(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4194258号
(P4194258)

(45)発行日　平成20年12月10日(2008.12.10)　(24)登録日　平成20年10月3日(2008.10.3)

(51)Int.Cl.		F I		
G06F 17/30	(2006.01)	G06F	17/30	150F
G06F 12/00	(2006.01)	G06F	17/30	407
		G06F	17/30	413
		G06F	12/00	520A

請求項の数 1　(全 15 頁)

(21)出願番号　特願2001-224463(P2001-224463)
(22)出願日　平成13年7月25日(2001.7.25)
(65)公開番号　特開2003-36258(P2003-36258A)
(43)公開日　平成15年2月7日(2003.2.7)
審査請求日　平成17年2月14日(2005.2.14)

(73)特許権者　000001889
三洋電機株式会社
大阪府守口市京阪本通２丁目５番５号
(74)代理人　100064746
弁理士　深見　久郎
(74)代理人　100085132
弁理士　森田　俊雄
(74)代理人　100091409
弁理士　伊藤　英彦
(74)代理人　100096781
弁理士　堀井　豊
(72)発明者　岡田　誠
大阪府守口市京阪本通２丁目５番５号　三洋電機株式会社内

最終頁に続く

(54)【発明の名称】データ表示装置

(57)【特許請求の範囲】
【請求項１】
外部からデータを受信するための受信手段と、
前記受信手段が受信した複数の可変長レコードで構成されるデータ列から、ユーザの指示に従って任意のレコードデータを取得するデータ取得処理手段と、
取得したレコードデータの先頭位置のアドレス情報を格納するアドレス管理リストと、
取得したレコードデータを表示するための表示手段とを備え、
前記ユーザが指示したレコードデータのアドレス情報が前記アドレス管理リストに格納されていないとき、前記データ列の各可変長レコードの先頭位置にあるレコード長情報に基づいて、取得対象のレコードデータのアドレス情報を取得すると共に、該レコードデータまでの連続するすべてのレコード先頭位置のアドレス情報を抽出して、前記アドレス管理リストに保持させるアドレス情報抽出手段を備える、データ表示装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、外部から受信した可変長レコードで構成されるデータ列を処理するためのデータ処理システムおよびそれを用いるデータ表示装置の構成に関する。
【０００２】
【従来の技術】
近年、テレビ放送等においても、放送される映像情報および音声情報と関連する文字デー

タが放送局から送信され、受信機側において映像情報および音声情報と並行して、あるいは文字情報のみがディスプレイ上に表示されるという放送形態がとられる場合がある。

【０００３】

たとえば、テレビ放送において、天気予報の情報をデータ放送により伝送する場合、このような天気予報に関する情報は、連続したバイナリデータであって、その内容自体は表形式であるデータとして伝送されることが多い。受信機側では、ユーザの指示に従って、このようにして放送されたバイナリデータから所望のデータの取得を行なって、ユーザが求めた地域や時刻等の天気予報情報をディスプレイ上に表示することになる。

【０００４】

ところが、このような天気予報の情報等は、複数の可変長なレコードデータがバイナリデータとして順次伝送される。このようなバイナリデータから複数の可変長レコードデータの取得を行なう際には、受信機においては、先頭から１行ずつスキャンを行なって所望のレコード位置を探索する必要がある。

【０００５】

【発明が解決しようとする課題】

しかしながら、上述したような従来の技術を用いた場合、所望のレコード位置が受信したバイナリデータ中の後方にあると、先頭から１行ずつスキャンすることにより、レコードデータ取得までに膨大な時間を要するという問題点があった。

【０００６】

この発明は、上記のような問題点を解決するためになされたものであって、その目的は、外部から順次受信した複数の可変長レコードで構成されるバイナリデータから、ユーザの指示に従って任意のレコードデータを取得する際に、レコードデータ取得までの時間を短縮することが可能なデータ処理システムおよびこれを用いたデータ表示装置を提供することである。

【０００７】

【課題を解決するための手段】

本発明のデータ表示装置は、外部からデータを受信するための受信手段と、前記受信手段が受信した複数の可変長レコードで構成されるデータ列から、ユーザの指示に従って任意のレコードデータを取得するデータ取得処理手段と、取得したレコードデータの先頭位置のアドレス情報を格納するアドレス管理リストと、取得したレコードデータを表示するための表示手段とを備え、前記ユーザが指示したレコードデータのアドレス情報が前記アドレス管理リストに格納されていないとき、前記データ列の各可変長レコードの先頭位置にあるレコード長情報に基づいて、取得対象のレコードデータのアドレス情報を取得すると共に、該レコードデータまでの連続するすべてのレコード先頭位置のアドレス情報を抽出して、前記アドレス管理リストに保持させるアドレス情報抽出手段を備える。

【００２１】

【発明の実施の形態】

以下、図面を参照して本発明の実施の形態について説明する。

【００２２】

図１は、本発明に係るデータ処理システムを備えるデータ表示装置１０００の構成の例を示した概念図である。

【００２３】

図１においては、一例として、上述したようなデータ放送を受信することが可能な受信機１０１において、本発明に係るデータ処理システムが使用され、データ表示装置１０００が実現されている場合の構成を示している。

【００２４】

図１を参照して、受信機１０１は、アンテナ２を介して、映像および音声ストリームと多重されたデータ放送コンテンツを受信し、データ放送コンテンツに記述された情報に従って、ディスプレイ１０２およびスピーカ１０３を介して、映像、音声、文字、図形などを表示する。

【００２５】
ユーザは、リモコン１０４を用いて、データ放送サービスを選択し、所望の情報を得ることができる。
【００２６】
図１に示した例においては、ユーザが、各地の明日の天気に対する天気予報の情報を得るにあたり、東京地方をその天気予報の対象として選択した状態を示している。したがって、ディスプレイ１０２上において、「東京」が選択され、画面上で四角形の枠で囲まれている。
【００２７】
図２は、図１に示した受信機１０１の内部構成を説明するための概略ブロック図である。
【００２８】
図２を参照して、受信機１０１において、アンテナ２により受信されたＲＦ信号は、受信部２０１において、選局されて復調処理およびデコード処理等を行なわれ、分離部２０２に与えられる。分離部２０２は、受信部２０１からの信号を、映像音声情報に対する信号と、データ放送コンテンツとに分離する。分離部２０２から、映像および音声ストリームは、ＡＶ処理部２０３へ送られ、データ放送コンテンツはコンテンツ用メモリ２０４に送られる。ＡＶ処理部２０３は、映像および音声ストリームを受けて、対応するカラー画像を表示可能な信号に変換する。
【００２９】
一方、コンテンツ用メモリ２０４に蓄積されたデータ放送コンテンツに対して、ＣＰＵ２０７が処理を行なうことにより、文字や図形等をデータ放送コンテンツによって指定されたディスプレイ１０２の画面上の指定位置へ表示するための信号が、オンスクリーンディスプレイ処理部（以下、「ＯＳＤ処理部」と呼ぶ）２０５に与えられ、さらにＯＳＤ処理部２０５によって、映像データと文字や図形等が合成された後、ディスプレイ１０２上に表示される。
【００３０】
ユーザからの入力信号は、リモコン１０４からユーザインターフェイス２０８を介してＣＰＵ２０７に伝えられる。ＣＰＵ２０７では、データ放送コンテンツとして受信されたプログラム内に記述された入力情報に対する処理を実行して、メディアの表示制御や受信機の動作制御を行なう。
【００３１】
次に、データ放送コンテンツ内にバイナリデータが含まれている場合の処理手順について説明する。
【００３２】
図３および図４は、データ放送コンテンツに基づいた表示例を示す概念図である。
【００３３】
まず、図３に示すように、天気予報情報の画面において、地域選択画面が表示される。図３に示した例では、たとえば、地域として札幌、東京、大阪および福岡の中から、ユーザが天気予報を見たいと思う地域を選択する構成となっているものとする。図３では、ユーザからのリモコン１０４を介しての指示により、東京が選択されている状態を示す。
【００３４】
図４は、図３のようにしてユーザによって選択された東京地方の明日の天気の情報が表示されている状態を示す。
【００３５】
図４に示されている表示のうち１行目の「の明日の天気」、２行目の「天気」、３行目の「最高気温」および「℃」、４行目の「最低気温」、「℃」および「戻る」との表示は、ユーザがどの地域を選択するかにかかわらず固定された表示が行なわれる領域である。
【００３６】
一方、１行目の「東京」、２行目の「晴れのち曇り」、３行目の「２２」、４行目の「１６」等の表示は、ユーザがどの地域を選択するかに応じてその表示内容が変更される部分

である。

【００３７】

後に説明するように、このようなユーザの選択に応じてその表示内容が変更される部分については、放送局からは、表形式のデータとして、受信機１０１に対して送信されている。

【００３８】

すなわち、データ放送コンテンツに、選択した地域の天気情報が、表形式データを表現するバイナリデータとして含まれており、図４の画面を表示する際に、このバイナリデータから天気情報データの検索が行なわれることになる。

【００３９】

図５は、図３および図４で示したような天気予報提供サービスを実現するためのデータ放送コンテンツのうち、ユーザからの指示に従って、バイナリデータから天気情報データを取得するための検索処理に対応するプログラムリストを示す。

【００４０】

ここで、図５に示したようなプログラムは、ＸＭＬ（eXtensible Markup Language）に基づいて、放送用に仕様変更を行なった言語である「ＢＭＬ（Broadcast Markup Language）」により記述されている。

【００４１】

図５を参照して、まず、１行目の「var bt」によって、変数btの宣言が行なわれる。

【００４２】

続いて、放送局から受信した表形式のデータである”tenki.bt”というバイナリデータを変数btに読込むための関数bt__new（）の定義が行なわれる。

【００４３】

このようにして、変数btに読込まれた天気情報に対応したバイナリデータから、ユーザによって指定される地域を表わす変数locationを引き数とする関数getTenkiData（location）により、選択された地域に応じて、天気、最高気温、最低気温の値の検索および取得が行なわれる。取得された天気、最高気温、最低気温は、関数dispData（）により画面に表示される。

【００４４】

ここで、まず関数bt__new（）の定義式中で用いられる関数newBinaryTable（X,Y）は、第１番目の引き数Ｘによって指定されるバイナリデータからのデータを読出すことを示す。さらに、この関数の２番目の引き数Ｙの“１，Ｓ：１Ｖ，Ｓ：１Ｖ，Ｉ：１Ｂ，Ｉ：１Ｂ”は、Ｘで示されるバイナリデータに格納されているレコードデータのフォーマットを示している。

【００４５】

すなわち、１番最初の「１」は、バイナリデータの可変長レコードの先頭の１バイト目には、各可変長レコードのレコード長が格納されていることを示す。続く「Ｓ：１Ｖ」は、その各可変長レコードの先頭には、１バイト目にそのデータ長が記述されている第１の可変長文字列データがあることを示す。さらに、この第１の可変長文字列データに続いて、「Ｓ：１Ｖ」で表される第２の可変長文字列データが存在する。この第２の可変長文字列データに続いて、各可変長レコードには、「Ｉ：１Ｂ」で表される２つのデータが存在する。ここで、「Ｉ：１Ｂ」は、１バイト長の固定長（「１Ｂ」で表される）の整数値データ（「Ｉ」で表される）がレコード内に格納されていることを示している。

【００４６】

一方、５行目からは、ユーザから指定された地域locationに基づいて、天気情報のデータをバイナリデータから獲得するための関数ｇetTenkiData（）の定義が行なわれる。この定義中において、まず、「var tenki, maxTemp,minTemp」は、それぞれ、天気、最高気温、最低気温を表す変数tenki, maxTemp,minTempを宣言するものである。また、関数bt.toString（X,Y）は、バイナリデータのテーブル中から、文字列データを取得するための関数であり、１番目の引き数Ｘはバイナリテーブル内のレコード位置のインデックスを表わ

し、２番目の引き数Ｙはレコード内のデータ位置のインデックスを示している。

【００４７】

関数toNumber（X,Y）も、バイナリデータ中から、整数値データを取得するための関数であり、引き数Ｘは、バイナリテーブル内のレコード位置のインデックスであり、引き数Ｙはレコード内のデータ位置のインデックスを示している。

【００４８】

図６は、天気情報を表わす可変長レコード形式のバイナリデータtenki.btのダンプ内容を示す概念図である。

【００４９】

図６中では、バイナリデータのダンプ内容に対応した文字および数値データを、ダンプデータの下に対比のために示している。

【００５０】

図６において、四角で囲んだバイトは各レコードの長さを示している。すなわち、図５で説明した関数newBinaryTable（）関数中の２番目の引き数において、先頭の「１」が示していたように、各レコードの１バイト目には、当該レコードのレコード長を示すデータが存在している。

【００５１】

たとえば、１番最初の札幌の天気情報を示すレコードにおいては、レコード長として「０Ｃ」が指定されているために、このレコードは、このレコード長を示すデータの後に、１２バイト分のデータが存在するレコードであることが示されている。すなわち、各レコードのデータは、四角で囲んだバイトの次バイトから始まり、次の四角で囲んだバイトの手前までとなる。

【００５２】

続いて、この「札幌」の天気情報が格納されているデコードについて見ると、関数newBinaryTable（）の２番目の引き数に示されているとおり、そのフォーマットとして次のデータは「Ｓ：１Ｖ」である。このため、まず先頭の「０４」がこれに続く可変長文字列データのバイト数を示す。つまり、「０４」に続く「８Ｅ」～「７９」までの４バイトのデータが、「札幌」という文字列に対応するデータである。

【００５３】

さらに、関数newBinaryTable（）関数の第２引き数のデータフォーマットに従うと、これに続いて、「Ｓ：１Ｖ」のデータ、すなわち先頭バイトにデータ長が示されている可変長文字列データが格納されていることになる。すなわち、「札幌」のデータに続く「０４」は、これに続く文字列データが４バイトであることを示す。したがって、「９０」～「ＥＡ」が、「晴れ」との文字列に対応している。

【００５４】

さらに、関数newBinaryTable（）の第２引き数のデータフォーマットによれば、この「晴れ」との文字列データの後に、「Ｉ：１Ｂ」で表わされるデータが２つ続いていることになる。上述のとおり、「Ｉ：１Ｂ」は、整数値のデータ（「Ｉ」により表現される）が、固定長の１バイトのデータであることを示している。

【００５５】

したがって、「晴れ」のデータに続く１６進数の「１２」という１バイトのデータが、１０進表示で最高気温の「１８」を示し、これに続く１バイトの「０Ａ」との１６進データが、１０進表示では「１０」という最低気温を表現していることになる。

【００５６】

以上で、第１番目のレコードの「札幌」の天気情報に対するデータが終了し、引続いて、「東京」に対応するレコードが存在している。「東京」に対応するレコードは、「１４」との１バイトのデータの直後から始まっており、すなわち、２０バイトの長さを有していることがわかる。

【００５７】

以下同様にして、「大阪」および「福岡」の天気情報が可変長レコードにより構成される

一連のバイナリデータとして、放送局から送信されてくることになる。

【００５８】

以上のような連続したバイナリデータで表わされる可変長レコードの天気情報が放送により伝送された後に、ユーザが、たとえば「東京」の天気情報を知りたい場合、リモコン１０４から、受信機１０１に対して「東京」の選択を指示する。これに応じて、ＣＰＵ２０７では、コンテンツ用メモリ２０４中に格納されている図５に示したような検索用のプログラムに基づいて、やはりコンテンツ用メモリ２０４中に格納されている放送されたバイナリデータから東京の天気情報を獲得する。

【００５９】

より具体的に言えば、図５に示したプログラムにおいて、関数bt.toString（1,1）の処理によって、１番目のレコード内の１番目のデータ位置からのデータの取得が行なわれる。

【００６０】

このとき、従来の方法では、まず第０番目の札幌の天気の情報を含むレコードについては、その先頭の「０Ｃ」で表わされるレコードの長さ、すなわち１２バイト分だけデータを先頭からスキャンすることにより、続く１番目のレコードの東京の天気情報が格納されたレコードの長さが２０バイト（０ｘ１４）であることを検知して、この２０バイトのレコードのデータのうち、１番目のデータから天気情報として“晴れ後曇り”の文字列を取出す。同様にして、関数bt.toNumber（1,2）、関数bt.toNumber（1,3）の処理により、東京の最高気温、最低気温をバイナリデータから取出す。

【００６１】

大阪や福岡の天気情報を取出す場合も、一連のバイナリデータから、札幌、東京のレコードデータを先頭からスキャンしていくことにより、大阪のデータさらには福岡のデータの抽出を行なう。

【００６２】

このように、従来の方法では、所望のレコードがバイナリデータの後方に存在する場合でも、先頭から順次レコードをスキャンする必要があり、大きなサイズのバイナリデータからレコードデータを取出すとき処理に時間を要してしまうことになる。

【００６３】

そこで、本発明に係る受信機１０１においては、図２に示すようにアドレス管理用メモリ２０６を設け、たとえば、図５の関数bt__new（）がＣＰＵ２０７から呼出された際などに、バイナリデータの各レコードの先頭位置のアドレスをアドレス管理用メモリ２０６に対してリストとして保持しておく処理をさらに追加する。

【００６４】

このようなリストを参照して、各天気情報を抽出することとすれば、ユーザがリモコン１０４を用いて地域選択を行なう２回目以降の処理においては、検索時間を短縮することが可能となる。

【００６５】

図７は、アドレス管理用メモリ２０６に格納される、アドレス管理用リストの例を示す図である。図７に示すとおり、０番目のレコードであるrecord［０］の先頭アドレスがアドレス“００００００００１”であることが格納されている。他のレコードについても同様である。

【００６６】

つまり、図７に示すとおり、アドレス管理用メモリ２０６は、各レコードのインデクスの０～３の値と対応づけて、各レコードデータの先頭位置である、図６において四角で囲んだバイトの次のバイトのアドレスを保持している。このアドレス管理用メモリ２０６に格納されるアドレス管理用リストは、固定長レコードのリストデータである。

【００６７】

［アドレス管理用リストの作成タイミング］

このようなアドレス管理用リストをいつの時点で、また、どのレコードのアドレスまでアドレス管理用メモリ２０６に作成して格納しておくかについては、以下に説明するような

いくつかの方式が考えられる。

【００６８】

［作成タイミング１］

図８は、このようなアドレス管理用メモリ２０６に、レコードの先頭位置のアドレスを格納する第１番目の処理を示すフローチャートである。

【００６９】

なお、以下に説明するような関数newBinaryTable（）によるバイナリデータの読出しの処理とアドレス管理リストの作成処理は、このバイナリデータの受信完了時に行なっておき、予め受信されたすべてのレコードデータの先頭位置のアドレス情報をアドレス管理リストとして作成しておくことが可能である。

【００７０】

すなわち、図８を参照して、ＣＰＵ２０７が、図５に示したプログラムを実行するに当たり、関数newBinaryTable（）が呼出される（ステップＳ１００）。

【００７１】

バイナリデータが格納されたファイルを、コンテンツ用メモリ２０４から読出処理が行なわれる（ステップＳ１０２）。

【００７２】

このとき、ファイル読込に失敗した場合は（ステップＳ１０４）、エラー終了のため、メインルーチンにはエラーが生じたことを示す“false”を返す処理を行なう（ステップＳ１０６）。

【００７３】

一方、ステップＳ１０４において、ファイル読込が正常に行なわれた場合、続いて、読込まれたファイルに対するスキャン動作が終了したか否かの判定が行なわれる（ステップＳ１１０）。

【００７４】

スキャン動作が終了していない場合、続いて、変数tempにスキャン対象となるレコードのレコード長を格納する（ステップＳ１１２）。

【００７５】

次に、レコード長が格納されたバイトをスキャンし（ステップＳ１１４）、レコードデータが格納されているアドレスへ移り、そのアドレスをアドレス管理用リストに格納する（ステップＳ１１６）。

【００７６】

その後、変数tempで示されているレコードデータのバイト長分だけさらにバイナリデータのスキャンを行なって（ステップＳ１１８）、処理はステップＳ１１０に復帰する。

【００７７】

次のレコードに対しても同様に、変数tempにレコード長のデータを格納し（ステップＳ１１２）、レコード長のバイトをスキャンして（ステップＳ１１４）、現在のアドレスすなわちレコード内のデータの先頭アドレスをアドレス管理用リストに格納する（ステップＳ１１６）。続いて、変数tempバイト分だけスキャンを行なって（ステップＳ１１８）、処理が再びステップＳ１１０に復帰する。

【００７８】

以下同様にして、バイナリデータに含まれるすべてのレコードの先頭位置のアドレスの格納が完了した場合に、関数newBinaryTable（）の処理を終了する。

【００７９】

このような処理が終了すると、ＣＰＵ２０７は、ユーザがリモコン１０４により天気予報の対象地域の選択を行なうのを待機することになる。

【００８０】

図９は、図８に示したような処理を行なって、アドレス管理用メモリ２０６中にアドレス管理用リストが格納された後の、レコードデータの取得処理を示すフローチャートである。

【００８１】
すなわち、ユーザがリモコン１０４により天気予報の対象地域の選択を行なった後に、図５に示したプログラムの処理において、関数toString（）や、関数toNumber（）に処理が行なわれる際の処理のフローを示している。
【００８２】
図９を参照して、関数toString（row，column）や関数toNumber（row，Column）が呼出されると（ステップＳ２００）、これら引数rowの大きさとアドレス管理用リストの格納レコード数との比較が行なわれる（ステップＳ２０２）。
【００８３】
引数rowがアドレス管理用リストの格納レコード数以上の場合、関数toString（）や関数toNumber（）は、引数値がエラーを示す値“false”を返す（ステップＳ２０４）。
【００８４】
一方、ステップＳ２０２において、引数rowがアドレス管理用リストの格納レコード数よりも少ない場合、続いて、アドレス管理用リストのインデックスの値が引数rowである位置に対応するアドレス値の取得が行なわれる（ステップＳ２１０）。
【００８５】
続いて、アドレス管理用リストから取得したアドレスをもとに、バイナリデータからレコードデータの読込が行なわれ（ステップＳ２１２）、読込まれたレコードデータから、図５のフォーマット指定に基づいて、引数columnで指定される位置のデータの取得が行なわれる（ステップＳ２１４）。
【００８６】
以上の処理が終了した段階で処理が終了し、正常終了であることを示す返り値“true”がメインルーチンに返される（ステップＳ２１６）。
【００８７】
このような処理により、ユーザが天気情報を画面上で見ようとした場合に、各地の天気情報を格納したバイナリデータの先頭から、所望の地域に対応したレコードまでスキャンしてデータの読出しを行なうという必要がなく、当該所望の地域に対応したレコードをそのアドレスにより直接参照できるので、天気情報の表示までの所要時間を短縮することができる。
【００８８】
［作成タイミング２］
以上説明したような作成タイミング１の処理では、アドレス管理用リストへバイナリデータの各レコードの先頭位置のアドレスを格納する処理を、関数newBinaryTable（）がメインルーチンから呼出された時点で行なっていた。
【００８９】
これに対して、メインルーチンにおいて、関数newBinaryTable（）が呼出された時点ではなく、図５に示したプログラム処理内において、関数toString（）や関数toNumber（）が実行された際に、この関数の引数で指定されたレコードまでについて、その都度アドレス管理用リストを作成して、アドレス管理用メモリ２０６に格納するという処理を行なうことも可能である。
【００９０】
図１０は、このようにレコードデータの取得処理が行なわれるごとに逐次的にアドレス管理用リストを作成する処理を示すフローチャートである。
【００９１】
図１０に示す処理では、関数toString（）や関数toNumber（）の処理で初めて読込が行なわれるファイルの場合、まだアドレス管理用リストにはレコード先頭位置を示すアドレスが格納されていない状態である。このため、所望のレコードまでのデータを１レコードずつスキャンし、その際、各レコードの先頭位置を逐次アドレス管理用リストに格納するという処理を行なう。
【００９２】

すなわち、図１０を参照して、関数toString（row，column）や関数toNumber（row，column）が呼出されると（ステップＳ３００）、まず、読出対象となっているファイルが、初めてデータの読込を行なうファイルであるか否かの判定が行なわれる（ステップＳ３０２）。初めて読込むファイルでない場合は、後に説明する「処理２」に処理が移行する（ステップＳ３０４）。

【００９３】

一方、初めて読込むファイルである場合（ステップＳ３０２）、変数indexに０が代入され、変数maxIndexに引数rowが代入される（ステップＳ３１０）。

【００９４】

続いて、バイナリデータのスキャンが開始され、先頭のレコードに対応するレコード長が変数tempに代入され、インデックスの値が１だけインクリメントされる（ステップＳ３１２）。

【００９５】

次に、レコード長を表すバイト（たとえば、図６の説明では１バイト分）についてはバイナリデータをスキャンし（ステップＳ３１４）、レコード長を表わすバイトの次のアドレスをアドレス管理用リストに当該レコードの先頭アドレスとして格納する（ステップＳ３１６）。

【００９６】

次に、変数indexの値と、引数rowの値とが比較される。変数indexの値が引数row以下である場合は、処理はステップＳ３２０に移行して、変数tempで表わされるバイト分だけバイナリデータをスキャンした上で、再び処理がステップＳ３１２に復帰する。ステップＳ３１２では、次のレコードに対応するレコード長を表すバイトから次のレコードのレコード長のデータを読取り、これを変数tempに代入し、変数indexの値を１だけインクリメントする。

【００９７】

以下、同様の処理がステップＳ３１８まで繰返される。

一方、ステップＳ３１８において、変数indexが引数rowよりも大きい場合は、現在の位置が引数rowによって指定されたレコードであるため、現在の位置からレコードデータの読込が行なわれる（ステップＳ３２２）。

【００９８】

読込まれたレコードデータから、続いて、引数columnの位置のデータの取得が行なわれ（ステップＳ３２４）、処理が終了して、正常終了であることを示す値“true”がメインルーチンに返される（ステップＳ３２６）。

【００９９】

図１１は、図１０において示した「処理２」、すなわち、関数toString（row，column）や関数toNumber（row，column）が呼出された際に、読出の対象となるファイルが初めて読込むファイルではなく、２回目以降の読出処理である場合の処理を示すフローチャートである。

【０１００】

すなわち、「処理２」が開始されると（ステップＳ３０４）、まず、所望のレコードのインデックスrowが、アドレス管理用リストに格納されたレコード数を示す値maxIndexよりも小さいか否かの判定が行なわれる（ステップＳ４０２）。

【０１０１】

インデックスrowがレコード数maxIndex以下である場合、続いて、アドレス管理用リストのインデックスが引数rowである位置のアドレスをアドレス管理用リストから取得する（ステップＳ４１０）。

【０１０２】

取得したアドレスをもとに、バイナリデータからレコードデータの読込が行なわれ（ステップＳ４１２）、レコードデータから引数columnの位置のデータの取得が行なわれる（ステップＳ４１４）。

【０１０３】
以上により、処理が終了し、正常終了であることを示す値“true”がメインルーチンに返される（ステップＳ４１６）。
【０１０４】
一方、ステップＳ４０２において、所望のレコードのインデックスrowがアドレス管理用リストに格納されたレコード数maxIndexよりも大きい場合、アドレス管理用リストのインデックスがmaxIndexである位置のアドレスを、すなわち、アドレス管理用リストに格納された最後尾のレコードの先頭位置のアドレスを取得する（ステップＳ４２０）。
【０１０５】
この取得されたアドレスからレコード長のバイト分だけアドレスを引き戻した後（ステップＳ４２２）、所望のレコードまでのデータを１レコードずつスキャンする。すなわち、まず、変数indexに変数maxIndexの値が代入され、変数maxIndexには引数rowの値が代入される（ステップＳ４２４）。
【０１０６】
続いて、現状のレコードの先頭位置の１バイトのデータから、レコード長に相当するデータを取得して、変数tempに代入し、変数indexの値を１だけインクリメントする（ステップＳ４２６）。
【０１０７】
次に、レコード長のバイト＋変数tempのバイト分のデータをスキャンし（ステップＳ４２８）、処理は、「処理３」に移行する（ステップＳ４３０）。
【０１０８】
図１２は、図１１に示した「処理３」を示すフローチャートである。
図１２を参照して、「処理３」にＣＰＵ２０７の処理が移行すると（ステップＳ４３０）、続いて、読出対象となっているバイナリデータの最終データまで処理が終了しているか否かの判定が行なわれる（ステップＳ４３２）。
【０１０９】
データが最終データまで到達している場合は、エラー終了として、メインルーチンに値“false”が返される（ステップＳ４３４）。
【０１１０】
一方、データが最終位置までスキャンされていない場合は（ステップＳ４３２）、変数tempに現在のレコードの先頭のデータからレコード長が読出され代入され、かつ変数indexの値が１だけインクリメントされる（ステップＳ４４０）。
【０１１１】
次に、レコード長を表わすバイトをスキャンし（ステップＳ４４２）、現在のレコードのデータ内容を表わす領域の先頭のアドレスをアドレス管理用リストに格納する（ステップＳ４４４）。
【０１１２】
続いて、変数indexの値と引数rowとの値が比較され（ステップＳ４４６）、変数indexが引数row以下である場合は、変数tempバイト分だけデータのスキャンが行なわれて（ステップＳ４４８）、処理は、ステップＳ４３２に復帰する。
【０１１３】
一方、ステップＳ４４６において、変数indexの値が引数rowよりも大きい場合は、現在の位置からレコードデータの読込が行なわれ（ステップＳ４５０）、レコードデータから引数columnの位置のデータの取得が行なわれる（ステップＳ４５２）。
【０１１４】
以上により処理が終了して、正常終了であることを示す値“true”がメインルーチンに返される（ステップＳ４５４）。
【０１１５】
このような処理を行なうことで、関数toString（）や関数toNumber（）がメインルーチンから呼出されるごとに、逐次各レコードの先頭位置をアドレス管理用リストに格納するの

で、一括して各レコードの先頭位置を探索してアドレス管理用リストに格納する場合に比べて、最初のデータの表示までの時間を短縮することが可能となる。

【０１１６】

なお、以上の説明では、可変長レコードの先頭の直前に１バイトのレコード長を示すデータが配置されるものとして説明したが、レコード長を示すデータの大きさは、さらに大きなものでも、あるいはより小さいものでもよく、レコード長を示すデータの位置も、可変長レコードの先頭の直前に限定されるものではない。たとえば、可変長レコードの先頭とレコード長を示すデータと間に所定の大きさの管理データ等が挿入されていてもよい。

【０１１７】

今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【０１１８】

【発明の効果】

以上説明したとおり、本発明に係るデータ処理システムおよびそれを用いたデータ表示装置によれば、データ列に対応したアドレス管理リストを予め作成することで、少なくとも一度取得したレコードのインデックスよりも小さいインデックスのレコードデータを取得する場合、高速にレコードデータを取得できる。

【０１１９】

また、一度取得したレコードのインデックスよりも大きいインデックスのレコードデータを取得する場合も、一度取得したレコードのアドレス位置からスキャンし当該レコードデータを取得することにより、データ列の先頭から検索する方法に比べて、高速にレコードデータを取得できる。

【０１２０】

すなわち、アドレスを管理するためのリストの追加のみで、可変長のレコードから構成されるデータ列から所望のレコードデータを取得する際の所要時間を短縮することが可能となる。

【図面の簡単な説明】

【図１】　本発明に係るデータ処理システムを備えるデータ表示装置１０００の構成の例を示した概念図である。

【図２】　図１に示した受信機１０１の内部構成を説明するための概略ブロック図である。

【図３】　データ放送コンテンツに基づいた表示例を示す第１の概念図である。

【図４】　データ放送コンテンツに基づいた表示例を示す第２の概念図である。

【図５】　ユーザからの指示に従って、バイナリデータから天気情報データを取得するための検索処理に対応するプログラムリストを示す。

【図６】　天気情報を表わす可変長レコード形式のバイナリデータtenki.btのダンプ内容を示す概念図である。

【図７】　アドレス管理用メモリ２０６に格納される、アドレス管理用リストの例を示す図である。

【図８】　アドレス管理用メモリ２０６に、レコードの先頭位置のアドレスを格納する第１番目の処理を示すフローチャートである。

【図９】　アドレス管理用メモリ２０６中にアドレス管理用リストが格納された後の、レコードデータの取得処理を示すフローチャートである。

【図１０】　レコードデータの取得処理が行なわれるごとに逐次的にアドレス管理用リストを作成する処理を示すフローチャートである。

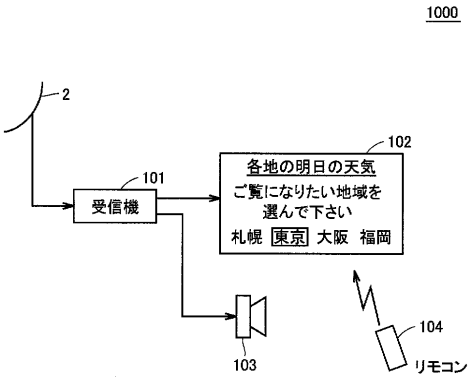
【図１１】　読出の対象となるファイルが初めて読込むファイルではなく、２回目以降の読出処理である場合の処理を示すフローチャートである。

【図１２】　図１１に示した「処理３」を示すフローチャートである。

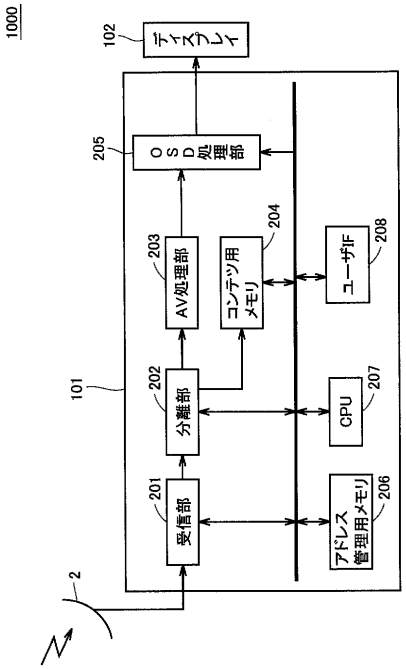
【符号の説明】

２　アンテナ、１０１　受信機、１０２　ディスプレイ、１０３　スピーカ、１０４　リモコン、２０１　受信部、２０２　分離部、２０３　ＡＶ処理部、２０４　コンテンツ用メモリ、２０５　ＯＳＤ処理部、２０６　アドレス管理用メモリ、２０７　ＣＰＵ、２０８　ユーザインターフェイス。

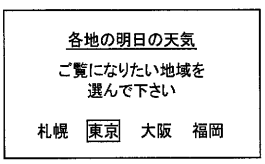
【図１】



【図２】



【図３】



【図４】

【図５】

```
var bt;
function bt_new() {
  bt=new BinaryTable("tenki.bt","1,S:1V,S:1V,I:1B,I:1B");
}
function  getTenkiData(location) {
  var tenki,maxTemp,minTemp;
  if(location=="Sapporo") {
    tenki=bt.toString(0,1);
    maxTemp=bt.toNumber(0,2);minTemp=bt.toNumber(0,3);
  } else if(location=="Tokyo") {
    tenki=bt.toString(1,1);
    maxTemp=bt.toNumber(1,2);minTemp=bt.toNumber(1,3);
  } else if(location=="Osaka") {
    tenki=bt.toString(2,1);
    maxTemp=bt.toNumber(2,2);minTemp=bt.toNumber(2,3);
  } else if(location=="Fukuoka") {
    tenki=bt.toString(3,1);
    maxTemp=bt.toNumber(3,2);minTemp=bt.toNumber(3,3);
  } else return false;
  dispData(tenki,maxTemp,minTemp);// 天気情報の表示
  return true;
}
```

【図６】

[バイナリデータのダンプ内容]

```
        レコード長 "札幌"          "晴れ"      18 10
00000000 0C 04 8E 44 96 79 04 90 B0 82 EA 12 0A 14 04 93
00000010 8C 8B 9E 0C 90 B0 82 EA 82 CC 82 BF 93 DC 82 E8
00000020 16 10 0C 04 91 E5 8D E3 04 93 DC 82 E8 19 0E 0A
00000030 04 95 9F 89 AA 02 89 4A 16 0A
```

[バイナリデータの内容]

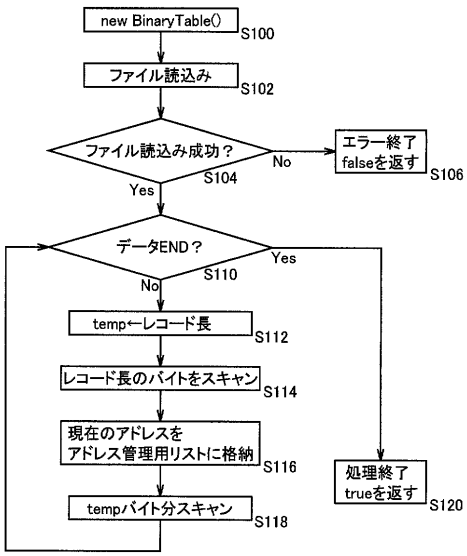
札幌, "晴れ", 18, 10
東京, "晴れのち曇り", 22, 16
大阪, "曇り", 25, 14
福岡, "雨", 22, 10

【図７】

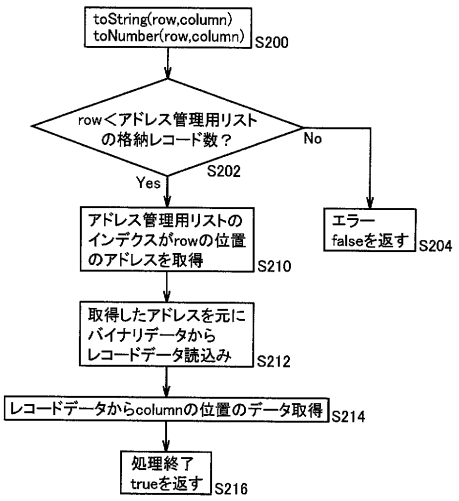
アドレス管理用リスト

record[0]	00000001
record[1]	0000000E
record[2]	00000023
record[3]	00000030

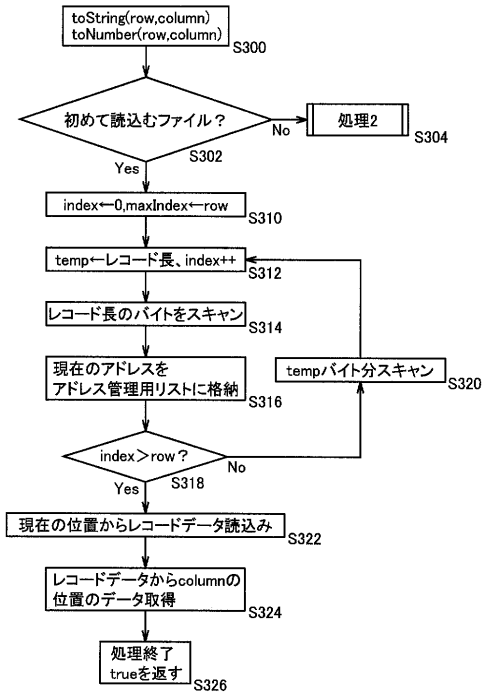
【図８】



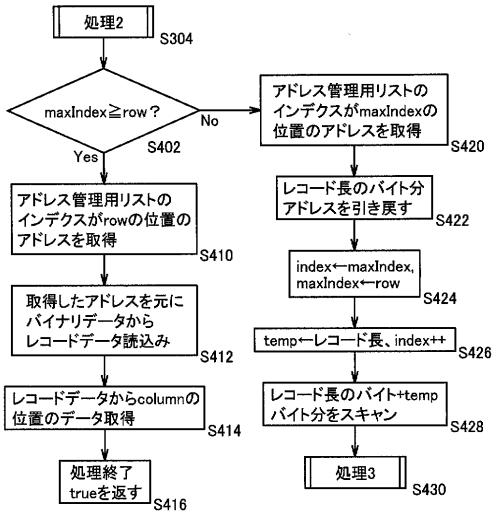
【図９】



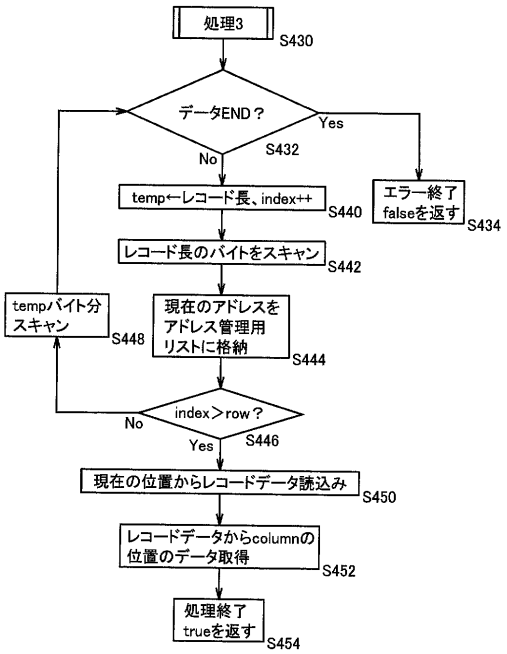
【図１０】


toString(row,column)
toNumber(row,column)
S300
初めて読込むファイル？
No
処理2
S304
S302
Yes
index←0,maxIndex←row
S310
temp←レコード長、index++
S312
レコード長のバイトをスキャン
S314
現在のアドレスを
アドレス管理用リストに格納
S316
tempバイト分スキャン
S320
index＞row？
No
S318
Yes
現在の位置からレコードデータ読込み
S322
レコードデータからcolumnの
位置のデータ取得
S324
処理終了
trueを返す
S326

【図１１】


処理2
S304
maxIndex≧row？
No
S402
Yes
アドレス管理用リストの
インデクスがmaxIndexの
位置のアドレスを取得
S420
アドレス管理用リストの
インデクスがrowの位置の
アドレスを取得
S410
レコード長のバイト分
アドレスを引き戻す
S422
index←maxIndex,
maxIndex←row
S424
取得したアドレスを元に
バイナリデータから
レコードデータ読込み
S412
temp←レコード長、index++
S426
レコードデータからcolumnの
位置のデータ取得
S414
レコード長のバイト+temp
バイト分をスキャン
S428
処理終了
trueを返す
S416
処理3
S430

【図１２】


処理3
S430
データEND？
Yes
No
S432
temp←レコード長、index++
S440
エラー終了
falseを返す
S434
レコード長のバイトをスキャン
S442
tempバイト分
スキャン
S448
現在のアドレスを
アドレス管理用
リストに格納
S444
index＞row？
No
S446
Yes
現在の位置からレコードデータ読込み
S450
レコードデータからcolumnの
位置のデータ取得
S452
処理終了
trueを返す
S454

フロントページの続き

審査官　川口　貴裕

(56)参考文献　特開平０６－２８２４３７（ＪＰ，Ａ）
特開平０７－２６４５６２（ＪＰ，Ａ）
特開平０６－３３８１７９（ＪＰ，Ａ）
特開平０６－３０９８４８（ＪＰ，Ａ）
BinaryTable クラスの運用ガイドライン，デジタル放送におけるデータ放送符号化方式と伝送方式　標準規格　ＡＲＩＢ　ＳＴＤ－Ｂ２４　１．２版　第二編　付属２，社団法人電波産業会，２０００年　６月２０日，ｐ．３１１－３１４
真紀俊男，真紀俊男のローテク講座 第５１回 低レベルの巻，Ｃ　ＭＡＧＡＺＩＮＥ，日本，ソフトバンクパブリッシング株式会社，２０００年　５月　１日，第１２巻，第５号，ｐ．１２０－１２３
森俊也，下地達也，菱田利浩，平位純一，森田克之，石川亮，藤田公一，ＢＳデータ放送Ｅ－Ｅシステム，Ｍａｔｓｕｓｈｉｔａ　Ｔｅｃｈｎｉｃａｌ　Ｊｏｕｒｎａｌ，２０００年１２月１８日，第４６巻，第６号，ｐ．９－１８

(58)調査した分野(Int.Cl.，ＤＢ名)

H04H 20/00 - 20/95
H04H 40/00 - 40/90
H04H 60/00 - 60/98
G06F 17/30
G06F 12/00