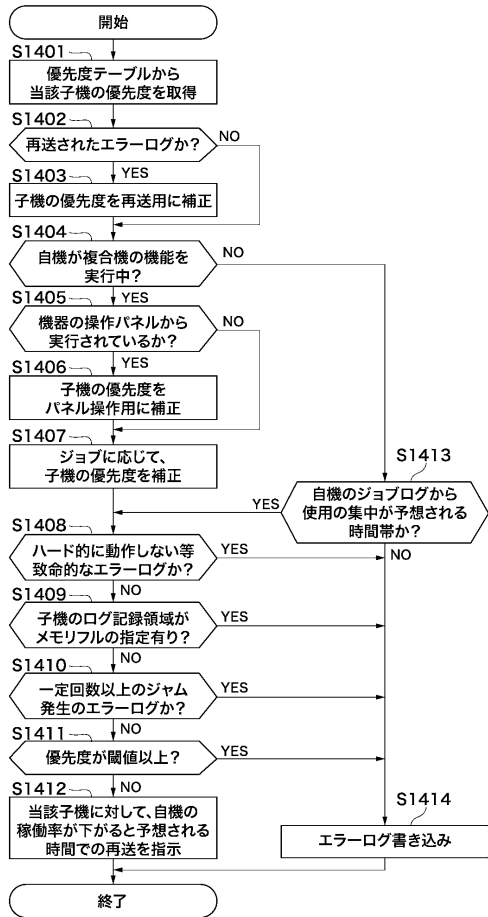
閾値  $\theta=70$   
再送用補正值  $\pi=145$   
パネル操作補正值  $\rho=-20$

(b)

ジョブ種類	補正值
Color	-15
BW	-5
FAX	-20
Print	-10
Nin1	-5

ジョブ種類別補正テーブル

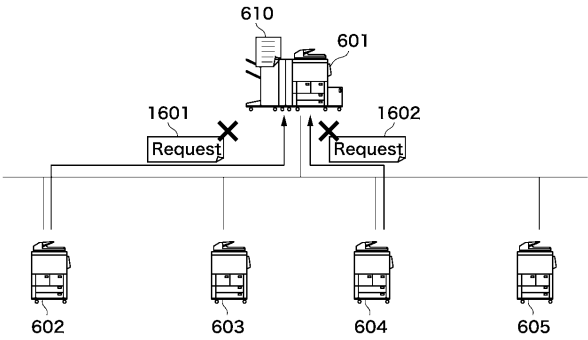
【図 1 3】



【図 1 6】

Device	Date	Time
B	2007.10.5	09:31.22
D	2007.10.5	11:11.02
B	2007.10.5	13:43.32
B	2007.10.5	14:11.24

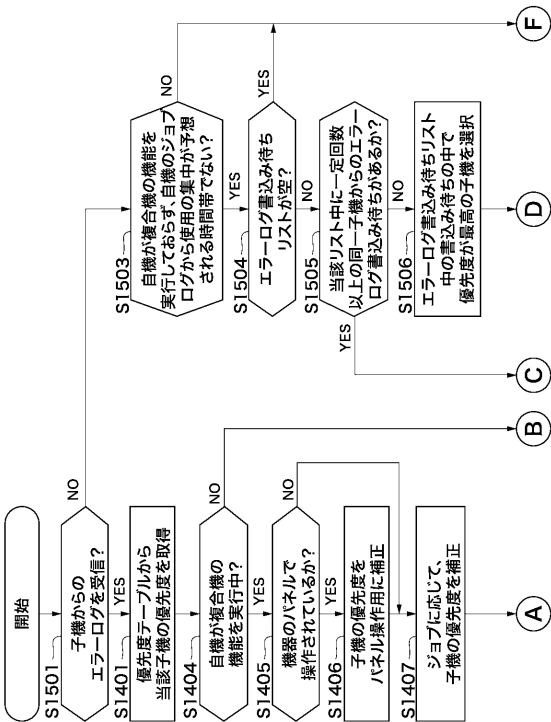
【図 1 4】



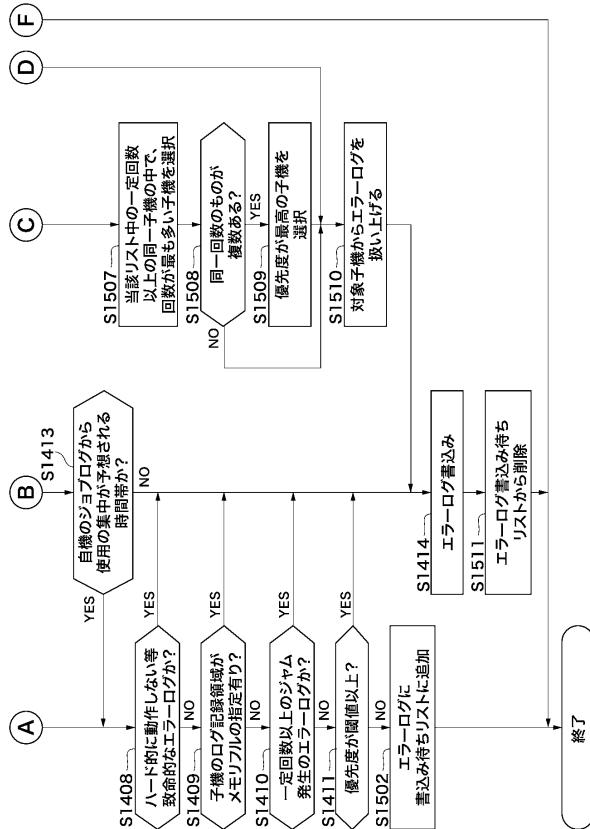
【図 1 5】

	1701	1702	1703
	Device	Date	Time
1704	C	2007.10.5	11:24.55
1705	A	2007.10.5	13:10.00

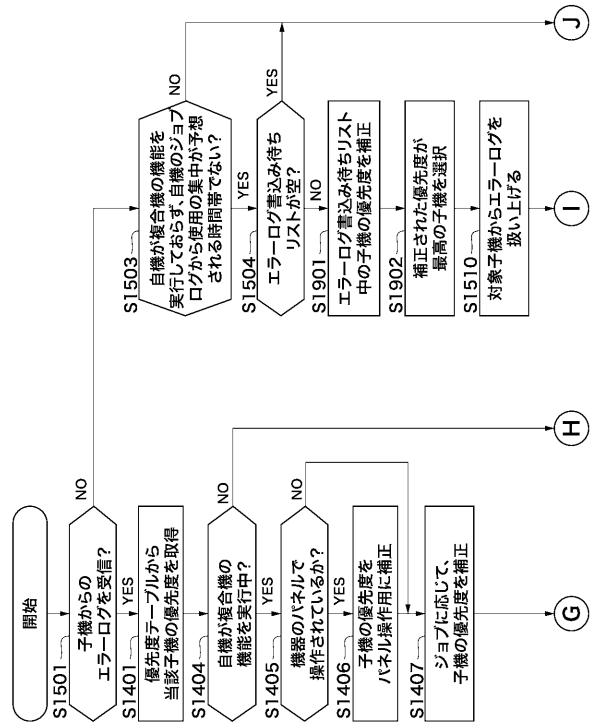
【図 1 7】



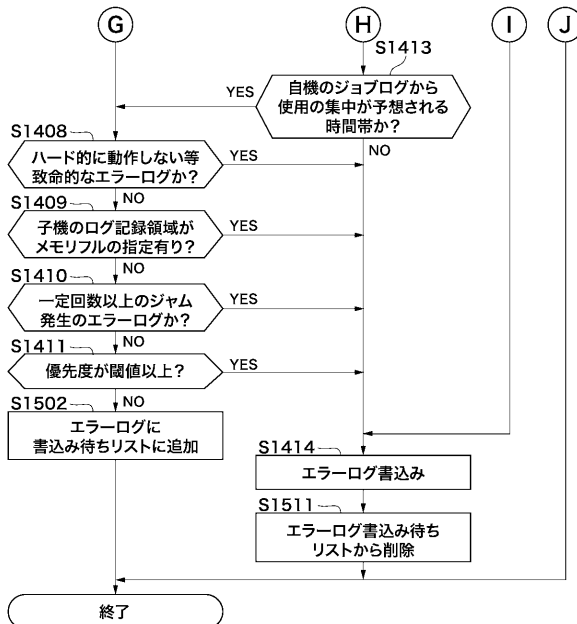
【図 18】



【図 19】



【図 20】



---

フロントページの続き

(56)参考文献 特開平 1 0 - 0 4 0 1 4 0 ( J P , A )  
特開 2 0 0 6 - 3 4 4 0 2 0 ( J P , A )  
特開 2 0 0 8 - 0 7 1 0 8 5 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)  
G 0 6 F 1 1 / 2 8 - 1 1 / 3 4  
G 0 6 F 1 3 / 0 0