

【公報種別】特許法第 17 条の 2 の規定による補正の掲載

【部門区分】第 6 部門第 3 区分

【発行日】令和 1 年 11 月 7 日 (2019.11.7)

【公開番号】特開 2019-71120 (P2019-71120A)

【公開日】令和 1 年 5 月 9 日 (2019.5.9)

【年通号数】公開・登録公報 2019-017

【出願番号】特願 2019-4419 (P2019-4419)

【国際特許分類】

G 0 6 F 9/48 (2006.01)

【F I】

G 0 6 F 9/48 3 7 0

【手続補正書】

【提出日】令和 1 年 9 月 25 日 (2019.9.25)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

コンピューティングシステムによって実行されるタスクを制御するための方法であって

、  
グラフに基づくプログラムの仕様に関するデータ構造をデータストレージシステムに記憶するステップであって、前記グラフに基づくプログラムの仕様が、タスクを表す複数のノードを含み、前記グラフに基づくプログラムの仕様が、前記ノードによって表される複数のタスクの間の少なくとも部分的な順序を規定するノードの間の有向辺を含む、前記記憶するステップと、

前記グラフに基づくプログラムの仕様によって規定されたプログラムを少なくとも 1 つのプロセッサを使用して実行するステップであって、

第 1 のタスクに対応する第 1 のサブルーチンを第 1 のプロセスによって実行することであって、前記第 1 のサブルーチンが、前記第 1 のタスクを実行するための第 1 のタスクセクション、及び前記グラフに基づくプログラムの仕様の中で前記第 1 のタスクのために 1 又は 2 以上の対応する後続のタスクの実行を制御するための論理を含む前記第 1 のサブルーチンに含まれる第 1 の制御セクションを含み、前記第 1 の制御セクションが、第 2 のタスクに対応する第 2 のサブルーチンを呼び出すための第 1 の関数呼び出しを含み、前記第 2 のタスクが、前記第 2 のタスクの実行を続けるために前記第 1 のプロセスに第 2 のプロセスをスポーニングさせる遅延閾値タイマーが超えられるまで前記第 1 のプロセスを使用して実行を開始するように構成され、前記第 1 のサブルーチンが、前記遅延閾値タイマーが超えられるときに前記第 2 のタスク及び第 3 のタスクが同時に実行されるように、前記第 1 のプロセスが前記第 2 のプロセスをスポーニングすると前記第 1 のプロセス上で実行される、前記第 3 のタスクに対応する第 3 のサブルーチンを呼び出すための第 2 の関数呼び出しをさらに含む、前記実行すること、

前記第 1 のタスクセクションが前記第 1 のタスクを実行するのを待っている保留状態及び前記第 1 のタスクセクションが前記第 1 のタスクを実行することを防止された抑制状態を含む 1 組の可能な状態から選択された前記第 1 のタスクの状態を示す状態情報を記憶することを含む、前記実行するステップと、

前記第 2 のタスクに対応する前記第 2 のサブルーチンを実行するステップであって、前記第 2 のサブルーチンが、前記第 2 のタスクを実行するための第 2 のタスクセクション、

及び記憶された状態情報によって示される前記第 1 のタスクの前記状態に少なくとも部分的に基づいて前記第 2 のタスクセクションの実行を制御する第 2 の制御セクションを含む、前記実行するステップとを含む、前記方法。

**【請求項 2】**

上流のノードから下流のノードへの制御有向辺が、部分的な順序で前記上流のノードによって表されるタスクが前記下流のノードによって表される別のタスクに先立つことを示す、請求項 1 に記載の方法。

**【請求項 3】**

1 組の可能な状態が、第 1 のタスクセクションが第 1 のタスクを実行し終えた完了状態又は前記第 1 のタスクセクションが前記第 1 のタスクを実行する過程にあるアクティブ状態のうちの 1 又は 2 以上をさらに含む、請求項 1 に記載の方法。

**【請求項 4】**

第 2 のサブルーチンの第 2 の制御セクションの論理が、第 2 のタスクセクションが呼び出されるか否かをフラグの値に基づいて判定する、請求項 1 に記載の方法。

**【請求項 5】**

第 2 のサブルーチンの第 2 の制御セクションの論理が、第 2 のタスクを表すノードに制御有向辺によって接続されるノードによって表されるすべてのタスクが抑制状態にある場合、第 2 のタスクセクションが呼び出されないと判定する、請求項 1 に記載の方法。

**【請求項 6】**

第 2 のサブルーチンの第 2 の制御セクションの論理が、第 2 のタスクを表すノードに制御有向辺によって接続されるノードによって表される少なくとも 1 つのタスクが完了状態にあり、前記第 2 のタスクを表すノードに制御有向辺によって接続されるノードによって表されるタスクのいずれも保留状態にない場合、第 2 のタスクセクションが呼び出されると判定する、請求項 1 に記載の方法。

**【請求項 7】**

第 2 のサブルーチンの第 2 の制御セクションが、抑制状態を 1 又は 2 以上のその他のタスクに伝播させるために、記憶される状態情報を判定する論理を含む、請求項 1 に記載の方法。

**【請求項 8】**

1 群の複数のタスクの各々の状態情報が、前記 1 群の複数のタスクの前記各々のそれぞれ 1 つによって記憶され、第 2 のサブルーチンに含まれる第 2 の制御セクションによってアクセスされる、請求項 1 に記載の方法。

**【請求項 9】**

第 2 のタスクが、第 2 のタスクセクションが呼び出されるかどうかを前記第 2 のタスクに関連するアクティベーションカウンタに基づいて判定し、前記アクティベーションカウンタが、複数のタスクの状態の変化に応じて調整されるカウンタ値を含み、前記第 2 のタスクを実行することが、調整されたアクティベーションカウンタが所定の値に一致することに応じて前記第 2 のタスクを実行することを含む、請求項 1 に記載の方法。

**【請求項 10】**

第 2 のタスクの実行を続けるように構成されたスポーニングされた第 2 のプロセスが、前記第 2 のタスクに関連するスタックフレームを取得し、前記スタックフレームに保有されたリターンポイントを取り除くように構成される、請求項 1 に記載の方法。

**【請求項 11】**

タスクを制御するためのコンピューティングシステムであって、  
グラフに基づくプログラムの仕様に関するデータ構造を記憶するための手段であって、前記グラフに基づくプログラムの仕様が、タスクを表す複数のノードを含み、前記グラフに基づくプログラムの仕様が、前記ノードによって表される複数のタスクの間の少なくとも部分的な順序を規定するノードの間の有向辺を含む、前記記憶するための手段と、前記グラフに基づくプログラムの仕様によって規定されたプログラムを実行するための手段とを含み、実行することが、

第 1 のタスクに対応する第 1 のサブルーチンを第 1 のプロセスによって実行することであって、前記第 1 のサブルーチンが、前記第 1 のタスクを実行するための第 1 のタスクセクション、及び前記グラフに基づくプログラムの仕様の中で前記第 1 のタスクのために 1 又は 2 以上の対応する後続のタスクの実行を制御するための論理を含む前記第 1 のサブルーチンに含まれる第 1 の制御セクションを含み、前記第 1 の制御セクションが、第 2 のタスクに対応する第 2 のサブルーチンを呼び出すための第 1 の関数呼び出しを含み、前記第 2 のタスクが、前記第 2 のタスクの実行を続けるために前記第 1 のプロセスに第 2 のプロセスをスポーニングさせる遅延閾値タイマーが超えられるまで前記第 1 のプロセスを使用して実行を開始するように構成され、前記第 1 のサブルーチンが、前記遅延閾値タイマーが超えられるときに前記第 2 のタスク及び第 3 のタスクが同時に実行されるように、前記第 1 のプロセスが前記第 2 のプロセスをスポーニングすると前記第 1 のプロセス上で実行される、前記第 3 のタスクに対応する第 3 のサブルーチンを呼び出すための第 2 の関数呼び出しをさらに含む、前記実行すること、

前記第 1 のタスクセクションが前記第 1 のタスクを実行するのを待っている保留状態及び前記第 1 のタスクセクションが前記第 1 のタスクを実行することを防止された抑制状態を含む 1 組の可能な状態から選択された前記第 1 のタスクの状態を示す状態情報を記憶すること、並びに

前記第 2 のタスクに対応する前記第 2 のサブルーチンを実行することであって、前記第 2 のサブルーチンが、前記第 2 のタスクを実行するための第 2 のタスクセクション、及び記憶された状態情報によって示される前記第 1 のタスクの前記状態に少なくとも部分的に基づいて前記第 2 のタスクセクションの実行を制御する第 2 の制御セクションを含む、前記実行することを含む、前記コンピューティングシステム。

**【請求項 1 2】**

上流のノードから下流のノードへの制御有向辺が、部分的な順序で前記上流のノードによって表されるタスクが前記下流のノードによって表される別のタスクに先立つことを示す、請求項 1 1 に記載のコンピューティングシステム。

**【請求項 1 3】**

1 組の可能な状態が、第 1 のタスクセクションが第 1 のタスクを実行し終えた完了状態又は前記第 1 のタスクセクションが前記第 1 のタスクを実行する過程にあるアクティブ状態のうちの 1 又は 2 以上をさらに含む、請求項 1 1 に記載のコンピューティングシステム

。

**【請求項 1 4】**

第 2 のサブルーチンの第 2 の制御セクションの論理が、第 2 のタスクセクションが呼び出されるか否かをフラグの値に基づいて判定する、請求項 1 1 に記載のコンピューティングシステム。

**【請求項 1 5】**

第 2 のサブルーチンの第 2 の制御セクションの論理が、第 2 のタスクを表すノードに制御有向辺によって接続されるノードによって表されるすべてのタスクが抑制状態にある場合、第 2 のタスクセクションが呼び出されないと判定する、請求項 1 1 に記載のコンピューティングシステム。

**【請求項 1 6】**

第 2 のサブルーチンの第 2 の制御セクションの論理が、第 2 のタスクを表すノードに制御有向辺によって接続されるノードによって表される少なくとも 1 つのタスクが完了状態にあり、前記第 2 のタスクを表すノードに制御有向辺によって接続されるノードによって表されるタスクのいずれも保留状態にない場合、第 2 のタスクセクションが呼び出されると判定する、請求項 1 1 に記載のコンピューティングシステム。

**【請求項 1 7】**

第 2 のサブルーチンの第 2 の制御セクションが、抑制状態を 1 又は 2 以上のその他のタスクに伝播させるために、記憶される状態情報を判定する論理を含む、請求項 1 1 に記載のコンピューティングシステム。

**【請求項 18】**

1 群の複数のタスクの各々の状態情報が、前記 1 群の複数のタスクの前記各々のそれぞれ 1 つによって記憶され、第 2 のサブルーチンに含まれる第 2 の制御セクションによってアクセスされる、請求項 11 に記載のコンピューティングシステム。

**【請求項 19】**

第 2 のタスクが、第 2 のタスクセクションが呼び出されるかどうかを前記第 2 のタスクに関連するアクティベーションカウンタに基づいて判定し、前記アクティベーションカウンタが、複数のタスクの状態の変化に応じて調整されるカウンタ値を含み、前記第 2 のタスクを実行することが、調整されたアクティベーションカウンタが所定の値に一致することに応じて前記第 2 のタスクを実行することを含む、請求項 11 に記載のコンピューティングシステム。

**【請求項 20】**

第 2 のタスクの実行を続けるように構成されたスポーニングされた第 2 のプロセスが、前記第 2 のタスクに関連するスタックフレームを取得し、前記スタックフレームに保有されたリターンポイントを取り除くように構成される、請求項 11 に記載のコンピューティングシステム。

**【請求項 21】**

タスクを制御するための、コンピュータ可読ストレージ媒体に記憶されたコンピュータプログラムであって、コンピューティングシステムに請求項 1 ~ 10 のいずれかに記載の方法を実行させるための命令を含む、前記コンピュータプログラム。

**【請求項 22】**

タスクを制御するためのコンピューティングシステムであって、  
グラフに基づくプログラムの仕様に関するデータ構造を記憶するデータストレージシステムと、

請求項 1 ~ 10 のいずれかに記載の方法を実行するように構成された少なくとも 1 つのプロセッサとを含む、前記コンピューティングシステム。