

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5867120号
(P5867120)

(45) 発行日 平成28年2月24日(2016.2.24)

(24) 登録日 平成28年1月15日(2016.1.15)

(51) Int. Cl. F 1
G 0 6 F 9 / 4 8 (2 0 0 6 . 0 1) G 0 6 F 9 / 4 6 4 5 2 Z

請求項の数 4 (全 14 頁)

<p>(21) 出願番号 特願2012-19821 (P2012-19821) (22) 出願日 平成24年2月1日(2012.2.1) (65) 公開番号 特開2013-161108 (P2013-161108A) (43) 公開日 平成25年8月19日(2013.8.19) 審査請求日 平成27年1月8日(2015.1.8)</p>	<p>(73) 特許権者 000004237 日本電気株式会社 東京都港区芝五丁目7番1号 (74) 代理人 110002044 特許業務法人プライタス (72) 発明者 中山 康 東京都港区芝五丁目7番1号 日本電気株式会社内 審査官 田中 幸雄 (56) 参考文献 特開平09-218844 (JP, A) 米国特許出願公開第2011/0179252 (US, A1) 最終頁に続く</p>
---	---

(54) 【発明の名称】 情報処理装置、情報処理方法、及びプログラム

(57) 【特許請求の範囲】

【請求項1】

コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、設定されたデータ量となるまで前記メッセージを溜めることによって、前記データ経路を流れる前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、モジュール挿入部を、
備え、

前記データ経路が、前記プロセスからの呼び出しの窓口となる第1のモジュールと、前記プロセスに応じた処理を実行する第2のモジュールと、前記ドライバとを備え、更に、前記第1のモジュール、前記第2のモジュール、及び前記ドライバ、それぞれが、更に、
書込み用のキュー及び読み出し用のキューを備えている場合において、

前記データ量調整モジュールが、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューを備え、且つ、前記対応するキューそれぞれによって、設定されたデータ量となるまで前記メッセージを溜めており、

前記モジュール挿入部は、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューそれぞれの高水位値及び低水位値を設定することによって、前記データ量を設定する、
ことを特徴とする情報処理装置。

【請求項2】

前記モジュール挿入部が、前記データ経路に、複数の前記データ量調整モジュールを挿

10

20

入する、請求項 1 に記載の情報処理装置。

【請求項 3】

コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、設定されたデータ量となるまで前記メッセージを溜めることによって、前記データ経路を流れる前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、ステップを、

有し、

前記データ経路が、前記プロセスからの呼び出しの窓口となる第 1 のモジュールと、前記プロセスに応じた処理を実行する第 2 のモジュールと、前記ドライバとを備え、更に、前記第 1 のモジュール、前記第 2 のモジュール、及び前記ドライバ、それぞれが、更に、

10

書込み用のキュー及び読み出し用のキューを備えている場合において、前記データ量調整モジュールが、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューを備え、且つ、前記対応するキューそれぞれによって、設定されたデータ量となるまで前記メッセージを溜めており、

前記ステップにおいて、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューそれぞれの高水位値及び低水位値を設定することによって、前記データ量を設定する、

ことを特徴とする情報処理方法。

【請求項 4】

コンピュータによって、

20

前記コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、設定されたデータ量となるまで前記メッセージを溜めることによって、前記データ経路を流れる前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、ステップを、

実行させ、

前記データ経路が、前記プロセスからの呼び出しの窓口となる第 1 のモジュールと、前記プロセスに応じた処理を実行する第 2 のモジュールと、前記ドライバとを備え、更に、前記第 1 のモジュール、前記第 2 のモジュール、及び前記ドライバ、それぞれが、更に、書込み用のキュー及び読み出し用のキューを備えている場合において、

前記データ量調整モジュールが、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューを備え、且つ、前記対応するキューそれぞれによって、設定されたデータ量となるまで前記メッセージを溜めており、

30

前記ステップにおいて、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューそれぞれの高水位値及び低水位値を設定することによって、前記データ量を設定する、

プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、STREAMS機構が実装されたコンピュータに用いられる、情報処理装置、情報処理方法、及びプログラムに関する。

40

【背景技術】

【0002】

従来から、オペレーティングシステムであるUNIXでは、カーネルの機能の一部として、STREAMS機構が利用されている。STREAMS機構は、種々のタイプのデバイスドライバと同様に、データ通信におけるネットワークプロトコルを実施するための事実上の業界標準フレームワークになっている。

【0003】

また、STREAMS機構において、「ストリーム」とは、アプリケーションプログラムのよ

50

性のデータ経路であり、全二重パスを意味する。

【0004】

ここで、図8を用いてストリームについて説明する。図8は、ストリームを説明するための概念図である。図8に示すように、ストリーム70は、プログラムモジュールによって構築された、ストリームヘッド71と、処理モジュール72と、ドライバ73とを備えている。

【0005】

また、各モジュールは、一組のキューを備えていることから、ストリーム70は、線形に連結された一組のキューの集まりで構成されているといえる。また、一組のキューは、それぞれ、ライトキュー（「WQ」とも表記する。）と、リードキュー（「RQ」とも表記する。）とで構成されている。

10

【0006】

そして、ユーザ空間におけるプロセス（図8において図示せず）が出力用のメッセージ74をストリームヘッド71に書き込むと、カーネルはそのメッセージを下のライトキューに送る。また、ドライバ73が入力用のメッセージ74を受け取ると、カーネルはそのメッセージ74を上のリードキューに送る。

【0007】

このようなストリームの基本的概念は、設計者であるRitchie, D.M.により記述された非特許文献1に開示されている。また、現在の標準的となっている概念は、上記基本概念のものより改良がなされている。現在のSTREAMS機構の概念は、例えば、非特許文献2に

20

【先行技術文献】

【非特許文献】

【0008】

【非特許文献1】Ritchie, D.M.著、石田晴久監修、長谷部紀元、清水謙多郎訳、「ストリーム入出力システム」、「UNIX原典」、パーソナルメディア、1986、p.365-379

【非特許文献2】Maurice J. Bach著、坂本文・多田好克・村井純 訳、「UNIXカーネルの設計」、共立出版、1990年、p.291-297

【発明の概要】

【発明が解決しようとする課題】

30

【0009】

ところで、このようなSTREAMS機構を実装しているコンピュータでは、ストリーム70において送受信データ量が多くなりすぎると、フロー制御状態に陥ってしまうことがある。上述の図8の例では、ストリーム70を構成している各ライトキュー及び各リードキューそれぞれにメッセージ群が溜められている。そして、各ライトキュー及び各リードキューそれぞれにおいて、データ量はキューの高水位値に達し、「QFULL」状態になってしまっている。

【0010】

そして、このようなフロー制御状態に陥るかどうかは、ドライバ側のデータ処理（ライト側）の処理速度と、アプリケーション側のデータ処理（リード側）の処理速度とに依存している。従って、これらの処理速度によっては、慢性的なフロー制御状態が続いてしまう場合がある。このため、STREAMS機構を実装しているコンピュータでは、この慢性的なフロー制御状態を緩和することが、求められている。

40

【0011】

本発明の目的の一例は、上記問題を解消し、STREAMS機構を実装しているコンピュータにおいて、フロー制御状態の緩和を図り得る、情報処理装置、情報処理方法、及びプログラムを提供することにある。

【課題を解決するための手段】

【0012】

上記目的を達成するため、本発明の一側面における情報処理装置は、コンピュータに構

50

築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、前記データ経路での前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、モジュール挿入部を、備えていることを特徴とする。

【0013】

また、上記目的を達成するため、本発明の一側面における情報処理方法は、コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、前記データ経路での前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、ステップを有することを特徴とする。

【0014】

更に、上記目的を達成するため、本発明の一側面におけるプログラムは、コンピュータによって、前記コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、前記データ経路での前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、ステップを、実行させることを特徴とする。

10

【発明の効果】

【0015】

以上のように、本発明によれば、STREAMS機構を実装しているコンピュータシステムにおいて、フロー制御状態の緩和を図ることができる。

【図面の簡単な説明】

【0016】

【図1】図1は、本発明の実施の形態における情報処理装置の構成を示すブロック図である。

20

【図2】図2は、本実施の形態で挿入されるデータ量調整モジュールの一例を示す図である。

【図3】図3は、発明の実施の形態における情報処理装置の動作を示すフロー図である。

【図4】図4は、データ経路にデータ量調整モジュールが挿入される前後の状態をそれぞれ示す図である。

【図5】図5は、本実施の形態における情報処理装置を実現するプログラムの一例を示す図である。

【図6】図6は、本実施の形態による効果を説明するための図である。

【図7】図7は、本発明の実施の形態における情報処理装置を実現するコンピュータの一例を示すブロック図である。

30

【図8】図8は、ストリームを説明するための概念図である。

【発明を実施するための形態】

【0017】

(実施の形態)

以下、本発明の実施の形態における、情報処理装置、情報処理方法、及びプログラムについて、図1～図8を参照しながら説明する。

【0018】

[装置構成]

最初に、本実施の形態における情報処理装置の構成について説明する。図1は、本発明の実施の形態における情報処理装置の構成を示すブロック図である。

40

【0019】

図1に示すように、本実施の形態における情報処理装置10は、オペレーティングシステム(OS)30が構築されたコンピュータ50において利用される装置である。具体的には、情報処理装置10は、OS30上で動作するプログラムによって構築されている。本実施の形態では、オペレーティングシステムは、UNIX(登録商標)である。

【0020】

図1に示すよう、情報処理装置10は、モジュール挿入部11を備えている。モジュール挿入部11は、OS30に構築されたデータ経路20に対して、そこでのメッセージのデータ量を調整するデータ量調整モジュール12を挿入する。

50

【 0 0 2 1 】

また、データ経路 2 0 は、アプリケーションプログラム 4 0 が実行するプロセス 4 1 とドライバ 2 3 との間でメッセージを送受信するために用いられる経路である。本実施の形態では、データ経路 2 0 は、OS 3 0 に実装されたSTREAMS機構が提供する「ストリーム」である。

【 0 0 2 2 】

また、本実施の形態では、データ量調整モジュール 1 2 は、設定されたデータ量となるまでメッセージを溜める機能を有しており、この機能によって、データ経路 2 0 を流れるメッセージのデータ量を調整する。

【 0 0 2 3 】

このように、情報処理装置 1 0 では、データ経路 2 0 である「ストリーム」において、そこを流れるデータ量が調整されることから、従来と異なり、STREAMS機構を実装しているコンピュータ 5 0 において、フロー制御状態の緩和が図られることになる。

【 0 0 2 4 】

また、データ量を調整するデータ量調整モジュール 1 2 は、データ経路 2 0 の一部となることから、OS 3 0 の機能によってフロー制御状態の緩和が図られている。そして、OS 3 0 の機能によってフロー制御状態の緩和を図ることは、従来においては不可能である。なお、図 1 において、6 0 は、アプリケーションプログラム 4 0 を操作するユーザの端末を示している。

【 0 0 2 5 】

ここで、本実施の形態における情報処理装置 1 0 の構成について更に具体的に説明する。図 1 に示すように、本実施の形態では、データ経路 2 0 は、プログラムモジュールによって構築された、ストリームヘッド 2 1 と、処理モジュール 2 2 と、ドライバ 2 3 とを備えている。

【 0 0 2 6 】

このうち、ストリームヘッド 2 1 は、アプリケーションプログラム 4 0 が実行するプロセス 4 1 からの呼び出しの窓口として機能するモジュールである。処理モジュール 2 2 は、プロセス 4 1 に応じた処理を実行するモジュールである。ドライバ 2 3 は、印刷機、ネットワーク装置といったデバイスを動作させるためのモジュールである。

【 0 0 2 7 】

また、ストリームヘッド 2 1、処理モジュール 2 2、及びドライバ 2 3 は、それぞれ、書込み用のキュー（ライトキュー：WQ）と、読み出し用のキュー（リードキュー：RQ）とを備えている。なお、各キューの動作は、図 8 の例と同様である。

【 0 0 2 8 】

また、図 1 に示すように、モジュール挿入部 1 1 は、データ経路 2 0 を構成しているモジュールとモジュールとの間に、データ量調整モジュール 1 2 を挿入する。そして、データ量調整モジュール 1 2 は、書込み用のキューに対応するキュー（図 1 において「WQ」と表記する。）及び読み出し用のキューに対応するキュー（図 1 において「RQ」と表記する。）を備えている。データ量調整モジュール 1 2 は、これらの対応するキューそれぞれによって、設定されたデータ量となるまでメッセージを溜めることができる。

【 0 0 2 9 】

[データ量調整モジュールの具体例]

ここで、図 2 を用いて、データ量調整モジュール 1 2 の具体例について説明する。図 2 は、本実施の形態で挿入されるデータ量調整モジュールの一例を示す図である。

【 0 0 3 0 】

図 2 の例では、データ量調整モジュール 1 2 のドライバエントリルーチンのみが示されている。具体的には、write側putプロシジャである「xxx_wput()」のみが示されている。ここで、接頭辞の「xxx_」は、そのデータ経路 2 0 を構築するモジュールのモジュール名であり、その文字列はそのモジュール名にふさわしいものであれば良く、特に限定されるものではない。なお、上記ドライバエントリルーチンの形式は、「UNIX System V R

10

20

30

40

50

「release4」に準拠しており、以降においても同様である。

【0031】

また、図2に示すように、「xxx_wput()」は、処理部分として、高水位値を取得する「GET_HIWATコマンド」と、低水位値を取得する「GET_LOWATコマンド」と、高水位値を取得した値に設定する「SET_HIWATコマンド」と、低水位値を取得した値に設定する「SET_LOWATコマンド」とを有している。これらのコマンドは、システムコールioctl(I_STR)から実行される。

【0032】

また、「xxx_wput()」は、メッセージタイプM_FLUSHといった特殊なメッセージを除き、メッセージを自身のライトキューにputq()する。このため、図2の例では、データ量調整モジュールは、データ量が高水位値に達するまで、ライトキューにメッセージを溜める処理を実行する。

10

【0033】

続いて、write側サービスプロシジャである「xxx_wsrv()」、read側putプロシジャである「xxx_rput()」、及びread側サービスプロシジャの「xxx_rsrv()」について説明する。なお、これらの図2における図示は省略する。

【0034】

まず、write側サービスプロシジャである「xxx_wsrv()」については、メッセージに対して特に処理(加工)を行わずに、geq()を実行した後にputnext()を実行するだけの簡素な実装とすれば良い。

20

【0035】

また、read側putプロシジャである「xxx_rput()」については、メッセージタイプM_FLUSHといった特殊なメッセージを除き、メッセージを自身のリードキューにputq()する実装とすれば良い。これにより、データ量調整モジュールは、データ量が高水位値に達するまで、リードキューにメッセージを溜める処理を実行する。

【0036】

そして、read側サービスプロシジャである「xxx_rsrv()」については、write側サービスプロシジャと同様に、メッセージに対して特に処理(加工)を行わずに、getq()を実行した後にputnext()を実行するだけの簡素な実装とすれば良い。

【0037】

なお、データ量調整モジュールのwrite側及びread側のハイウォーター値及びローウォーター値は、デフォルトの状態、データ経路20を構成する他のモジュール及びドライバのハイウォーター値及びローウォーター値に比べて十分大きくなるように、設定しておくのが好ましい。具体的な値は、ドライバ側のデータ処理の処理速度と、アプリケーション側のデータ処理の処理速度とに基づいて適宜設定される。

30

【0038】

そして、このように、データ量調整モジュール12において、それを構成する各キューの高水位値及び低水位値の設定が可能であるため、本実施の形態では、モジュール挿入部11は、データ量調整モジュール12に溜めておくことができるメッセージのデータ量を変更することができる。

40

【0039】

[装置動作]

次に、本発明の実施の形態における情報処理装置の動作について図3及び図4を用いて説明する。図3は、発明の実施の形態における情報処理装置の動作を示すフロー図である。図4は、データ経路にデータ量調整モジュールが挿入される前後の状態をそれぞれ示す図である。

【0040】

また、以下の説明においては、適宜図1及び図2を参照する。更に、本実施の形態では、情報処理装置を動作させることによって、情報処理方法が実施される。よって、本実施の形態における情報処理方法の説明は、以下の情報処理装置10の動作説明に代える。

50

【 0 0 4 1 】

図 3 に示すように、まず、モジュール挿入部 1 1 は、プロセス 4 1 からデータ量調整モジュールの挿入の指示を受け付け（ステップ A 1）、データ経路 2 0 に、データ量調整モジュール 1 2 を挿入する（ステップ A 2）。

【 0 0 4 2 】

具体的には、本実施の形態では、プロセス 4 1 によってシステムコール `ioctl(I_PUSH)` が行なわれると、モジュール挿入部 1 1 は、データ経路 2 0 に、データ量調整モジュールを挿入する。

【 0 0 4 3 】

次に、モジュール挿入部 1 1 は、予め設定されている高水位値を取得し、取得した高水位値を、データ量調整モジュール 1 2 を構成するキューに設定する（ステップ A 3）。続いて、モジュール挿入部 1 1 は、予め設定されている低水位値を取得し、取得した低水位値を、データ量調整モジュール 1 2 を構成するキューに設定する（ステップ A 4）。

【 0 0 4 4 】

ステップ A 1 ~ A 4 が実行されると、図 4 に示すように、データ経路（ストリーム）2 0 において、データ量調整モジュール 1 2 が挿入される（図 4 の右側参照）。図 4 の例では、図 1 の例と同様に、ストリームヘッド 2 1 の直下に、データ量調整モジュール 1 2 が挿入されている。ステップ A 4 の実行後、モジュール挿入部 1 1 における処理は終了する。

【 0 0 4 5 】

なお、図 3 の例では、データ量調整モジュール 1 2 の挿入のきっかけとして、プロセス 4 1 によるシステムコール `ioctl(I_PUSH)` が用いられているが、本実施の形態は、これに限定される趣旨ではない。

【 0 0 4 6 】

例えば、データ量調整モジュール 1 2 の挿入は、データ量調整モジュールの挿入が必要なドライバ 2 3 として事前に指定されたドライバが、プロセス 4 1 による `autopush` コマンドの実行によってオープンされた場合に、それをきっかけに行なわれても良い。

【 0 0 4 7 】

また、データ量調整モジュール 1 2 の挿入は、プロセス 4 1 が `strchg` コマンドを用いて、「`strchg -h [データ量調整モジュール名] < [ドライバのデバイスファイル名]`」を実行した場合に、それをきっかけにして、特定のオープン中のストリームに対して行なわれても良い。

【 0 0 4 8 】

但し、データ量調整モジュール 1 2 の高水位値、低水位値を取得及び設定できるのは、プロセス 4 1 がシステムコール `ioctl(I_STR)` を発行する場合のみである。

【 0 0 4 9 】

〔プログラム〕

また、本実施の形態におけるプログラムは、コンピュータに、図 3 に示すステップ A 1 ~ A 4 を実行させるプログラムであれば良い。このプログラムをコンピュータ 5 0 にインストールし、実行することによって、本実施の形態における情報処理装置 1 0 と情報処理方法とを実現することができる。この場合、コンピュータの CPU（Central Processing Unit）は、モジュール挿入部 1 1 として機能し、処理を行なう。更に、本実施の形態では、情報処理装置 1 0 を実現するプログラムは、アプリケーションプログラム 4 0 の一部であっても良い。

【 0 0 5 0 】

ここで、図 5 を用いて、本実施の形態におけるプログラムの一例について説明する。図 5 は、本実施の形態における情報処理装置を実現するプログラムの一例を示す図である。図 5 の例では、プログラムは、アプリケーションプログラム 4 0 において、そのモジュールとして実装されている。

【 0 0 5 1 】

10

20

30

40

50

図5の例では、プログラムによって実行される処理の一例として、ストリームドライバのオープン、データ量調整モジュール12のプッシュ処理、データ量調整モジュール12の高水位値の取得処理、データ量調整モジュール12の高水位値の設定処理が示されている。

【0052】

[実施の形態による効果]

以上のように、本実施の形態によれば、フロー制御状態に陥りやすいストリームに対してデータ量調整モジュール12を挿入しておくことができるので、OS側の機能によってフロー制御状態の緩和が図られることになる。この点について図6を用いて説明する。図6は、本実施の形態による効果を説明するための図である。

10

【0053】

図6に示すように、データ経路(ストリーム)20にデータ量調整モジュール12が挿入されると、メッセージ24の一部はデータ量調整モジュールに一旦溜められた状態となる。従って、送受信されるデータ量が、図8に示した例と同様であっても、アプリケーション側(図6ではライト側)及びデバイス側(図6ではリード側)の各キューにおいて、QFULL状態が緩和される。

【0054】

また、本実施の形態では、データ量調整モジュール12の高水位値及び低水位値を変更することができるので、フロー制御状態が起こりやすいストリームに対して、確実にフロー制御状態を緩和することが可能となる。加えて、本実施の形態では、データ量調整モジュールは、ストリーム毎に挿入できるため、ストリームドライバ単位で、データ量調整モジュールを利用するかどうかを選択することができる。

20

【0055】

[変形例]

また、図1～図6の例では、挿入されているデータ量調整モジュールは、1つであるが、本実施の形態では、データ量調整モジュール12の数は限定されるものではない。例えば、フロー制御の発生状態が慢性的なストリームに対しては、複数のデータ量調整モジュール12が挿入されていても良い。

【0056】

但し、上記の場合、「ioctl(I_STR)」による高水位値及び低水位値についての取得及び設定は、最もストリームヘッド側にあるフロー制御緩和モジュールのみに限られてしまう。

30

【0057】

なお、データ量調整モジュール12は、ストリームモジュールとしてのデータ処理(メッセージの中身のデータの加工)を行わない作りになっている。従って、カーネルパラメータNSTRPUSHにて設定している値(個数)に達するまで、同一ストリームに、データ量調整モジュールを複数挿入する事が可能となっている。

【0058】

[コンピュータ]

ここで、実施の形態におけるプログラムを実行することによって、情報処理装置10を実現するコンピュータ50について図7を用いて説明する。図7は、本発明の実施の形態における情報処理装置を実現するコンピュータの一例を示すブロック図である。

40

【0059】

図7に示すように、コンピュータ110は、CPU111と、メインメモリ112と、記憶装置113と、入力インターフェイス114と、表示コントローラ115と、データリーダ/ライタ116と、通信インターフェイス117とを備える。これらの各部は、バス121を介して、互いにデータ通信可能に接続される。

【0060】

CPU111は、記憶装置113に格納された、本実施の形態におけるプログラム(コード)をメインメモリ112に展開し、これらを所定順序で実行することにより、各種の

50

演算を実施する。メインメモリ 112 は、典型的には、D R A M (Dynamic Random Access Memory) 等の揮発性の記憶装置である。また、本実施の形態におけるプログラムは、コンピュータ読み取り可能な記録媒体 120 に格納された状態で提供される。なお、本実施の形態におけるプログラムは、通信インターフェイス 117 を介して接続されたインターネット上で流通するものであっても良い。

【0061】

また、記憶装置 113 の具体例としては、ハードディスクの他、フラッシュメモリ等の半導体記憶装置が挙げられる。入力インターフェイス 114 は、C P U 111 と、キーボード及びマウスといった入力機器 118 との間のデータ伝送を仲介する。表示コントローラ 115 は、ディスプレイ装置 119 と接続され、ディスプレイ装置 119 での表示を制御する。データリーダ/ライタ 116 は、C P U 111 と記録媒体 120 との間のデータ伝送を仲介し、記録媒体 120 からのプログラムの読み出し、及びコンピュータ 110 における処理結果の記録媒体 120 への書き込みを実行する。通信インターフェイス 117 は、C P U 111 と、他のコンピュータとの間のデータ伝送を仲介する。

10

【0062】

また、記録媒体 120 の具体例としては、C F (Compact Flash (登録商標)) 及び S D (Secure Digital) 等の汎用的な半導体記憶デバイス、フレキシブルディスク (Flexible Disk) 等の磁気記憶媒体、又は C D - R O M (Compact Disk Read Only Memory) などの光学記憶媒体が挙げられる。

【0063】

上述した実施の形態の一部又は全部は、以下に記載する(付記1)~(付記15)によって表現することができるが、以下の記載に限定されるものではない。

20

【0064】

(付記1)

コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、前記データ経路での前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、モジュール挿入部を、備えていることを特徴とする情報処理装置。

【0065】

(付記2)

前記データ量調整モジュールが、設定されたデータ量となるまで前記メッセージを溜めることによって、前記データ経路を流れる前記メッセージのデータ量を調整する、付記1に記載の情報処理装置。

30

【0066】

(付記3)

前記データ経路が、前記プロセスからの呼び出しの窓口となる第1のモジュールと、前記プロセスに応じた処理を実行する第2のモジュールと、前記ドライバとを備え、更に、前記第1のモジュール、前記第2のモジュール、及び前記ドライバ、それぞれが、更に、書き込み用のキュー及び読み出し用のキューを備えている場合において、

前記データ量調整モジュールが、前記書き込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューを備え、且つ、前記対応するキューそれぞれによって、設定されたデータ量となるまで前記メッセージを溜める、付記2に記載の情報処理装置。

40

【0067】

(付記4)

前記モジュール挿入部が、前記書き込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューそれぞれの高水位値及び低水位値を設定することによって、前記データ量を設定する、付記2または3に記載の情報処理装置。

【0068】

(付記5)

50

前記モジュール挿入部が、前記データ経路に、複数の前記データ量調整モジュールを挿入する、付記 1 ~ 4 のいずれかに記載の情報処理装置。

【 0 0 6 9 】

(付記 6)

コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、前記データ経路での前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、ステップを、

有することを特徴とする情報処理方法。

【 0 0 7 0 】

(付記 7)

前記データ量調整モジュールが、設定されたデータ量となるまで前記メッセージを溜めることによって、前記データ経路を流れる前記メッセージのデータ量を調整する、付記 6 に記載の情報処理方法。

【 0 0 7 1 】

(付記 8)

前記データ経路が、前記プロセスからの呼び出しの窓口となる第 1 のモジュールと、前記プロセスに応じた処理を実行する第 2 のモジュールと、前記ドライバとを備え、更に、前記第 1 のモジュール、前記第 2 のモジュール、及び前記ドライバ、それぞれが、更に、書込み用のキュー及び読み出し用のキューを備えている場合において、

前記データ量調整モジュールが、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューを備え、且つ、前記対応するキューそれぞれによって、設定されたデータ量となるまで前記メッセージを溜める、付記 7 に記載の情報処理方法。

【 0 0 7 2 】

(付記 9)

前記モジュール挿入部が、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューそれぞれの高水位値及び低水位値を設定することによって、前記データ量を設定する、付記 7 または 8 に記載の情報処理方法。

【 0 0 7 3 】

(付記 10)

前記ステップにおいて、前記データ経路に、複数の前記データ量調整モジュールを挿入する、付記 6 ~ 9 のいずれかに記載の情報処理方法。

【 0 0 7 4 】

(付記 11)

コンピュータによって、

前記コンピュータに構築され、且つ、プロセスとドライバとの間でメッセージを送受信する、データ経路に対して、前記データ経路での前記メッセージのデータ量を調整するデータ量調整モジュールを挿入する、ステップを、

実行させるプログラム。

【 0 0 7 5 】

(付記 12)

前記データ量調整モジュールが、設定されたデータ量となるまで前記メッセージを溜めることによって、前記データ経路を流れる前記メッセージのデータ量を調整する、付記 11 に記載のプログラム。

【 0 0 7 6 】

(付記 13)

前記データ経路が、前記プロセスからの呼び出しの窓口となる第 1 のモジュールと、前記プロセスに応じた処理を実行する第 2 のモジュールと、前記ドライバとを備え、更に、前記第 1 のモジュール、前記第 2 のモジュール、及び前記ドライバ、それぞれが、更に、書込み用のキュー及び読み出し用のキューを備えている場合において、

10

20

30

40

50

前記データ量調整モジュールが、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューを備え、且つ、前記対応するキューそれぞれによって、設定されたデータ量となるまで前記メッセージを溜める、
付記 1 2 に記載のプログラム。

【 0 0 7 7 】

(付記 1 4)

前記モジュール挿入部が、前記書込み用のキューに対応するキュー及び前記読み出し用のキューに対応するキューそれぞれの高水位値及び低水位値を設定することによって、前記データ量を設定する、付記 1 2 または 1 3 に記載のプログラム。

10

【 0 0 7 8 】

(付記 1 5)

前記ステップにおいて、前記データ経路に、複数の前記データ量調整モジュールを挿入する、付記 1 1 ~ 1 4 のいずれかに記載のプログラム。

【 産業上の利用可能性 】

【 0 0 7 9 】

以上のように、本発明によれば、STREAMS機構を実装しているコンピュータにおいて、フロー制御状態を緩和できる。本発明は、特に、OSの機能としてストリームを提供するUNIXが導入されているコンピュータに有用である。

【 符号の説明 】

20

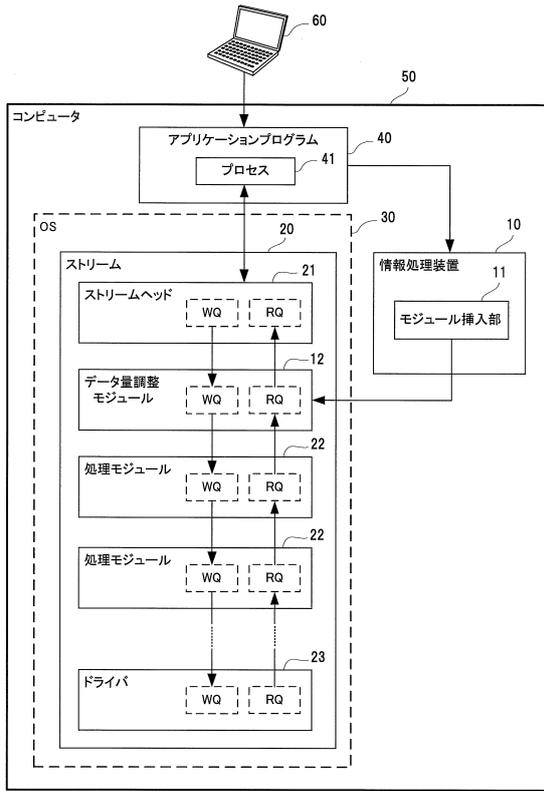
【 0 0 8 0 】

- 1 0 情報処理装置
- 1 1 モジュール挿入部
- 2 0 データ経路 (ストリーム)
- 2 1 ストリームヘッド
- 2 2 処理モジュール
- 2 3 ドライバ
- 2 4 メッセージ
- 3 0 オペレーティングシステム (OS)
- 4 0 アプリケーションプログラム
- 4 1 プロセス
- 5 0 コンピュータ
- 6 0 端末
- 1 1 0 コンピュータ
- 1 1 1 CPU
- 1 1 2 メインメモリ
- 1 1 3 記憶装置
- 1 1 4 入力インターフェイス
- 1 1 5 表示コントローラ
- 1 1 6 データリーダー/ライター
- 1 1 7 通信インターフェイス
- 1 1 8 入力機器
- 1 1 9 ディスプレイ装置
- 1 2 0 記録媒体
- 1 2 1 バス

30

40

【図1】



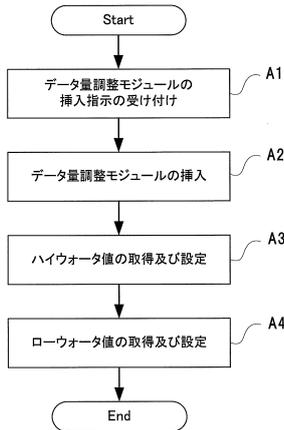
【図2】

```

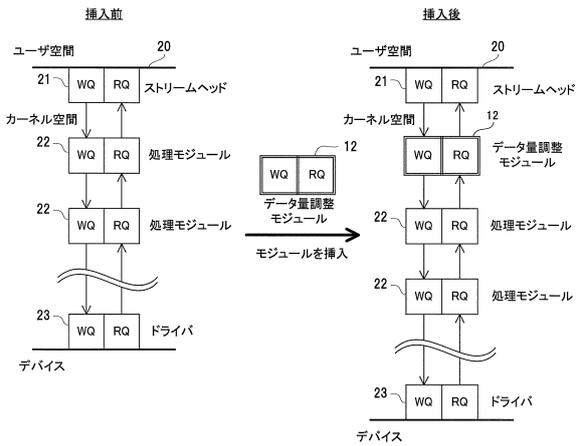
12
xxx_wput(q,mp) / write側put プロシジャの一部 /
{
  struct iocblk ioc;
  :
  swich(mp->b_datap->db_type{
    case M_IOCTL:
      ioc=(struct iocblk *) (mp->b_rptr);
      swich(ioc->ioc_cmd){
        case GET_HIWAT:
          ・ このキューのハイウォーター値を返す処理
          break;
        case GET_LOWAT:
          ・ このキューのローウォーター値を返す処理
          break;
        case SET_HIWAT:
          ・ 引数で渡されたハイウォーター値をライトキュー、リードキューにセットする処理
          break;
        case SET_LOWAT:
          ・ 引数で渡されたローウォーター値をライトキュー、リードキューにセットする処理
          break;
      }
    default:
      putq(q, mp);
      break;
  }
}

```

【図3】



【図4】



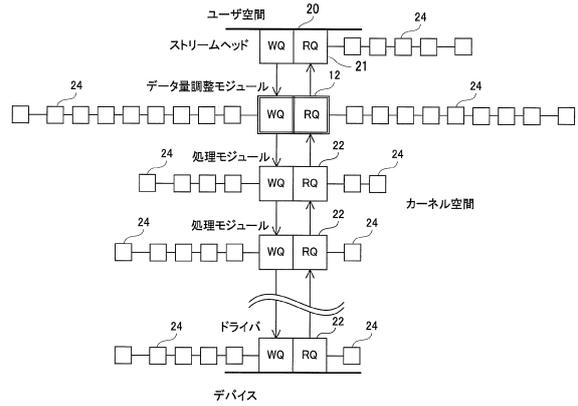
【図5】

```

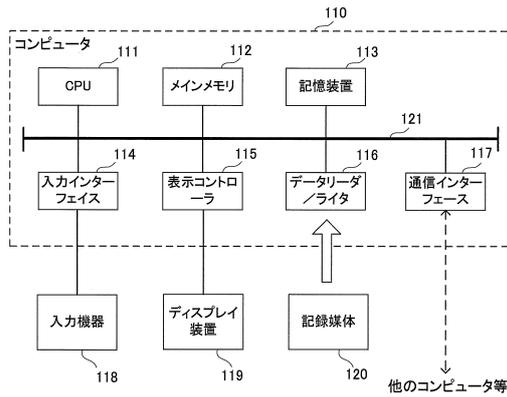
情報処理装置を実現するプログラムの一例
11
main()
{
    :
    struct striocctl ic;
    int hiwat, lowat;
    :
    fd=open("/dev/YYYY", O_RDWR); /* ストリームドライバのオープン */
    :
    i=ioctl(fd, I_PUSH, "xxx"); /* データ量調整モジュールのPUSH */
    :
    /* ハイウォーター値を取得する処理（ローウォーター値の場合も同様） */
    ic.ic_cmd=GET_HIWAT;
    ic.ic_limout=0;
    ic.ic_len=0;
    ic.ic_dp=0;
    i=ioctl(fd, I_STR, &ic);
    hiwat = ic.ic_dp;
    :
    /* ハイウォーター値を設定する処理（ローウォーター値の場合も同様） */
    ic.ic_cmd=SET_HIWAT;
    ic.ic_limout=0;
    ic.ic_len=4;
    ic.ic_dp=5000;
    i=ioctl(fd, I_STR, &ic);
    :
}

```

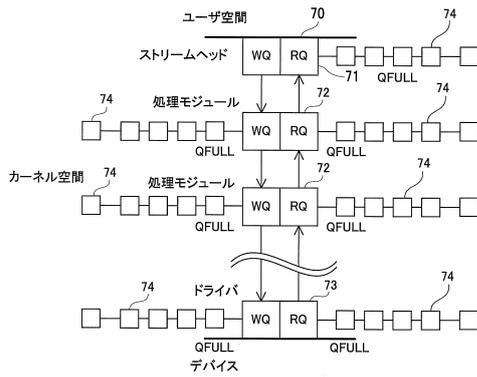
【図6】



【図7】



【図8】



フロントページの続き

(58)調査した分野(Int.Cl. , DB名)

G 0 6 F 9 / 4 8