

(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2009-86733

(P2009-86733A)

(43) 公開日 平成21年4月23日(2009.4.23)

(51) Int.Cl.		F I		テーマコード (参考)
G06F 9/50	(2006.01)	G06F 9/46	462A	
G06F 9/48	(2006.01)	G06F 9/46	452F	

審査請求 未請求 請求項の数 10 O L (全 20 頁)

(21) 出願番号	特願2007-252025 (P2007-252025)	(71) 出願人	000003078 株式会社東芝 東京都港区芝浦一丁目1番1号
(22) 出願日	平成19年9月27日 (2007.9.27)	(74) 代理人	100109900 弁理士 堀口 浩
		(72) 発明者	吉田 英樹 東京都港区芝浦一丁目1番1号 株式会社東芝内
		(72) 発明者	崎山 伸夫 東京都港区芝浦一丁目1番1号 株式会社東芝内
		(72) 発明者	木村 哲郎 東京都港区芝浦一丁目1番1号 株式会社東芝内

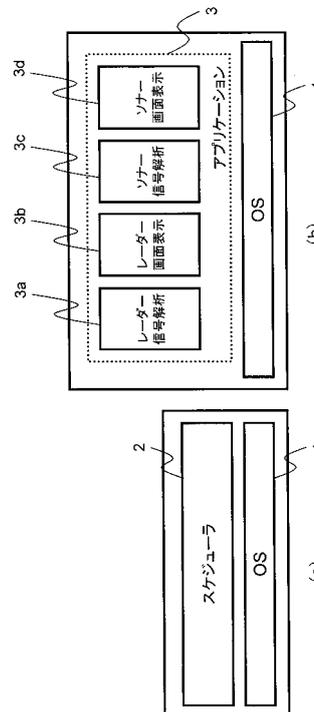
(54) 【発明の名称】 情報処理装置、情報処理装置の制御方法および情報処理装置の制御プログラム

(57) 【要約】

【課題】 周期ごとに発生するアプリケーションの実行処理の打ち切りを抑制できる。

【解決手段】 本発明のプログラムは、一定周期ごとに複数のアプリケーション 3 a ~ 3 d に対して CPU 時間を割り当てるプログラム 2 であって、情報処理装置に、次の周期に処理されるアプリケーション 3 a ~ 3 d ごとに CPU 時間を割り当てる機能と、現在の周期に処理されるアプリケーション 3 a ~ 3 d のうち、現在の周期内にその処理が完了しないアプリケーション 3 a ~ 3 d が存在するか否かを判定する機能と、次の周期に処理されるアプリケーション 3 a ~ 3 d の割当 CPU 時間の合計と、周期の長さとの差分とから、次の周期での CPU 空き時間を算出する機能と、次の周期での CPU 空き時間を上限として、現在の周期内にその処理が完了しないアプリケーションに対して、CPU 時間をさらに割り当てる機能とを実現させることを特徴とする。

【選択図】 図 2



【特許請求の範囲】**【請求項 1】**

一定周期ごとに複数のアプリケーションを実行するCPUを備える情報処理装置の制御プログラムであって、

情報処理装置に、

現在の周期内に、次の周期に実行する各アプリケーションに対してCPU時間を割り当てる第1の割当機能と、

前記第1の割当機能によって割り当てられたCPU時間を記憶手段へ書き込む書込機能と、

現在の周期に実行するアプリケーションのうち、現在の周期内にその実行処理が完了しないアプリケーションが存在するか否かを判定する判定機能と、

10

現在の周期内に実行処理が完了しないアプリケーションが存在すると判定された場合に、前記記憶手段に記憶された各アプリケーションに対して割り当てられたCPU時間の合計と、周期の長さとの差分から、次の周期におけるCPUの空き時間を算出する算出機能と、

前記次の周期におけるCPUの空き時間を上限として、前記現在の周期内に実行処理が完了しないアプリケーションに対して、CPU時間をさらに割り当てる第2の割当機能とを実現させることを特徴とする情報処理装置の制御プログラム。

【請求項 2】

前の周期が現在の周期に切り替わるまでに、現在の周期に実行する各アプリケーションに対してCPU時間を割り当てる第3の割当機能と、

20

前記現在の周期に処理される各アプリケーションに対して割り当てられたCPU時間を第2の記憶手段へ書き込む第2の書込機能とを、

前記判定機能によって現在の周期内に実行処理が完了しないアプリケーションが存在すると判定された場合に、前記第2の記憶手段に記憶された各アプリケーションに対して割り当てられたCPU時間の合計と、周期の長さとの差分から、現在の周期におけるCPUの空き時間を算出する第2の算出機能と、

前記現在の周期におけるCPUの空き時間を上限として、前記現在の周期内に実行処理が完了しないアプリケーションに対して、CPU時間をさらに割り当てる第4の割当機能とを、前記情報処理装置に実現させることを特徴とする請求項1に記載の情報処理装置の制御プログラム。

30

【請求項 3】

前記現在の周期におけるCPUの空き時間がない場合に、前記第2の割当機能を、前記情報処理装置に実現させることを特徴とする請求項2に記載の情報処理装置の制御プログラム。

【請求項 4】

前記判定機能は、前記情報処理装置に、現在の周期内にその実行処理が完了しないアプリケーションが存在するか否かを判定する時点において、当該アプリケーションの実行処理に要しているCPU時間と、当該アプリケーションの進捗度とから、当該アプリケーションの実行処理が現在の周期内に完了するか否かを判定させることを特徴とする請求項1または請求項2に記載の情報処理装置の制御プログラム。

40

【請求項 5】

前記現在の周期に処理されるアプリケーションのうち少なくとも1つは、前記判定機能を、前記情報処理装置に実現させることを特徴とする請求項1に記載の情報処理装置の制御プログラム。

【請求項 6】

前記アプリケーションは所定の処理を繰り返すためのループコードを有し、

前記アプリケーションは、前記ループコードによって繰り返した所定の処理の実行回数と、それまでの実行処理に要したCPU時間とを用いて、前記判定機能を、前記情報処理装置に実現させることを特徴とする請求項5に記載の情報処理装置の制御プログラム。

50

【請求項 7】

前記アプリケーションは、当該アプリケーションの実行処理の進捗度を示す進捗度指示コードを有し、前記進捗度指示コードが示すアプリケーションの実行処理の進捗度を用いて、前記判定機能を、前記情報処理装置に実現させることを特徴とする請求項 6 に記載の情報処理装置の制御プログラム。

【請求項 8】

一定周期ごとに複数のアプリケーションを実行する CPU を備える情報処理装置の制御方法であって、

現在の周期内に、次の周期に実行する各アプリケーションに対して CPU 時間を割り当て、

前記次の周期に処理される各アプリケーションに対して割り当てられた CPU 時間を記憶手段へ書き込み、

現在の周期に実行するアプリケーションのうち、現在の周期内にその実行処理が完了しないアプリケーションが存在するか否かを判定し、

現在の周期内に実行処理が完了しないアプリケーションが存在すると判定された場合に、前記記憶手段に記憶された各アプリケーションに対して割り当てられた CPU 時間の合計と、周期の長さとの差分から、次の周期における CPU の空き時間を算出し、

前記次の周期における CPU の空き時間を上限として、前記現在の周期内に実行処理が完了しないアプリケーションに対して、CPU 時間をさらに割り当てることを特徴とする情報処理装置の制御方法。

【請求項 9】

一定周期ごとに複数のアプリケーションを実行する CPU と、次の周期に処理される各アプリケーションに対して割り当てられた CPU 時間を記憶手段とを備える情報処理装置であって、

前記 CPU が、

現在の周期内に、次の周期に実行する各アプリケーションに対して CPU 時間を割り当てる第 1 の割当手段と、

前記第 1 の割当手段によって割り当てられた CPU 時間を前記記憶手段へ書き込む書込手段と、

現在の周期に実行するアプリケーションのうち、現在の周期内にその実行処理が完了しないアプリケーションが存在するか否かを判定する判定手段と、

現在の周期内に実行処理が完了しないアプリケーションが存在すると判定された場合に、前記記憶手段に記憶された各アプリケーションに対して割り当てられた CPU 時間の合計と、周期の長さとの差分から、次の周期における CPU の空き時間を算出する算出手段と、

前記次の周期における CPU の空き時間を上限として、前記現在の周期内に実行処理が完了しないアプリケーションに対して、CPU 時間をさらに割り当てる第 2 の割当手段とを備えることを特徴とする情報処理装置。

【請求項 10】

一定周期ごとに複数のアプリケーションを実行する CPU と、

現在の周期内に、次の周期に実行する各アプリケーションに対して CPU 時間を割り当てる第 1 の割当手段と、

前記第 1 の割当手段によって割り当てられた CPU 時間を記憶する記憶手段と、

現在の周期に実行するアプリケーションのうち、現在の周期内にその実行処理が完了しないアプリケーションが存在するか否かを判定する判定手段と、

現在の周期内に実行処理が完了しないアプリケーションが存在すると判定された場合に、前記記憶手段に記憶された各アプリケーションに対して割り当てられた CPU 時間の合計と、周期の長さとの差分から、次の周期における CPU の空き時間を算出する算出手段と、

前記次の周期における CPU の空き時間を上限として、前記現在の周期内に実行処理が

10

20

30

40

50

完了しないアプリケーションに対して、CPU時間をさらに割り当てる第2の割り当て手段とを備えることを特徴とする情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置、情報処理装置の制御方法および情報処理装置の制御プログラムに関するものである。

【背景技術】

【0002】

船舶、航空機、自動車などの輸送機械には、センサによって把握した周囲の状況をリアルタイムに画面に表示させるセンサ情報システムが用いられている。例えば、船舶は、レーダー、ソナー、赤外線センサ、カメラなどの複数のセンサを備え、各局面に応じて複数のセンサのセンシング結果を解析することにより周囲の状況を把握する。周囲の状況を把握することにより、船舶は、衝突事故を回避し、安全な航行を実現する。

10

【0003】

輸送機械に備えられるセンサ情報システムでは、センサによって把握した周囲の状況を即座にユーザへ知らせるために、所定時間以下の周期ごとに、画面に表示される周囲の状況が更新されなければならない。

【0004】

このため、センサ情報システムでは、一定周期ごとに、各種センサからのセンシング結果を解析するアプリケーションや、その解析結果を画面に表示するアプリケーションが実行される。このようなアプリケーションは、ある周期に実行処理が開始されたときにはその周期内に実行処理が終了することが要求されるものの、周期内であればどのようなタイミングで実行されたとしても問題がないという特徴がある。

20

【0005】

このような特徴を持つアプリケーションに対してCPU(Central Processing Unit)時間の割り当て(以下、CPU時間のスケジューリングと呼ぶ)を行う際には、ある周期においてアプリケーションに割り当てられるCPU時間の合計が重要となる。

【0006】

このような周期を考慮したセンサ信号解析用のアプリケーションに対してスケジューリングを行う方式として、例えば、市場メカニズムに基づくQoS(Quality of Service)適応リソース割り当て技術がある(例えば、非特許文献1。)

30

【0007】

上記の非特許文献1に開示されている技術では、各アプリケーションが仮想的な通貨を用いてCPU時間の入札を行い、CPU時間を所定単位ごとに獲得する。

【非特許文献1】“市場メカニズムに基づくQoS適応リソース割り当て技術”，東芝レビュー，Vol.62，No.3，2007，(2007年3月)，インターネット<http://www.toshiba.co.jp/tech/review/2007/03/62_03pdf/01_1.pdf>

40

【発明の開示】

【発明が解決しようとする課題】

【0008】

上記の非特許文献1に開示されている技術では、アプリケーションの実行処理に要する時間を事前に正しく見積もる必要がある。しかし、各種センサとの通信遅延やキャッシュミスなどの要因によって、実際にアプリケーションの実行処理に要する時間は変動する。

【0009】

そのため、たとえ、アプリケーションの実行処理に要する時間を事前に見積もり、周期内にアプリケーションの実行処理が完了するようにCPU時間を割り当てたとしても、その周期内にアプリケーションの実行処理が完了しない事態が発生する。

50

【 0 0 1 0 】

このように周期内にアプリケーションの実行処理が完了しない場合、そのアプリケーションの実行処理は途中で打ち切られる。アプリケーションの実行処理を途中で打ち切ることにより、センシング結果の解析が十分に行われない、あるいは周囲の状況を示す情報が画面に正しく表示されないなどの不具合が発生し、例えば、船舶の安全な航行に支障をきたす恐れがある。

【 0 0 1 1 】

本発明は、上記を鑑みてなされたものであって、周期ごとに発生するアプリケーションの実行処理の打ち切りを抑制できる情報処理装置、情報処理装置の制御方法、および情報処理装置の制御プログラムを提供することを目的とする。

10

【課題を解決するための手段】

【 0 0 1 2 】

上記目的を達成するために、本発明の実施形態に係る情報処理装置の制御プログラムは、一定周期ごとに複数のアプリケーションを実行するCPUを備える情報処理装置の制御プログラムであって、情報処理装置に、現在の周期内に、次の周期に実行する各アプリケーションに対してCPU時間を割り当てる第1の割当機能と、前記第1の割当機能によって割り当てられたCPU時間を記憶手段へ書き込む書込機能と、現在の周期に実行するアプリケーションのうち、現在の周期内にその実行処理が完了しないアプリケーションが存在するか否かを判定する判定機能と、現在の周期内に実行処理が完了しないアプリケーションが存在すると判定された場合に、前記記憶手段に記憶された各アプリケーションに対して割り当てられたCPU時間の合計と、周期の長さとの差分から、次の周期におけるCPUの空き時間を算出する算出機能と、前記次の周期におけるCPUの空き時間を上限として、前記現在の周期内に実行処理が完了しないアプリケーションに対して、CPU時間をさらに割り当てる第2の割当機能とを実現させることを特徴とする。

20

【発明の効果】

【 0 0 1 3 】

本発明によれば、周期ごとに発生するアプリケーションの実行処理の打ち切りを抑制できる。

【発明を実施するための最良の形態】

【 0 0 1 4 】

30

以下、本発明の実施形態について説明する。

【 0 0 1 5 】

(第1の実施形態)

図1は、本発明の第1の実施形態に係るセンサ情報システム100の構成を示すブロック図である。

この第1の実施形態に係るセンサ情報システム100は、情報処理装置10と、レーダー装置20と、ソナー装置30と、入力装置40と、表示装置50とを備える。情報処理装置10は、CPU12Aと、CPU12Bと、メインメモリ11と、入出力インタフェース13A～13Dとを備える。CPU12Aはキャッシュメモリ12A-1を内蔵し、CPU12Bはキャッシュメモリ12B-1を内蔵する。

40

【 0 0 1 6 】

CPU12A、CPU12B、メインメモリ11、および入出力インタフェース13A～13Dは、互いにバスラインで接続される。入出力インタフェース13Aは入力装置40と接続される。入出力インタフェース13Bは表示装置50と接続される。入出力インタフェース13Cはレーダー装置20と接続される。入出力インタフェース13Dはソナー装置30と接続される。

【 0 0 1 7 】

このセンサ情報システム100は、船舶、航空機などの輸送機械に搭載される。センサ情報システム100は、レーダー装置20とソナー装置30によって周囲の状況を測定し、その測定結果を表示装置50に画面情報として表示することにより、例えば、危険な障

50

害物などをいち早く乗組員（ユーザ）に知らせるためのリアルタイムシステムである。なお、センサ情報システム 100 は、1000ms ごとに、レーダー装置 20 とソナー装置 30 によって周囲の状況を測定し、逐次その測定結果を画面情報として表示する。

【0018】

レーダー装置 20 は、電波を発信し、障害物や目標物からの電波の反射波を測定する。レーダー装置 20 は、入出力インタフェース 13A を介して、測定結果である電波の反射波に関する情報（以下、センシング結果と呼ぶ。）をメインメモリ 11 に記憶させる。なお、レーダー装置 20 のセンシング結果を解析することによって、周囲に存在する障害物や目標物の距離や方位などを把握できる。

【0019】

ソナー装置 30 は、音波を発信し、障害物や目標物からの音波の反射波を測定する。ソナー装置 30 は、入出力インタフェース 13B を介して、測定結果である音波の反射波に関する情報（以下、センシング結果と呼ぶ。）をメインメモリ 11 に記憶させる。なお、ソナー装置 30 のセンシング結果を解析することによって、周囲に存在する障害物や目標物の距離や方位などを把握できる。

【0020】

入力装置 40 は、キーボードやマウスなど、ユーザが情報を入力するための装置である。ユーザは、入力装置 40 を操作することによって、ソナー装置 30 およびレーダー装置 20 によるセンシング結果を、どのように表示装置 50 に表示させるかを指定する。ユーザにより入力された情報は、入出力インタフェース 13C を介して、CPU 12A、12B へ伝達される。

【0021】

情報処理装置 10 は、レーダー装置 20 およびソナー装置 30 によるセンシング結果を解析する。情報処理装置 10 は、入力装置 40 から入力された情報に従って、その解析結果から画面情報を作成し、入出力インタフェース 13D を介して表示装置 50 へ送信する。

【0022】

表示装置 50 は、情報処理装置 10 から送信された画面情報を表示するディスプレイである。

【0023】

図 2 (a) は CPU 12A が実行するソフトウェアの構成を示す図であり、図 2 (b) は CPU 12B が実行するソフトウェアの構成を示す図である。

CPU 12A と CPU 12B は、マルチプロセッサに対応した同一の OS 1 (Operating System)、例えば、Windows (R)、Linux (R) を実行する。CPU 12A は、OS 1 の他、ミドルウェアとしてスケジューラ 2 を実行する。CPU 12B は、OS 1 の他、レーダー信号解析用アプリケーション 3a、レーダー画面表示用アプリケーション 3b、ソナー信号解析用アプリケーション 3c、およびソナー画面表示用アプリケーション 3d を実行する。

【0024】

OS 1 に従って、CPU 12A と CPU 12B は、メインメモリ 11 へのアクセス、入出力インタフェース 13A ~ 13D の制御、入力装置 40 からの入力情報の処理、表示装置 50 へ画面情報の表示などを行う。OS 1 に従って、CPU 12A と CPU 12B は、実行可能な状態にある複数のプログラム（例えば、スケジューラ 2 や各アプリケーション 3）に対して、ラウンドロビン方式によってタイムスライス (50ms) ごとに CPU 時間を割り当て、複数のプログラムを並列に実行する。なお、プログラムに割り当てられる CPU 時間とは、CPU 12A あるいは CPU 12B がそのプログラムの実行処理を行う時間である。

【0025】

スケジューラ 2 に従って、CPU 12A は、各アプリケーション 3 に、周期ごとに割り当てる CPU 時間を決定する。なお、CPU 12A は、それぞれのアプリケーション 3 の

10

20

30

40

50

優先度に応じて、アプリケーションごとにCPU時間を割り当てる。

【0026】

レーダー信号解析用アプリケーション3aに従って、CPU12Bは、レーダー装置20からのセンシング結果をメインメモリ11から読み出し、そのセンシング結果を解析して、周囲に存在する障害物や目標物の距離や方位などを示す情報を構成する。

【0027】

レーダー画面表示用アプリケーション3bに従って、CPU12Bは、レーダー信号解析用アプリケーション3aを実行することによって得られた解析結果を、表示装置50に表示するための画面情報に変替する。

【0028】

ソナー信号解析用アプリケーション3cに従って、CPU12Bは、ソナー装置30からのセンシング結果をメインメモリ11から読み出し、そのセンシング結果を解析して、周囲に存在する障害物や目標物の距離や方位などを示す情報を構成する。

【0029】

ソナー画面表示用アプリケーション3dに従って、CPU12Bは、ソナー信号解析用アプリケーション3cを実行することによって得られた解析結果を、表示装置50に表示するための画面情報に変替する。

【0030】

なお、センサ情報システム100は、1000msごとに、レーダー装置20とソナー装置30によって周囲の状況を測定し、逐次その測定結果を画面情報として表示するため、CPU12AとCPU12Bは、周期(1000ms)内に、レーダー信号解析用アプリケーション3a、レーダー画面表示用アプリケーション3b、ソナー信号解析用アプリケーション3c、およびソナー画面表示用アプリケーション3dの実行処理を完了する必要がある。

【0031】

図3は、CPU12Aが実行するスケジューラ2と、CPU12Bが実行するレーダー画面表示用アプリケーション3bとの関係を示す。なお、CPU12Aが実行するスケジューラ2と、CPU12Bが実行するレーダー信号解析用アプリケーション3a、ソナー信号解析用アプリケーション3c、およびソナー画面表示用アプリケーション3dとの関係も同様である。

【0032】

図4は、スケジューラ2およびレーダー画面表示用アプリケーション3bを実行する際の、一周期におけるCPU12AとCPU12Bの動作を示すフローチャートである。なお、センサ情報システム100のCPU12AとCPU12Bは、周期ごとに同様の処理を繰り返す。

【0033】

まず、前の周期(1000ms)が終了し、現在の周期(1000ms)へ切り替わる(ステップS101)。なお、前の周期において、現在の周期における各アプリケーション3に対するCPU時間の割り当て結果が、次周期用記憶配列2-3として、メインメモリ11に記憶されているものとする。

【0034】

図5は、現在の周期における各アプリケーション3に対するCPU時間の割り当て結果である。なお、周期の長さが1000msであり、各アプリケーション3に割り当てられたCPU時間の合計は800msであるため、現在の周期におけるCPU空き時間(未割当CPU時間)は200msである。

【0035】

次に、CPU12Aは、スケジューラ2の割当コード2-1に従い、次周期用記憶配列2-3としてメインメモリ11に記憶された、現在の周期における各アプリケーション3に対するCPU時間の割り当て結果を、現周期用記憶配列2-2としてメインメモリ11に記憶しなおす(ステップS102)。

10

20

30

40

50

【 0 0 3 6 】

次に、CPU 1 2 A は、割当コード 2 - 1 に従い、次の周期における各アプリケーション 3 に対する CPU 時間の割り当てを行い、その割り当て結果を、次周期用記憶配列 2 - 3 として、メインメモリ 1 1 に記憶する（ステップ S 1 0 3）。その後、CPU 1 2 A は待機状態となる。

【 0 0 3 7 】

図 6 は、次の周期における各アプリケーション 3 に対する CPU 時間の割り当て結果である。なお、周期の長さが 1 0 0 0 m s e c であり、各アプリケーション 3 に割り当てられた CPU 時間の合計は 8 5 0 m s e c であるため、次周期における CPU 空き時間（未割当 CPU 時間）は 1 5 0 m s e c である。

10

【 0 0 3 8 】

次に、CPU 1 2 B は、レーダー画面表示用アプリケーション 3 b に対して CPU 時間が割り当てられているため、レーダー画面表示用アプリケーション 3 b の実行を開始する。なお、CPU 1 2 B は、同様に、レーダー信号解析用アプリケーション 3 a、ソナー信号解析用アプリケーション 3 c、およびソナー画面表示用アプリケーション 3 d を実行するが、その説明については省略する。

【 0 0 3 9 】

次に、CPU 1 2 B は、レーダー画面表示用アプリケーション 3 b の実行コード 3 b - 1 に従い、レーダー装置 2 0 のセンシング結果の解析を行う（ステップ S 1 0 4）。

【 0 0 4 0 】

レーダー画面表示用アプリケーション 3 b の実行処理が終了した場合（ステップ S 1 0 5 のはい）、CPU 1 2 B は、その他のアプリケーションを実行するか、その動作を終了する。

20

【 0 0 4 1 】

一方、レーダー画面表示用アプリケーション 3 b の実行処理が終了していない場合（ステップ S 1 0 5 のいいえ）には、CPU 1 2 B は、進捗管理コード 3 b - 2 に従って、レーダー画面表示用アプリケーション 3 b の実行コード 3 b - 1 の進捗状況を調べ、レーダー画面表示用アプリケーション 3 b に割り当てられた CPU 時間 5 0 m s e c 以内に、その実行処理が完了するか否かを判定する（ステップ S 1 0 6）。

【 0 0 4 2 】

割り当てられた CPU 時間 5 0 m s e c 以内にレーダー画面表示用アプリケーション 3 b の実行処理が完了すると判定された場合（ステップ S 1 0 6 のはい）には、CPU 1 2 B は、実行コード 3 b - 1 の実行処理を継続する（ステップ S 1 0 4）。

30

【 0 0 4 3 】

一方、割り当てられた CPU 時間 5 0 m s e c 以内にレーダー画面表示用アプリケーション 3 b の実行処理が完了しないと判定された場合（ステップ S 1 0 6 のいいえ）には、CPU 1 2 B は、進捗管理コード 3 b - 2 に従って、CPU 時間の再割り当てを CPU 1 2 A に対して要求する。

【 0 0 4 4 】

レーダー画面表示用アプリケーション 3 b を実行する CPU 1 2 B からの CPU 時間の再割り当て要求を受信した場合、CPU 1 2 A は、スケジューラ 2 の割当コード 2 - 1 に従い、現周期用記憶配列 2 - 2（図 5）として記憶された未割り当て CPU 時間 2 0 0 m s e c を上限として、レーダー画面表示用アプリケーション 3 b に対して CPU 時間の再割り当てを行う（ステップ S 1 0 7 のはい、S 1 0 8）。

40

【 0 0 4 5 】

この CPU 時間の再割り当てによってレーダー画面表示用アプリケーション 3 b の実行処理用に割り当てられた CPU 時間に不足がなくなった場合（ステップ S 1 0 9 のはい）、CPU 1 2 A は、CPU 時間の再割り当てを行ったことを CPU 1 2 B に対して通知する。CPU 1 2 B は、CPU 時間の再割り当てが行われたレーダー画面表示用アプリケーション 3 b に従い、さらに、実行コード 3 b - 1 や進捗管理コード 3 b - 2 を実行する（

50

ステップ S 1 0 4、S 1 0 6)。

【 0 0 4 6 】

一方、レーダー画面表示用アプリケーション 3 b を実行する CPU 1 2 B からの CPU 時間の再割り当て要求を受信したとしても、現周期用記憶配列 2 - 2 として記憶された未割り当て CPU 時間が存在しない場合 (ステップ S 1 0 7 のいいえ)、あるいは、CPU 時間の再割り当てを行ったにも関わらずレーダー画面表示用アプリケーション 3 b の実行処理用に割り当てられた CPU 時間が不足する場合 (ステップ S 1 0 9 のいいえ)、CPU 1 2 B は、スケジューラ 2 の割り当てコード 2 - 1 に従い、次周期用記憶配列 2 - 3 (図 6) として記憶された未割り当て CPU 時間 1 5 0 m s e c を上限として、レーダー画面表示用アプリケーション 3 b に対して CPU 時間の再割り当てを行う (ステップ S 1 1 0 のはい、S 1 1 1)。CPU 1 2 A は、CPU 時間の再割り当てを行ったことを CPU 1 2 B に対して通知する。CPU 1 2 B は、CPU 時間の再割り当てが行われたレーダー画面表示用アプリケーション 3 b に従い、さらに、実行コード 3 b - 1 や進捗管理コード 3 b - 2 を実行する (ステップ S 1 0 4、S 1 0 6)。

10

【 0 0 4 7 】

一方、次周期用記憶配列 2 - 3 として記憶された未割り当て CPU 時間が存在しない場合 (ステップ S 1 1 0 のいいえ) は、CPU 1 2 B は、レーダー画面表示用アプリケーション 3 b の実行処理を打ち切る (ステップ S 1 1 2)。

【 0 0 4 8 】

なお、レーダー画面表示用アプリケーション 3 b の実行処理の打ち切りは、CPU 1 2 B が、レーダー画面表示用アプリケーション 3 b に従って行ってもよく、OS 1 に従って行っても良い。レーダー画面表示用アプリケーション 3 b が第三者によって作成された場合やレーダー画面表示用アプリケーション 3 b に不具合がある場合などは、CPU 1 2 B は、OS 1 に従って、強制的にレーダー画面表示用アプリケーション 3 b の実行処理を打ち切る。

20

【 0 0 4 9 】

図 7 は、CPU 1 2 A と CPU 1 2 B がスケジューラ 2 および各アプリケーション 3 を実行したときの処理の様子の一例を示す図である。なお、現在の周期における各アプリケーション 3 に対する CPU 時間の割り当て結果を図 5 とする。次周期における各アプリケーション 3 に対する CPU 時間の割り当て結果を図 6 とする。また、現在の周期において実際に各アプリケーション 3 を実行した際の各アプリケーション 3 の実行処理に要した CPU 時間が図 8 であったものとする。

30

【 0 0 5 0 】

まず、前の周期から現在の周期へ切り替わった際に、CPU 1 2 A はスケジューラ 2 の実行を開始し、CPU 1 2 B はレーダー信号解析用アプリケーション 3 a とソナー信号解析用アプリケーション 3 c の実行を開始する。

【 0 0 5 1 】

CPU 1 2 B は、前の周期で事前に割り当てられた CPU 時間 (図 5) に従い、レーダー信号解析用アプリケーション 3 a とソナー信号解析用アプリケーション 3 c とを、タイムスライス (5 0 m s e c) ごとに交互に実行する。

40

【 0 0 5 2 】

レーダー信号解析用アプリケーション 3 a とソナー信号解析用アプリケーション 3 c の実行処理が前の周期で事前に割り当てられた CPU 時間内には完了せず、現在の周期の未割り当て CPU 時間が存在するため、現在の周期の未割り当て CPU 時間から、CPU 時間の再割り当てが行われる。CPU 1 2 B は、前の周期で事前に割り当てられた CPU 時間と、再割り当てされた CPU 時間とをかけた、レーダー信号解析用アプリケーション 3 a とソナー信号解析用アプリケーション 3 c との実行処理を行う。

【 0 0 5 3 】

レーダー信号解析用アプリケーション 3 a とソナー信号解析用アプリケーション 3 c の実行処理が完了した後、CPU 1 2 B は、それぞれレーダー画面表示用アプリケーション

50

3 b とソナー画面表示用アプリケーション 3 d の実行処理を開始する。

【 0 0 5 4 】

C P U 1 2 B は、ソナー画面表示用アプリケーション 3 d の実行処理を、前の周期で事前に割り当てられた C P U 時間内に完了した。一方、レーダー画面表示用アプリケーション 3 b の実行処理は、前の周期で事前に割り当てられた C P U 時間内には完了しなかった。ここで、現在の周期の未割当 C P U 時間が存在せず、かつ、次周期の未割当 C P U 時間が存在するので、C P U 1 2 A は、スケジューラ 2 に従って、次周期の未割当 C P U 時間から、C P U 時間の再割り当てを行う。C P U 1 2 B は、前の周期で事前に割り当てられた C P U 時間と、再割り当てされた C P U 時間とをかけた、レーダー画面表示用アプリケーション 3 b の実行処理を行う。

10

【 0 0 5 5 】

図 9 は、表示装置 5 0 の画面のピクセルを示す図である。

表示装置 5 0 は、縦 m 個 (m は 1 以上の整数)、横 n 個 (n は 1 以上の整数)、全体として $m \times n$ 個のピクセルを有する。C P U 1 2 B は、レーダー画面表示用アプリケーション 3 b に従い、レーダー信号解析用アプリケーション 3 a を実行することにより得られた解析結果から、各ピクセルの輝度を算出する。なお、C P U 1 2 B は、レーダー画面表示用アプリケーション 3 b に従い、各ピクセルの色彩 (R G B : R e d - G r e e n - B l u e) を算出して良い。

【 0 0 5 6 】

図 1 0 は、レーダー画面表示用アプリケーション 3 b のコード 3 b - c o d e の一例を示す図である。なお、ソナー画面表示用アプリケーション 3 d についても、レーダー信号解析用アプリケーション 3 a を実行することにより得られた解析結果ではなく、ソナー信号解析用アプリケーション 3 c を実行することにより得られた解析結果を用いる点以外、図 1 0 と同様のコードで実現されるものと理解される。

20

【 0 0 5 7 】

レーダー画面表示用アプリケーション 3 b は、前処理コード 3 b - c o d e 1 と、ピクセル輝度の算出コード 3 b - c o d e 2 と、進捗状況算出コード 3 b - c o d e 3 と、ループコード 3 b - c o d e 4 と、後処理コード 3 b - c o d e 5 とを有する。

【 0 0 5 8 】

前処理コード 3 b - c o d e 1 は、メインメモリ 1 1 に記憶された解析結果の読み出しなどを行うためのコードである。

30

【 0 0 5 9 】

ピクセル輝度の算出コード 3 b - c o d e 2 は、レーダー信号解析用アプリケーション 3 a を実行することにより得られた解析結果から、 i 行 j 列のピクセルの輝度を算出するためのコードである。“ i ” と “ j ” はループカウンタ値であり、“ i ” は “1” から “ m ” まで、“ j ” は “1” から “ n ” まで、スイープされる。

【 0 0 6 0 】

進捗状況算出コード 3 b - c o d e 3 は、進捗状況算出コード 3 b - c o d e 3 を実行する時点におけるレーダー画面表示用アプリケーション 3 b の進捗状況を $[\{ (i / m) * 1 0 0 \} \%]$ なる式により算出するためのコードである。例えば、表示装置 5 0 が 1 6 0 0 \times 1 2 0 0 ピクセルで画面情報を表示するときには、“ m ” は 1 2 0 0 行となり、ループカウンタ値 “ i ” が 6 0 0 であれば、進捗度は、 $\{ (6 0 0 / 1 2 0 0) * 1 0 0 \} \% = 5 0 \%$ と算出される。なお、進捗状況算出コード 3 b - c o d e 3 は、前処理コード 3 b - c o d e 1 や後処理コード 3 b - c o d e 5 など考慮した式によって、レーダー画面表示用アプリケーション 3 b の進捗状況を算出するためのコードであっても良い。

40

【 0 0 6 1 】

進捗状況算出コード 3 b - c o d e 3 は、図 1 0 に示す例ではループ処理を “ n ” 回繰り返すたびに実行されるが、ループ処理を 1 回行うたびに実行されるものであってもよい。また、進捗状況算出コード 3 b - c o d e 3 は、ループ処理の回数とは無関係に、タイマーが所定の時刻になったときに実行されるものであってもよい。例えば、周期 1 0 0 0

50

m s e c の 9 0 %、すなわち、現在の周期に切り替わってから 9 0 0 m s e c 経過した時点で、進捗状況算出コード 3 b - c o d e 3 が実行されても良い。

【 0 0 6 2 】

ループコード 3 b - c o d e 4 は、例えば、図 1 0 に示す “ f o r 文 ” であり、ピクセル輝度の算出コード 3 b - c o d e 2 と進捗状況算出コード 3 b - c o d e 3 とを、定期的に繰り返し実行するためのコードである。

【 0 0 6 3 】

後処理コード 3 a b - c o d e 5 は、各ピクセルの輝度から表示装置 5 0 に送信する画面情報を構成するためのコードである。

【 0 0 6 4 】

図 1 1 は、レーダー信号解析用アプリケーション 3 a のコード 3 a - c o d e の一例を示す図である。なお、ソナー信号解析用アプリケーション 3 c についても、レーダー装置 2 0 のセンシング結果ではなく、ソナー装置 3 0 のセンシング結果を用いる点以外、図 1 1 と同様のコードで実現されるものと理解される。

【 0 0 6 5 】

レーダー信号解析用アプリケーション 3 a は、前処理コード 3 a - 1 と、反射データの解析コード 3 a - c o d e 2 と、進捗状況算出コード 3 a - c o d e 3 と、ループコード 3 a - c o d e 4 と、後処理コード 3 a - c o d e 5 とを有する。

【 0 0 6 6 】

前処理コード 3 a - c o d e 1 は、メインメモリ 1 1 に記憶されたセンシング結果の読み出しなどを行うためのコードである。

【 0 0 6 7 】

反射データの解析コード 3 a - c o d e 2 は、レーダー装置 2 0 のセンシング結果、即ち、仰角 “ i ”、方位角 “ j ” に対して電波を送信した際の反射波の情報を示すデータを、それぞれ解析するためのコードである。“ i ” と “ j ” はループカウンタ値であり、“ i ” は “ 1 ” から “ p ” まで、“ j ” は “ 1 ” から “ q ” まで、スイープされる。

【 0 0 6 8 】

進捗状況算出コード 3 a - c o d e 3 は、進捗状況算出コード 3 a - c o d e 3 を実行する時点におけるレーダー画面表示用アプリケーション 3 b の進捗状況を $[\{ (i / p) * 1 0 0 \} \%]$ なる式により算出するためのコードである。なお、進捗状況算出コード 3 a - c o d e 3 は、前処理コード 3 a - c o d e 1 や後処理コード 3 a - c o d e 5 などを考慮した式によって、レーダー信号解析用アプリケーション 3 a の進捗状況を算出するためのコードであっても良い。

【 0 0 6 9 】

ループコード 3 a - c o d e 4 は、例えば、図 1 1 に示す “ f o r 文 ” であり、反射データの解析コード 3 a - c o d e 2 と進捗状況算出コード 3 a - c o d e 3 とを、定期的に繰り返し実行するためのコードである。

【 0 0 7 0 】

後処理コード 3 a - c o d e 5 は、それぞれの反射データの解析結果を総合して、レーダー装置 2 0 からのセンシング結果を解析した結果を構成するためのコードである。

【 0 0 7 1 】

図 1 2 は、図 4 のステップ S 1 0 6 において、C P U 1 2 B が、各アプリケーション 3 の進捗管理コード 3 b - 2 に従って、現在の周期において割り当てられた C P U 時間内に、実行中のアプリケーションの実行処理が完了するか否かを判定する際の動作を示すフローチャートである。なお、進捗管理コード 3 b - 2 は、進捗状況算出コード 3 b - c o d e 3 を有する。

【 0 0 7 2 】

まず、C P U 1 2 B は、ループカウンタ値を総ループ回数で除算した結果を、進捗度とする (ステップ S 2 0 1)。

【 0 0 7 3 】

10

20

30

40

50

次に、CPU12Bは、現在までにアプリケーションの実行処理に要したCPU時間を、進捗度で除算した結果を、実行中のアプリケーションの実行処理に要すると予想されるCPU時間（以下、予想総CPU時間と呼ぶ）とする（ステップS202）。なお、CPU12Bは、OS1に従って、現在までにアプリケーションの実行処理に要したCPU時間を管理しているものとする。

【0074】

次に、CPU12Bは、予想総CPU時間から、現在の周期において割り当てられたCPU時間（以下、現周期の割当CPU時間と呼ぶ）を減算した結果を、アプリケーションの実行処理を行うにあたって不足するCPU時間（以下、不足時間と呼ぶ）とする（ステップS203）。

10

【0075】

不足時間が“0”以下であれば（ステップS204のいいえ）、CPU12Bは、実行中のアプリケーションの実行処理が、現周期の割当CPU時間内に、完了すると判定する（ステップS205）。

【0076】

一方、不足時間が“0”より大きければ（ステップS204のはい）、CPU12Bは、実行中のアプリケーションの実行処理が、現周期の割当CPU時間内に、完了しないと判定する（ステップS206）。

【0077】

図13は、図4のステップS107～S109において、CPU12Aが、スケジューラ2の割当コード2-1に従って、現周期の未割当CPU時間から、アプリケーションに対してCPU時間の再割当を行う際の動作を示すフローチャートである。

20

【0078】

まず、CPU12Aは、CPU12Bから、CPU時間の再割当要求として、アプリケーションの種別と、そのアプリケーションの実行に不足する不足時間とを受信する。

【0079】

次に、CPU12Aは、不足時間が現周期の未割当CPU時間よりも大きいかなんかを判定する（ステップS301）。なお、不足時間は、CPU12Bが、図12のステップS203において、進捗管理コード3b-2にしたがって算出したものである。

【0080】

不足時間が現周期の未割当CPU時間よりも大きいと判定された場合（ステップS301のはい）、CPU12Aは、アプリケーションに対してCPU時間の再割当を行う際に、追加するCPU時間（以下、追加割当時間と呼ぶ）を、現周期の未割当CPU時間の残りすべてとする（ステップS302）。

30

【0081】

一方、不足時間が現周期の未割当CPU時間より大きくないと判定された場合（ステップS301のいいえ）、CPU12Aは、追加割当時間を不足時間とする（ステップS303）。

【0082】

次に、CPU12Aは、現周期の未割当CPU時間から追加割当時間を減算した減算結果へ、現周期の未割当CPU時間を更新する（ステップS304）。

40

【0083】

次に、CPU12Aは、不足時間から追加割当時間を減算した減算結果へ、不足時間を更新する（ステップS305）。

【0084】

次に、CPU12Aは、CPU時間の再割当を行うアプリケーションに関しての、現周期の割当CPU時間と、追加割当時間とを加算した加算結果へ、現周期の割当CPU時間を更新する（ステップS306）。

【0085】

次に、CPU12Aは、不足時間が0より大きいかなんかを判定する（ステップS307

50

)。

【0086】

不足時間が0より大きくないと判定された場合(ステップS307のいいえ)、CPU12Bは、アプリケーションに対するCPU時間の再割当を完了できたものとして(ステップS308)、他の処理、例えばアプリケーションの実行処理などへ移行する。

【0087】

一方、不足時間が0より大きいと判定された場合(ステップS307のはい)、CPU12Aは、アプリケーションに対して再割り当てしたCPU時間が不足しているものとして(ステップS309)、さらなるCPU時間の割り当てを試みる。

【0088】

図14は、図4のステップS110、111において、CPU12Aが、スケジューラ2の割当コード2-1に従って、次周期の未割当CPU時間から、アプリケーションに対してCPU時間の再割当を行う際の動作を示すフローチャートである。

【0089】

まず、CPU12Aは、不足時間が次周期の未割当CPU時間よりも大きいか否かを判定する(ステップS401)。なお、不足時間は、CPU12Aが、図12のステップS203で算出したもの、あるいは、図13のステップS305で更新したものである。

【0090】

不足時間が次周期の未割当CPU時間よりも大きいと判定された場合(ステップS401のはい)、CPU12Aは、追加割当時間を、次周期の未割当CPU時間の残りすべてとする(ステップS402)。

【0091】

一方、不足時間が次周期の未割当CPU時間より大きくないと判定された場合(ステップS401のいいえ)、CPU12Aは、追加割当時間を不足時間とする(ステップS403)。

【0092】

次に、CPU12Aは、次周期の未割当CPU時間から追加割当時間を減算した減算結果へ、次周期の未割当CPU時間を更新する(ステップS404)。

【0093】

次に、CPU12Aは、不足時間から追加割当時間を減算した減算結果へ、不足時間を更新する(ステップS405)。

【0094】

次に、CPU12Aは、CPU時間の再割当を行うアプリケーションに関しての、現周期の割当CPU時間と、追加割当時間とを加算した加算結果へ、現周期の割当CPU時間を更新する(ステップS406)。

【0095】

このように、第1の実施形態に係るセンサ情報システム100によれば、周期ごとに発生するアプリケーションの実行処理を行う際に、次の周期に実行する各アプリケーション3に対してCPU時間をあらかじめ割り当てることによって、現在の周期中に実行処理が完了しないアプリケーションが存在したとしても、次の周期におけるCPUの空き時間を上限として、実行処理が完了しないアプリケーションに対してCPU時間を再割り当てすることによって、アプリケーションの実行処理の打ち切りを抑制できる。

【0096】

また、このセンサ情報システム100のスケジューリング機能は、情報処理装置10に搭載されたCPUにプログラムを実行させることにより実現されるものとしたが、専用のハードウェアを用いることで実現することができる。

【0097】

例えば、情報処理装置10は、現周期用記憶配列2-2と次周期用記憶配列2-3とをそれぞれ記憶するメモリである現周期用記憶部と次周期用記憶部とを備えていて、各アプリケーション3の実行処理を行うCPU12A、12Bの代わりに、専用のハードウェア

10

20

30

40

50

として、アプリケーションの進捗管理コードをCPU12Bが実行することによって実現する機能を有する進捗管理部と、スケジューラ2の割り当てコード2-1をCPU12Aが実行することによって実現する機能を有するCPU時間割当部とを備えていても良い。

【0098】

(第2の実施形態)

第1の実施形態に係るセンサ情報システム100が有するアプリケーションのコードは、ループコードを有していて、例えば、ループカウンタ値を総ループ回数で除算した除算結果を、そのアプリケーションの進捗状況としていた。一方、第2の実施形態に係るセンサ情報システムが有するアプリケーションのコードは、進捗度を指示するためのコードを有している。なお、この第2の実施形態に係るセンサ情報システムと第1の実施形態に係るセンサ情報システム100との相違点は、アプリケーションのコードであるため、その他の部分については、その説明を省略する。

10

【0099】

図15は、レーダー画面表示用アプリケーション3bのコード3b-code10の構成の一例を示す図である。なお、レーダー信号解析用アプリケーション3a、ソナー信号解析用アプリケーション3cおよびソナー画面表示用アプリケーション3dについても、図15と同様の構成のコードで実現されるものと理解される。

【0100】

レーダー画面表示用アプリケーション3bのコード3b-code10は、第1乃至第4の処理コード3b-code11~3b-code14と、進捗度指示コード3b-code15~3b-code17とを有する。

20

【0101】

第1乃至第4の処理コード3b-code11~3b-code14は、レーダー画面表示用アプリケーション3bのコード3b-code10を、機能の種類、フェイズなどに応じて4つに分割して得られるコードである。

【0102】

進捗度指示コード3b-code15~3b-code17は、それぞれ、第1の処理コード3b-code11と第2の処理コード3b-code12の間、第2の処理コード3b-code12と第3の処理コード3b-code13の間、および第3の処理コード3b-code13と第4の処理コード3b-code14の間に挿入される。

30

【0103】

第1の処理コード3b-code11と第2の処理コード3b-code12の間に挿入された進捗度指示コード3b-code15は、進捗度が20%であることを示す。即ち、第1の処理コード3b-code11の実行処理に要するCPU時間は、第1乃至第4の処理コード3b-code11~3b-code14の実行に要するCPU時間の20%であることを示す。

【0104】

第2の処理コード3b-code12と第3の処理コード3b-code13の間に挿入された進捗度指示コード3b-code16は、進捗度が50%であることを示す。即ち、第1および第2の処理コード3b-code11~3b-code12の実行処理に要するCPU時間は、第1乃至第4の処理コード3b-code11~3b-code14の実行に要するCPU時間の50%であることを示す。

40

【0105】

第3の処理コード3b-code13と第4の処理コード3b-code14の間に挿入された進捗度指示コード3b-code17は、進捗度が80%であることを示す。即ち、第1乃至第3の処理コード3b-code11~3b-code13の実行処理に要するCPU時間は、第1乃至第4の処理コード3b-code11~3b-code14の実行に要するCPU時間の80%であることを示す。

【0106】

CPU12Bは、各アプリケーション3の進捗指示コードに従って現周期の割り当てCPU

50

時間内に各アプリケーション3の実行処理が完了するか否かを判定する際、進捗度の算出処理(図12のステップS201)を行うのではなく、進捗指示コードによって示された進捗度を読み出す。CPU12Bは、読み出した進捗度をもとにCPU時間の再割り当て要求を行う。なお、各アプリケーション3のコードを機能の種類、フェイズなどに応じて分割する数は4つに限定されない。

【0107】

このように、各アプリケーションのコードに、そのアプリケーションの進捗度を指定するためのコードを挿入することによって、全体がループによって構成されていないアプリケーションにおいても進捗を把握することができる。なお、アプリケーションの進捗度を指定するためのコードは上記の形態に限定されるものではなく、さまざまな形態が考えられる。

10

【0108】

なお、本発明は上記実施形態そのままに限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で構成要素を変形して具体化できる。また、上記実施形態に開示されている複数の構成要素の適宜な組み合わせにより、種々の発明を形成できる。

【図面の簡単な説明】

【0109】

【図1】本発明の第1の実施形態に係るセンサ情報システムの構成を示すブロック図。

【図2】CPUが実行するソフトウェアの構成を示すブロック図。

【図3】レーダー画面表示用アプリケーションとスケジューラの構成を示すブロック図。

20

【図4】本発明の第1の実施形態に係る情報処理装置の動作を示すフローチャート。

【図5】現在の周期において各アプリケーションに割り当てられたCPU時間を示す図。

【図6】次周期において各アプリケーションに割り当てられたCPU時間を示す図。

【図7】スケジューラと各アプリケーションが実行されるタイミングを示す図。

【図8】各アプリケーションの実行処理に要したCPU時間を示す図。

【図9】表示装置の表示部のピクセル数を示すブロック図

【図10】レーダー画面表示用アプリケーションのコードの構成の一例を示すブロック図

。

【図11】レーダー信号解析用アプリケーションのコードの構成の一例を示すブロック図

。

30

【図12】本発明の第1の実施形態に係る情報処理装置の動作を示すフローチャート。

【図13】本発明の第1の実施形態に係る情報処理装置の動作を示すフローチャート。

【図14】本発明の第1の実施形態に係る情報処理装置の動作を示すフローチャート。

【図15】レーダー画面表示用アプリケーションのコードの構成の一例を示すブロック図

。

【符号の説明】

【0110】

1・・・OS

2・・・スケジューラ

2-1・・・割当コード

40

2-2・・・現周期用記憶配列

2-3・・・次周期用記憶配列

3・・・アプリケーション

3a・・・レーダー信号解析用アプリケーション

3a-code・・・レーダー画面表示用アプリケーションのコード

3a-code1・・・前処理コード

3a-code2・・・反射データの解析コード

3a-code3・・・進捗状況算出コード

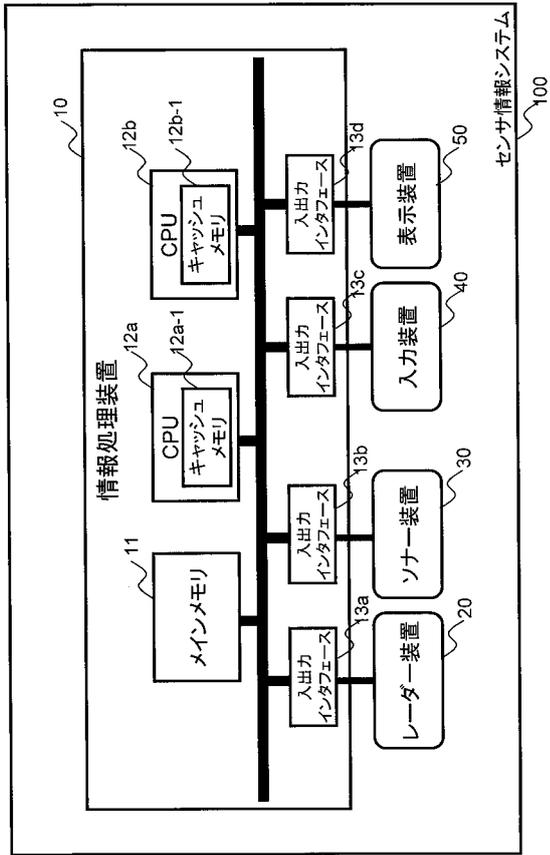
3a-code4・・・ループコード

3a-code5・・・後処理コード

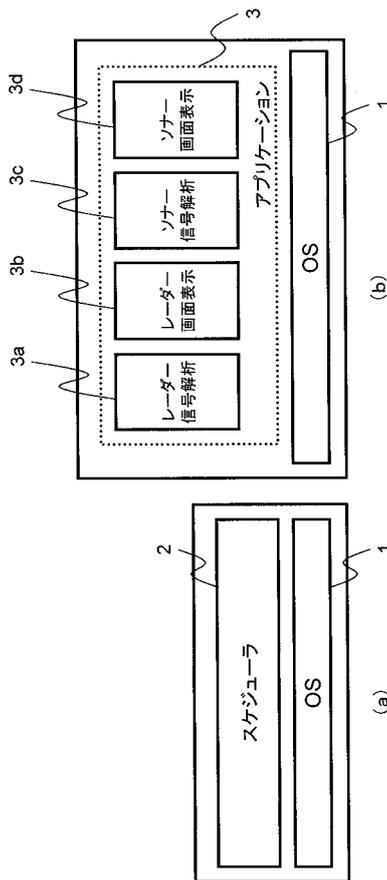
50

3 b . . . レーダー画面表示用アプリケーション	
3 b - c o d e . . . レーダー画面表示用アプリケーションのコード	
3 b - c o d e 1 . . . 前処理コード	
3 b - c o d e 2 . . . ピクセル輝度の算出コード	
3 b - c o d e 3 . . . 進捗状況算出コード	
3 b - c o d e 4 . . . ループコード	
3 b - c o d e 5 . . . 後処理コード	
3 b - c o d e 1 0 . . . レーダー画面表示用アプリケーションのコード	
3 b - c o d e 1 1 . . . 第 1 の処理コード	
3 b - c o d e 1 2 . . . 第 2 の処理コード	10
3 b - c o d e 1 3 . . . 第 3 の処理コード	
3 b - c o d e 1 4 . . . 第 4 の処理コード	
3 b - c o d e 1 5 ~ 3 b - c o d e 1 7 . . . 進捗度指示コード	
3 b - 1 . . . 実行コード	
3 b - 2 . . . 進捗管理コード	
3 c . . . ソナー信号解析用アプリケーション	
3 d . . . ソナー画面表示用アプリケーション	
1 0 . . . 情報処理装置	
1 1 . . . メインメモリ	
1 2 A、1 2 B . . . C P U	20
1 2 A - 1、1 2 B - 1 . . . キャッシュメモリ	
1 3 A ~ 1 3 D . . . 入出力インタフェース	
2 0 . . . レーダー装置	
3 0 . . . ソナー装置	
4 0 . . . 入力装置	
5 0 . . . 表示装置	
1 0 0 . . . センサ情報システム	

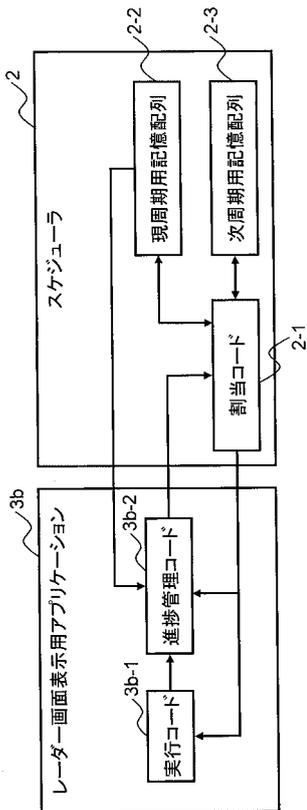
【図 1】



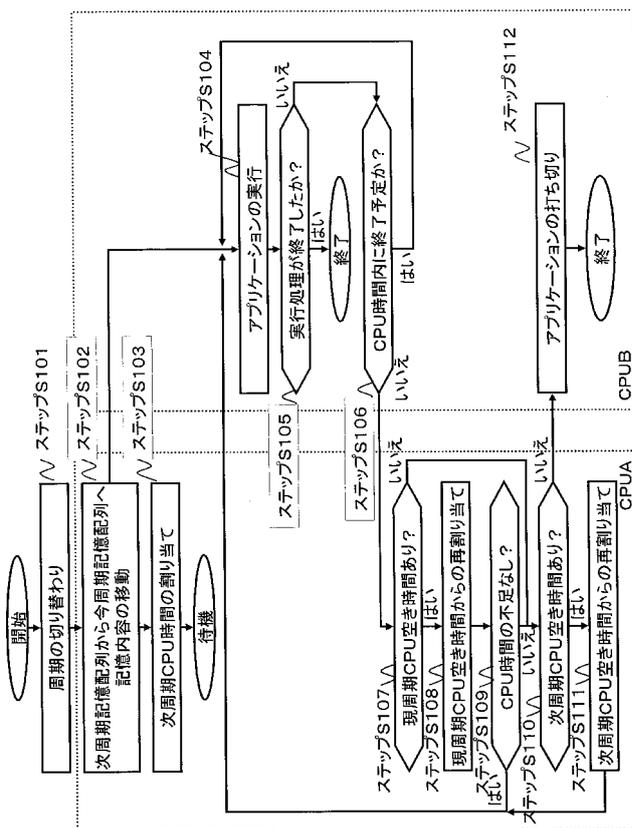
【図 2】



【図 3】



【図 4】



【 図 5 】

アプリケーション	CPU時間 [ms]
レーダー信号解析アプリケーション	400
ソナー信号解析アプリケーション	300
レーダー画面表示アプリケーション	50
ソナー画面表示アプリケーション	50
未割当CPU時間	200

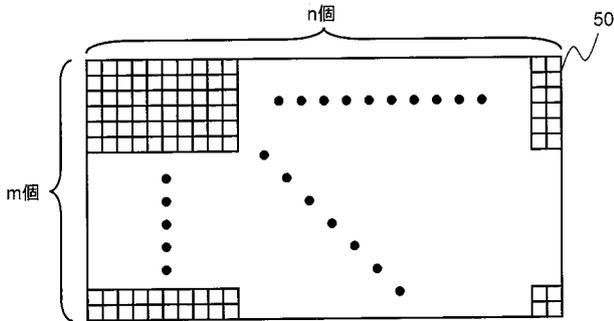
【 図 6 】

アプリケーション	CPU時間 [ms]
レーダー信号解析アプリケーション	400
ソナー信号解析アプリケーション	350
レーダー画面表示アプリケーション	50
ソナー画面表示アプリケーション	50
未割当CPU時間	150

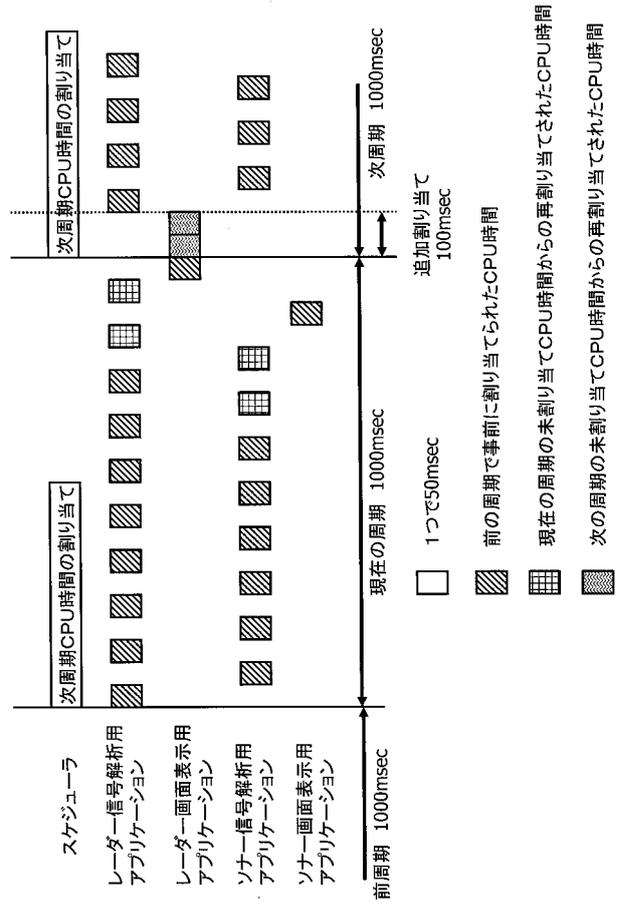
【 図 8 】

アプリケーション	実行に要したCPU時間 [ms]
レーダー信号解析アプリケーション	500
ソナー信号解析アプリケーション	400
レーダー画面表示アプリケーション	150
ソナー画面表示アプリケーション	50

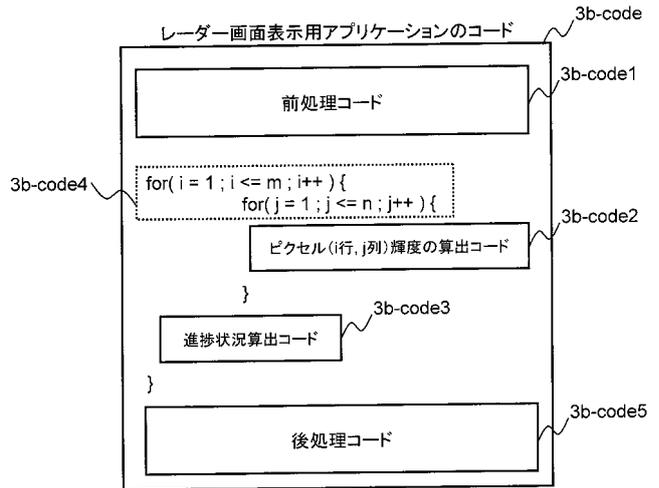
【 図 9 】



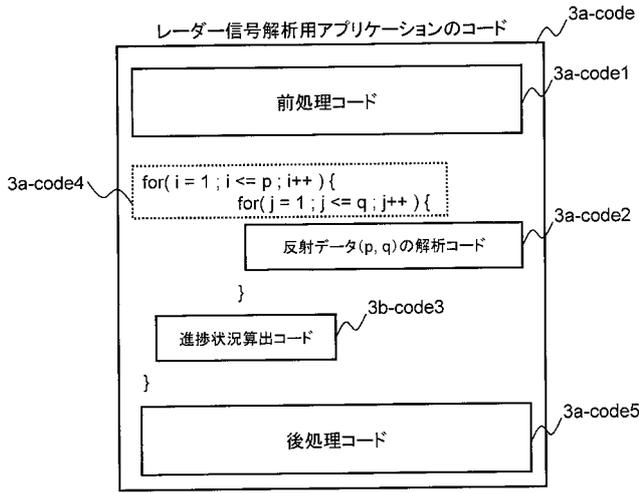
【 図 7 】



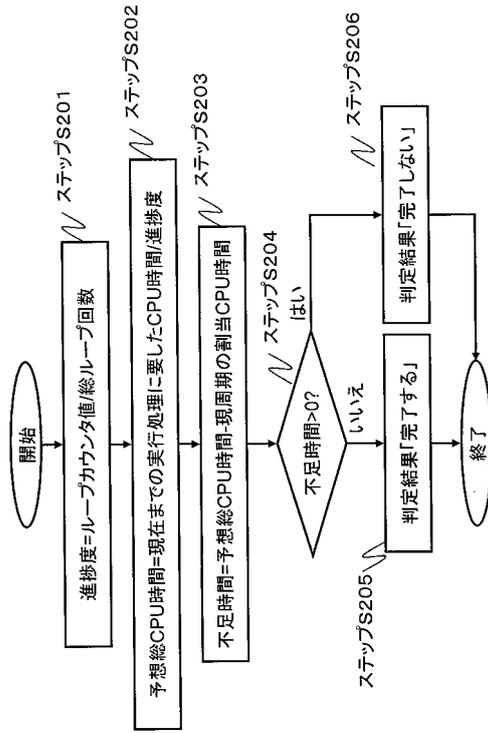
【 図 10 】



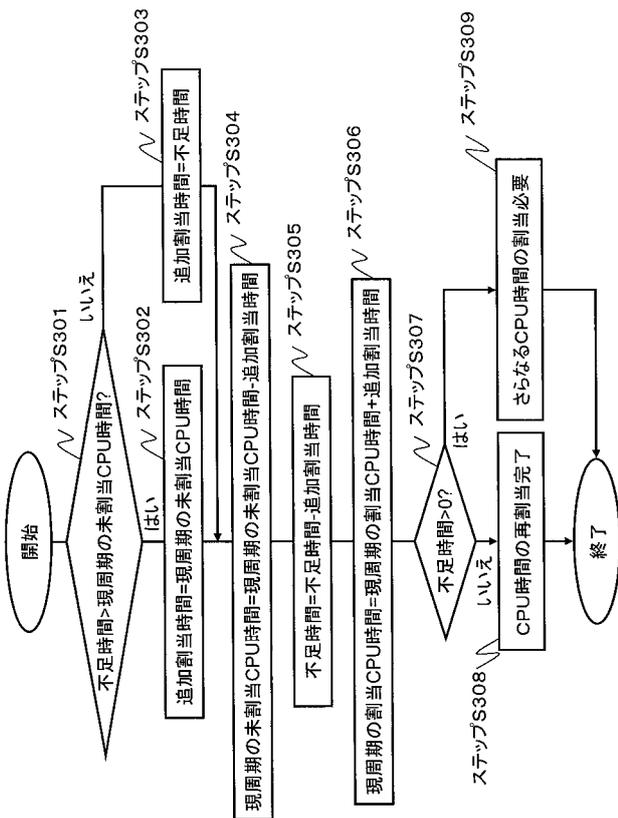
【 図 1 1 】



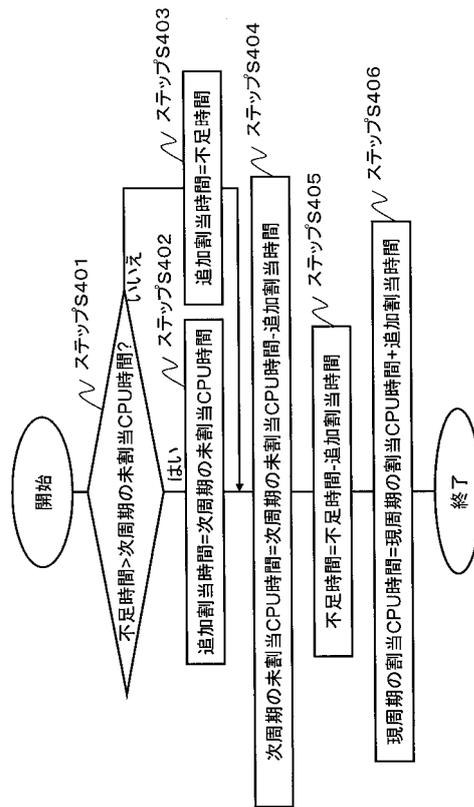
【 図 1 2 】



【 図 1 3 】



【 図 1 4 】



【 図 15 】

