(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3577095号
(P3577095)
(45)発行日　平成16年10月13日(2004.10.13)　(24)登録日　平成16年7月16日(2004.7.16)

(51)Int.Cl.[7]　F I
G06F 3/12　G06F 3/12　C

請求項の数 18　(全 12 頁)

(21)出願番号　特願平5-244002
(22)出願日　平成5年9月3日(1993.9.3)
(65)公開番号　特開平6-223021
(43)公開日　平成6年8月12日(1994.8.12)
審査請求日　平成12年8月15日(2000.8.15)
(31)優先権主張番号　939,785
(32)優先日　平成4年9月3日(1992.9.3)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
ＨＥＷＬＥＴＴ－ＰＡＣＫＡＲＤ　ＣＯＭＰＡＮＹ
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　３０００
(74)代理人　100099623
弁理士　奥山　尚一
(74)代理人　100096769
弁理士　有原　幸一
(74)代理人　100107319
弁理士　松島　鉄男
(72)発明者　ロス・キャンプベル
アメリカ合衆国アイダホ州ボイジー・スピンネーカ・コート　１４７３
最終頁に続く

(54)【発明の名称】周辺装置のための制御言語の境界判定方法

(57)【特許請求の範囲】
【請求項１】
入力データ・ストリーム中に現れうる複数の制御言語を、受信し、解釈し、応答するように設計された周辺装置が、第１の制御言語を識別した場合に、前記第１の制御言語と第２の制御言語との間の言語境界を判定する方法であって、
前記データ・ストリーム中のコマンドを解釈して、そのコマンドがページ上にマーキングを要求しているかどうかを判定するステップ（ａ）と、
前記マーキングを要求していると判定された場合に、先行のコマンドが前記ページ上にマーキングを以前におこなったかどうかを判定するステップ（ｂ）と、
前記マーキングを以前におこなってないと判定された場合に、前記言語境界に遭遇したかどうかを判定するステップ（ｃ）と、
前記言語境界に遭遇したと判定された場合に、入力データの制御言語を識別するステップ（ｄ）と、
を含むことを特徴とする周辺装置のための制御言語の境界判定方法。
【請求項２】
前記解釈したコマンドが前記マーキングを要求していないと判定されたばあいに、該コマンドが前記第１の制御言語のなかの予め画定された制御文字であるかどうかを判定し、前記制御文字であると判定されたばあいに、前記解釈されたコマンドを実行するステップをさらに含むことを特徴とする請求項１記載の周辺装置のための制御言語の境界判定方法。
【請求項３】

前記解釈したコマンドが前記制御文字でないと判定されたばあいに、先行のコマンドが前記ページ上にマーキングを以前におこなったかどうかを判定する前記ステップにすすむことを特徴とする請求項２記載の周辺装置のための制御言語の境界判定方法。

【請求項４】

前記周辺装置が前記第１の制御言語ではなくて前記第２の制御言語を識別したばあいに、次のコマンドがジョブマーカーの終端であるかどうかを判定するステップと、

前記終端でないと判定されたばあいに、前記次のコマンドが拡張文字であるかどうかを判定するステップと、

前記次のコマンドが拡張文字であると判定されたばあいに、通信モードがバイナリであるかどうかを判定するステップと、

前記通信モードがバイナリでないと判定されたばあいに、言語境界に遭遇したかどうかを判定するステップと、

言語境界に遭遇したと判定されたばあいに、前記入力データの制御言語を識別するステップと、

をさらに含むことを特徴とする請求項１記載の周辺装置のための制御言語の境界判定方法。

【請求項５】

前記第１の制御言語はＰＣＬであることを特徴とする請求項４記載の周辺装置のための制御言語の境界判定方法。

【請求項６】

前記第２の制御言語は、前記第１の制御言語によりページ上にプリントされる文字を定義するとして認識されるコードでそのコマンドを形成することを特徴とする請求項４記載の周辺装置のための制御言語の境界判定方法。

【請求項７】

前記第２の制御言語はポストスクリプトであることを特徴とする請求項６記載の周辺装置のための制御言語の境界判定方法。

【請求項８】

前記言語境界に遭遇したかどうかを判定するステップは、

前記データ・ストリームを構文解析して、前記各制御言語のキーを識別するステップと、

前記識別されたキーに基づいて、前記言語境界の存在の有無を指示するステップと、

を含むことを特徴とする請求項１または４記載の周辺装置のための制御言語の境界判定方法。

【請求項９】

前記言語境界の存在の有無を指示するステップは、前記識別されたキーにウエイトを付与するとともに、それらのウエイトを蓄積し、この蓄積されたキーウエイトの総和が予め決められたしきい値に到達したかどうかを判定するステップを含むことを特徴とする請求項８記載の周辺装置のための制御言語の境界判定方法。

【請求項１０】

前記キーウエイトの総和が現在使用中の制御言語に関するものであるばあい、その総和が前記しきい値以上のときに前記言語境界ではないと判定することを特徴とする請求項９記載の周辺装置のための制御言語 の境界判定方法。

【請求項１１】

前記キーウエイトの総和が新規の制御言語に関するものであるばあい、その総和が前記しきい値以上のときに前記言語境界であると判定することを特徴とする請求項９記載の周辺装置のための制御言語 の境界判定方法。

【請求項１２】

入力データ・ストリーム中に現れうる複数の制御言語を、受信し、解釈し、応答するように設計された周辺装置が、第1の制御言語を識別した場合に、前記第１の制御言語と第２の制御言語との間の言語境界を判定する装置であって、

ページ上にマーキングを要求するコマンドであって、前記データ・ストリーム中の新規に

受信したコマンドを識別する手段と、
先行のコマンドが前記ページ上にマーキングを以前おこなったかどうかを判定する第１の手段と、
前記第１の手段からのマーキングがおこなわれていないという判定に応答して、前記新規に受信したコマンドと前記入力データ・ストリームの一部が言語境界を示すものであるかどうかを判定する構文解析手段と、
を備えることを特徴とする周辺装置のための制御言語の境界判定装置。
【請求項１３】
前記第２の制御言語は、前記第１の制御言語によりページ上にプリントされる文字を定義するとして認識されるコードでそのコマンドを形成することを特徴とする請求項１２に記載の周辺装置のための制御言語の境界判定装置。
【請求項１４】
前記コマンドを識別する手段によって識別されないコマンドが前記第１の制御言語の第１の制御言語キーのあらかじめ定義されたセットのものであるかどうかを判定し、あらかじめ定義されたセットのものであるばあいに前記コマンドを実行する第２の手段をさらに備える
ことを特徴とする請求項１３記載の周辺装置のための制御言語の境界判定装置。
【請求項１５】
前記周辺装置が前記第１の制御言語ではなくて前記第２の制御言語を識別したばあいに、次のコマンドがジョブマーカーの終端であるかどうかを判定し、終端でないと判定したばあいに、次のコマンドが拡張文字であるかどうかを判定し、拡張文字であると判定したばあいに、通信モードはバイナリであるかどうかを判定し、バイナリでないと判定したばあいに、言語境界に遭遇したかとどうかを判定し、言語境界に遭遇したと判定したばあいに、前記入力データの制御言語を識別する手段をさらに備えることを特徴とする請求項１２記載の周辺装置のための制御言語の境界判定装置。
【請求項１６】
前記構文解析手段は、前記第１と第２の制御言語の言語キーを識別し、各識別されたキーにウエイトを付与して、それらのウエイトを蓄積するとともに、この蓄積したキーウエイトの総和が予め決められたしきい値に到達したかどうかを判定することを特徴とする請求項１５記載の周辺装置のための制御言語の境界判定装置。
【請求項１７】
前記第１制御言語はＰＣＬであることを特徴とする請求項１６記載の周辺装置のための制御言語 の境界判定装置。
【請求項１８】
前記第２の制御言語はポストスクリプトであることを特徴とする請求項１５記載の周辺装置のための制御言語の境界判定装置。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明は、多重制御言語（ｍｕｌｔｉｐｌｅ　ｃｏｎｔｒｏｌ　ｌａｎｇｕａｇｅｓ）をサポートする周辺ユニットに関係し、更に詳細には、異なる制御言語を使用するジョブの間の境界（ｂｏｕｎｄａｒｙ）を識別するための方法及び装置に関するものである。
【０００２】
【従来技術とその問題点】
現代のデータ処理ネットワークは、種々のタイプのコンピュータを通信ネットワークによって相互接続することを可能にする。例えば、プリンタ、プロッタ等の周辺ユニットは、同様に、通信ネットワークに接続され、接続されたコンピュータ用リソースとして役立つ。これらの周辺装置は、種々のコンピュータから受信したコマンドを理解して、これに対し応答することが可能でなければならない。接続された全てのコンピュータが同じ周辺制御言語を採用する場合には、周辺ユニットは、その言語に対してのみ認識および応答する

ことが必要である。しかし、コンピュータが異なる周辺制御言語を用いる場合、周辺ユニットは、受信する異なる言語に対して認識および応答ができるように設計されていなければならない。

【０００３】

米国特許出願番号０７／８２５，４７９（Ｌａｎｇｕａｇｅ Ｉｄｅｎｔｉｆｉｃａｔｉｏｎ Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｆｏｒ ａ Ｐｅｒｉｐｈｅｒａｌ Ｕｎｉｔ”）では、周辺ユニットが受信するデータ・ブロックの構文（ｓｙｎｔａｃｔｉｃａｌ）および文脈（ｃｏｎｔｅｘｔｕａｌ）の解析によって制御言語を識別できるようにするシステムが開示されている。このシステムは、予期される各言語に対して１つのボーター（ｖｏｔｅｒ）モジュールを有し、各ボーター・モジュールは「ディサイダー（ｄｅｃｉｄｅｒ）」モジュールへ、特定の言語に対して賛成または反対の投票を提供する。ディサイダー・モジュールは、各ボーター・モジュールからの入力の解析の際、周辺ユニットを、ボーター・モジュール入力に従ってディサイダーの制御言語プログラムのうちの１つに対して可能（ｅｎａｂｌｅ）にするか、或は、識別できる言語がない場合には、周辺ユニットの動作を中止させる。前述の統合論的及び文脈上の分析システムを確実に作動させるためには、同じでない制御言語を用いるない場合には、周辺ユニットの動作を中止させる。上述の構文及び文脈解析システムを確実に作動させるためには、異なる制御言語を採用するジョブの間の境界が正確に識別されなければならない。

【０００４】

多重制御言語をサポートする周辺ユニットで現在使用されている１つの方法では、「タイムアウト（期限切れ）」ジョブ分離と呼ばれる方法が用いられている。この手順では、周辺ユニットが次の言語を決定することが可能になる状態に入る以前にジョブの間において、期限切れとなる一定の時間を要求される。次のジョブが期限切れが終了する前に現れた場合、周辺ユニットは言語変化がなく、そして、数百ページの読みにくいテキストを与えるようなミスプリントが生じたと解釈する。

【０００５】

多重周辺制御言語を用いる多くのシステムにおいては、周辺ユニットに入力されるジョブに対して補助記憶装置を供給するために、通信ネットワークと周辺ユニットとの間にバッファ・デバイスが配置される。このようなバッファ・デバイスは、しばしばタイムアウト・ジョブ分離操作をサポートしない。従って、周辺ユニットは、バッファ・デバイスから順に到着するほぼ又は全くタイムアウト分離のない複数のジョブを受信することもあり得る。

【０００６】

一般に用いられる２つのプリンタ制御言語は、ヒューレット・パッカード・カンパニーのＰＣＬ（Ｐｒｉｎｔｅｒ Ｃｏｍｍａｎｄ Ｌａｎｇｕａｇｅ）と、ＰｏｓｔＳｃｒｉｐｔ（米国アドビー社の商標）である。ＰｏｓｔＳｃｒｉｐｔ言語のジョブは一般にジョブ・エンド・マーカ（ｅｎｄ－ｏｆ－ｊｏｂ）を使用し、それによって、複数の印刷ジョブの描写を簡素化する。これとは対照的に、ＰＣＬのジョブは、ジョブ・エンド・マーカを使用しないことが多い。いずれの場合も、現在のジョブと後続のジョブとの間で制御言語が変化した場合、現在のジョブのエンドが適切にマークされていないと、連続するジョブの制御言語はイネーブルされない。

【０００７】

【発明の目的】

従って、本発明の目的は、制御言語ジョブ境界の発生についてチェックするため、入力データ・ストリームをサンプリングするタイミングを決定するための周辺ユニットのための改良された方法及びシステムを提供することにある。

本発明の他の目的は、入力データが現在の制御言語または新規の制御言語のどちらで表現されているかを決定するための改良された言語境界識別手順を提供することにある。

本発明の更に他の目的は、ジョブ・エンド・マーカを用いない、周辺制御言語のための改良された言語境界識別手順を供給することにある。

【０００８】
【発明の概要】
周辺ユニットは、入力データ・ストリームの中で、複数の制御言語を受信し、そして、各制御言語に応答するように機能する。周辺ユニットは、制御言語ジョブ境界の発生をいつチェックするのかを決定する。
【０００９】
ＰＣＬでは、制御言語ジョブ境界の発生をいつチェックするのかを決定するステップを次に示す。入力データ・ストリームから次のコマンドを識別し、識別したコマンドがＰＣＬ制御キャラクタあるいはページ上に形成されるマークのための呼び出しであるか判断し、さらに、ページが以前にマークされているかどうかを判断する。前述の判断基準が満たされると、コマンド及び後続のデータを、ジョブの境界が生じいるかどうかを決定するために調べる。従って、データのフォーマット化されたＰＣＬページの開始時に、ジョブの境界が生じたかどうかを決定する。
【００１０】
ＰｏｓｔＳｃｒｉｐｔでは、制御言語ジョブ境界の発生をいつチェックするかを決定するステップは、次のとおりである。データ・ストリームをプリスキャン（予備走査）し、次のコマンドがジョブ・エンド・マーカであれば、ＰｏｓｔＳｃｒｉｐｔから出て、入力データの言語を識別するために戻り、また、次のコマンドが拡張文字であって、現在のＩ／Ｏプロトコルが２進数でない場合、ジョブ境界が生じたかどうかを決定するために、コマンド及び後続のデータを調べる。
【００１１】
ジョブ境界は、入力データ・サンプルを調べ、そして、そのシンタックスが現在のジョブの制御言語の特徴であるかまたは異なる制御言語のものであるかを決定する。入力データ・サンプルのシンタックスが現在の制御言語のシンタックスと異なる場合、ジョブ境界が生じている。
【００１２】
【発明の実施例】
図１では、パーソナルコンピュータ１０、１２、１４は、ローカルエリアネットワーク（ＬＡＮ）１６または他の同様な通信ネットワークによって相互接続されている。プリンタ１８は、プリント・ジョッブ・バッファ・デバイス（ｐｒｉｎｔ　ｊｏｂ　ｂｕｆｆｅｒ　ｄｅｖｉｃｅ）１９を介して、ＬＡＮ１６に接続されている。プリント・ジョブ・バッファ・デバイスによっては、内部通信機能を有し、これらはＬＡＮ１６を必要としないことに注意されたい。
【００１３】
プリンタ１８は、パーソナルコンピュータ（ＰＣ）１０、１２、１４の夫々にプリントのサービスを提供する。本願明細書では、本発明をプリンタ１８に適用する場合について説明するが、接続されたコンピュータに対してサービスを提供する他の周辺ユニットに対しても、本発明を同様に利用できることは明らかである。このような他の周辺ユニットとは、例えば、プロッタ、ファクシミリ・ユニット等がある。
【００１４】
コンピュータ１０、１２、１４は、それぞれ異ったプリンタ制御言語を用いることが可能である。同様に、これらコンピュータは１台で複数のプリンタ制御言語を採用することもできる。本発明を説明するために、ＰＣ１０は、ヒューレット・パッカード・カンパニーのＰＣＬとして知られているプリンタ制御言語を採用し、ＰＣ１２はＰｏｓｔＳｃｒｉｐｔプリンタ制御言語を採用するものと仮定する。ＰＣ１４は、他のプリンタ制御言語を採用するものと仮定する。
【００１５】
ＰＣ１０、１２、１４のうちの１つがプリント・ジョブを送信する場合、プリント・ジョブは、ＬＡＮ１６を介して、プリント・ジョブ・バッファ・デバイス１９へ送られる。この場合、プリント・ジョブは待ち行列を作り（ｑｕｅｕｅｄ）、プリンタ１８へシリアル

に供給される。

【００１６】

プリンタ１８は、中央処理装置（ＣＰＵ）２０、入出力（Ｉ／０）モジュール２２、ＲＡＭモジュール２４及びＲＯＭモジュール２５、２７を有する。もちろｎ、ＲＯＭモジュール２５、２７は、単一のＲＯＭとして構成されてもよい。これらの各ユニット／モジュールは、バス２６を介して、プリント・エンジン（ｐｒｉｎｔ　ｅｎｇｉｎｅ）２８まで相互接続される。ＲＯＭモジュール２５は、言語識別手順のために用いられるプログラムを有し、ＲＯＭ２７は、言語境界識別のためのプログラムを有する。ＲＡＭ内の３０で示される１つの領域は、ＰＣ１０、１２、１４のうちの１つから受信するデータのブロックを記憶する領域である。ＲＯＭ２５内のファームウェアは、プリンタ１８に対して言語識別手順を具体化し（ｅｍｂｏｄｉｅｓ）、複数のボーター（ｖｏｔｅｒ）モジュール３４、３６、３８から構成される。これらはそれぞれ採用が予測されるプリンタの言語の解析および解釈に特別に割り当てられる。ディサイダ（ｄｅｃｉｄｅｒ）モジュール４０は、ボーター・モジュール３４，３６、３８からの出力を受信し、そして、それらのボーター・モジュール入力に基づいて、ＲＡＭ領域３０に記憶されているデータ制御言語のうちのどの制御言語が最も類似していると識別されるかを決定する。次に、ディサイダ・モジュール４０は、ＣＰＵ２０に対して、決定がくだされた言語を解釈し、実行することを可能にする構文デコーディング・ソフトウェアへアクセスするように指示する。言語識別手順の動作、多様なボーター・モジュール及びディサイダ・モジュールの動作については、例えば、前述した米国特許出願にも詳細に説明されている。

【００１７】

言語識別手順を適切に作動させるためには、各ジョブの境界が明確に識別されなければならない。ＰｏｓｔＳｃｒｉｐｔジョブは、典型的には、ジョブ・エンド・マーカを使用するので、このような識別は一般に簡単である。対照的に、ＰＣＬジョブは、ジョブの終端を識別するためにジョブ・エンド・マーカを採用していないので、別の制御言語（すなわちＰｏｓｔＳｃｒｉｐｔ）への切り換えは容易ではない。本発明は、ＰｏｓｔＳｃｒｉｐｔがそのコマンド（同じくデータ）を具体化するためにＡＳＣＩＩ文字群を使用している事実を利用するものである。しかし、ＰｏｓｔＳｃｒｉｐｔの文法解析（ｐａｒｓｅｒ）は、定義されたＡＳＣＩＩ群はコマンドとして認識され、解釈されるが、ＰＣＬ文法解析では、ＡＳＣＩＩ文字を表現された特定の文字に従ってページをマークするためのコマンドとして認識、解釈される。従って、プリンタがＰＣＬ文棒解析によって制御される場合、コマンドがページ・マークを必要としないコマンドでない限り、プリンタはページをマークすることにより、ＰｏｓｔＳｃｒｉｐｔのコマンドの最初の文字に応答する。加えて、本発明は、ＰＣＬが多くのコマンドを開始するために拡張文字（ＡＳＣＩＩ２７）を用いること及び非２進数ＰｏｓｔＳｃｒｉｐｔデータ・ストリーム中に拡張文字が存在することは極めてまれであるということも利用する。従って、ＰＣＬジョブを開始するには感知された拡張文字を使用することが極めて好ましく、この場合には、ＰｏｓｔＳｃｒｉｐｔジョブが適切なジョブ・エンド・マーカで終結しなかったことにより、ＰｏｓｔＳｃｒｉｐｔジョブが「不良形成された（ｍａｌ－ｆｏｒｍｅｄ）」ものと仮定される。

【００１８】

ＲＯＭ２７における言語境界識別手順は、（１）ＰＣＬと、異なる制御言語を用いる後続のジョブとの間のジョブ境界の識別を可能とし、（２）ＰｏｓｔＳｃｒｉｐｔと、異なる制御言語を用いる後続のジョブとの間のジョブ境界の識別を可能にする。言語境界識別手順（ＣＰＵ２０との組み合わせ）によって用いられるステップを図２に示す。言語境界（ＬＢ）文法解析４２はＲＯＭ２７内に含まれており、そして、境界識別手順において支援する（以下に詳細に説明する）。

【００１９】

最初に、図２において、プリンタ１８（図１）に示される言語識別手順により、現在処理中の入力データ・ストリームの言語を識別する（デシジョン・ボックス５０）。識別された言語がＰｏｓｔＳｃｒｉｐｔであれば、図２の左側に示す手順を下方に向かって繰り返

す。これとは対照的に、現在識別された言語がＰＣＬであれば、システムは図２の右側に示す手順を下方に向かって繰り返す。識別された言語がＰｏｓｔＳｃｒｉｐｔである場合、手順は次のコマンド（ボックス５４）にアクセスする。非２進数（ｎｏｎ－ｂｉｎａｒｙ）ＰｏｓｔＳｃｒｉｐｔジョブには、拡張文字が存在することは極めて稀れである。次のコマンドがジョブ・エンド・マーカ（ボックス５６）を指示するならば、現在のジョブを終結し、手順サイクルはボックス５０に戻り、新たににジョブの言語を識別する。コマンドがジョブ・エンド・マーカでないならば、拡張文字（ＡＳＣＩＩ２７）であるかどうかを見るために、次のコマンドを試験する（デシジョン・ボックス５７）。コマンドが拡張文字で、しかも、現在のＩ／０プロトコルが２進法でないならば（ボックス５９）、制御言語境界が生じたかどうかを決定するためにデシジョン・ボックス６８を呼び出す。

【００２０】

現在識別済みの言語がＰＣＬであるとデシジョン・ボックス５０によって示された場合、手順は次のコマンドを取り出すように繰り返される（ボックス６０）。ここで、ＰＣＬは文字をプリントするためのコマンドとして、ＡＳＣＩＩ表現された文字が認識する。従って、ページ上に１つのマークがプリントされるべきであることを示すＰＣＬコマンドのグループの１つであるかどうかを決定するために、取り出したコマンドを構文的に解析する（デシジョン・ボックス６２）。そうでない場合、ページをマークしないが、言語は継続してＰＣＬであることを高い信頼性をもって指示する１組の予め定義されたＰＣＬ制御文字と当該コマンドが一致するかどうかを決定するために、当該コマンドを更に解析する。それらのコマンドは、ＡＳＣＩＩ１０進数の値は０、７－１５、２７（即ち、「拡張」文字）を有している。このようなコマンドを見つけた場合（デシジョン・ボックス６３）、決定されたＰＣＬのマーキングしないコマンドが実行され（ボックス６４）、そして、次のコマンドを取り出すために手順が繰り返される。

【００２１】

ページのマーキングを要求するコマンド（例えば、ＡＳＣＩＩ文字）を認識し（デシジョン・ボックス６２）またはデシジョン・ボックス６３に示された解析手順によって、当該コマンドが定義されたＰＣＬ制御文字でないと示す場合、現在のページが既にマーキング済みかどうかを決定するため繰り返す（デシジョン・ボックス６６）。そうでないならば、可能なジョブ境界に到達したものと仮定する。マークがページ上に配置されると、直ちに、マーキング・フラグをセットし、それ以後、ページがマークされたかどうかを決定するためにはマーキング・フラグがセットされたかどうかを調べる。現在のページが完成し、プリントされるまで、このフラグはセットされたままに維持され、プリントされた時点でマーキング・フラグをリセットする。

【００２２】

デシジョン・ボックス６８では、コマンド及びシンタックスが「現在の」制御言語または「新規な」制御言語における「キー」を表すかどうかを決定するためにデータ・サンプルを解析することにより、ジョブ境界が生じたどうかを決定する。「キー」は、制御言語のコマンド・ルール内で定義された単一の文字または文字群である。ボックス６８では、データ・サンプル中からキーの識別を継続しておこない、そして、データ・サンプルの終端が生じたかまたは予め決められた現在の言語または新規な言語のキー・ウエイトしきい値（ｋｅｙ－ｗｅｉｇｈｔ　ｔｈｒｅｓｈｏｌｄ）を越えるまで、現在の言語及び新規な言語のウエイトを蓄積する。現在の言語または新規な言語のキーウエイトしきい値を越える前に、データ・サンプルの終端が生じていたら、ジョブ境界は生成されていない。現在の言語のキーウエイトしきい値を越えると、ジョブ境界は生成されていない。新規な言語のキーウエイトしきい値を越えると、ジョブ境界が生成されている。ボックス６８の解析によって、ジョブ境界が生成されたことが指示された場合、入力データの言語を識別するためにボックス５０が呼び出されて、新規な言語がイネーブルとなり、そして、イネーブルの言語を用いる新規なジョブが開始される。

【００２３】

上述より、ジョブ境界に関してデータ・ストリーム中の一点を探しだす機構が可能となり

、そして、その点において、制御言語が現在の制御言語と同じであるか、または、新規な制御言語と同じであるかを決定するために、データの簡単な解析がおこなわれることになる。

【００２４】

図３のフローチャートでは、図２のデシジョン・ボックス６８に示されたソフトウェアによって行われる動作を示す。以下に手順に際して識別される機能は、ＲＯＭ２７内のＬＢ文法解析（ＬＢ　ｐａｓｅｒ）４２によって実施される（図１参照）。ＬＢ文法解析４２は、入力データ・ストリームの開始時に、データ・サンプルからデータを読み取る。ＬＢ文法解析４２は一連のキーを認識し、データ・サンプルを解析すると各キーをカウントする。キーが生成されると、アキュムレータにインクリメントが加算される。各キーを受信し、対応するアキュムレータがインクリメントされた後、データ・サンプルがＰｏｓｔＳｃｒｉｐｔまたはＰＣＬであるかどうかを決定するために、アキュムレータ値を手順にしたがって評価する。ＬＢ文法解析４２は、ＰｏｓｔＳｃｒｉｐｔまたはＰＣＬによって明確な決定ができるまで、入力データを文法解析する。明確な決定がくだすことが可能となると、それ以上の解析は完了する。

【００２５】

特定のキーにはそれぞれウエイト値（ｗｅｉｇｈｔ　ｖａｌｕｅ）が割り当てられ、それが検出されると、キーのタイプに割り当てられたウエイトの量だけそれぞれ対応するアキュムレータがインクリメントされる。従って、あるタイプのキーは他のタイプのキーよりもアキュムレータ値に及ぼす影響が大きく、前者は後者よりも言語の存在に対してより指示される。ＬＢ文法解析４２によって認識され、ＰＣＬとＰｏｓｔＳｃｒｉｐｔとを区別するために用いられるキーを次に示す。キーは、ＰＣＬディスクリミネータ（ｄｉｓｃｒｉｍｉｎａｔｏｒ）及びＰｏｓｔＳｃｒｉｐｔディスクリミネータとに分類される。特に注記しない限り、キーの生成毎にキー・ウエイトがカウントされる。

【００２６】

ＰｏｓｔＳｃｒｉｐｔキー

（１）特殊ジョブ開始制御文字

次の制御文字シーケンスのうちの１つが発生すると、データ・サンプルの開始時に一度だけカウントされる。：＾Ａ、＾Ｃ、＾Ｄ、＾Ｓ、＾０、＾Ｔ

（２）ＰｏｓｔＳｃｒｉｐｔヘッダコメント

サブストリング「％！ＰＳ」と同じラインのテキスト・ストリング又はサブストリング「％！ＰＳ」、「％！」、「ＥＰＳＦ」。

（３）非同期制御文字（ａｓｙｎｃｈｒｏｎｏｕｓ　Ｃｏｎｔｒｏｌ　Ｃｈａｒａｃｔｅｒｓ）は、データ・サンプルの残りを評価する前に、カウントされ、ストリップされる。：＾Ｔ、＾Ｃ

（４）ＰｏｓｔＳｃｒｉｐｔ一意キーワード

予め定義されたＰｏｓｔＳｃｒｉｐｔオペレータは、一般に、ＰｏｓｔＳｃｒｉｐｔ言語の外側では発見されない。

（５）経験的なＰｏｓｔＳｃｒｉｐｔキーワード

ＰｏｓｔＳｃｒｉｐｔアプリケーションは使用するが、ＰｏｓｔＳｃｒｉｐｔ言語の一部ではない周知の非英語（ｎｏｎ－Ｅｎｇｌｉｓｈ）ワード。

（６）ＰｏｓｔＳｃｒｉｐｔ文書化キーワード

ＡｄｏｂｅによってＰｏｓｔＳｃｒｉｐｔ文書化キーワードとして定義されたワード。これらのワードは、ＰｏｓｔＳｃｒｉｐｔコメントにおける最初のワードとして発生する場合に限り、カウントされる。

（７）ＰｏｓｔＳｃｒｉｐｔコメント

テキストは「％」で始まり、ラインの終端を通過して継続する。「フォームフィード拡張」文字に遭遇した場合には、コメントの評価は早期に完了される。

（８）ＰｏｓｔＳｃｒｉｐｔ中立キーワード

ＰｏｓｔＳｃｒｉｐｔ言語以外の意味を有し、通常の英語に含まれることもある、予め定

義されたＰｏｓｔＳｃｒｉｐｔオペレータ。
【００２７】
ＰＣＬキー－
（１）拡張文字
ＰＣＬ拡張シーケンス・コマンドの開始時に、用いられるＡＳＣＩＩ値が２７の文字。
（２）フォームフィード文字
現在フォーマット化されたＰＣＬページをプリントさせるＡＳＣＩＩ値が１２の文字。
（３）未知ワード
予め定義されたＰｏｓｔＳｃｒｉｐｔキーとマッチしないテキスト・ストリング。
【００２８】
読み取られた最初の文字が特殊制御文字であった場合には、上記の特殊ジョブ開始制御文字は、ＰｏｓｔＳｃｒｉｐｔジョブの開始を強く指示する。従って、このキーはＬＢ文法解析４２によってただ１度だけカウントされるが、データ・フローの最初にみつけた場合には大きく加重される。これとは対照的に、中立キーワードはＰｏｓｔＳｃｒｉｐｔばかりでなく他の言語も同様に指示することができるので、わずか加重されるだけである。しきい値は経験に基づいて決定することも可能である。これにより、蓄積されたキーワード値ウエイトの合計が限界を越えた場合、ＰｏｓｔＳｃｒｉｐｔ言語が決定されていると判定する。ＬＢ文法解析４２は、解析手順を短縮法（ｆｏｒｅｓｈｏｒｔｅｎ）で表わし、一時的な選択以外の言語であることを明確に示す特定の文字に遭遇した場合には、一時的な言語の選択を拒否することができる。ＬＢ文法解析４２によって実施される解析には、言語識別にとって同様に重要であるキーの文脈上の関係が考慮されていない。従って、上述の特許出願に開示される詳細な言語識別手順によって、誤りが実施されたことが指示された場合には、判定が逆転することもあり得る。それにも拘わらず、上述ンの分析は、ジョブ言語境界の可能性が高いことを示すには十分である。
【００２９】
図３に示す前述の手順において、ボックス８０はデータ・サンプルが現在の制御言語または別の制御言語を表すかどうかを決定するためにＬＢ文法解析４２によって実施されるキー識別手順を示す。このキー識別手順は、言語決定が可能になるかあるいはデータ・サンプルの終端に到達する時まで、データ・サンプル内において認識されたキーに関するキーウエイトを蓄積することによって実施される。データ・サンプルの終端に到達し、しかも、言語決定がおこなわれなかった場合には、ジョブ境界は生成されない。
【００３０】
この手順は、データ・サンプルに対して次のように繰り返される。データ・サンプル中の次の識別されたコマンド（ボックス８０）が現在の言語キーにマッチする場合（デシジョン・ボックス８２）、そのキーへのキーウエイトは、現在の言語総キーウエイトに蓄積される（ボックス８８）。データ・サンプル中の次の識別されたコマンドが、新規な言語キーとマッチする場合には（デシジョン・ボックス８２）、キーウエイトは、新規な言語総キーウエイトに蓄積される（ボックス８４）。現在の言語キーウエイト総量が予め決められたしきい値に等しいか、あるいは、それより大きい場合には（デシジョン・ボックス９０）、ジョブ境界はなく（ボックス９２）、現在の制御言語に対して処理を継続する。新規言語のキーウエイト総量が、予めきめられたしきい値に等しいか、或いは、それより大きい場合には（デシジョン・ボックス８６）、ジョブ境界が生じ（ボックス８７）、図２のボックス５０の手順をでて、入力データ・サンプルの新規な制御言語の識別をおこなう。
【００３１】
以上の説明は本願発明の例を示すに過ぎないことを理解されたい。当業者であれば、本発明に趣旨から逸脱することなく、さまざまな変更および改善が可能であることは自明のことである。
【００３２】
【発明の効果】

以上説明したように、ジョブ・エンド・マーカを用いない、周辺制御言語のジョブ境界を識別し、周辺装置のための入力データ・ストリームをサンプリングするタイミングを決定することを可能とする。さらに、本発明によって、入力データが現在の制御言語または新規の制御言語のどちらで表現されているかを決定することを可能とする。

【図面の簡単な説明】

【図１】本発明の周辺装置のための制御言語の境界判定方法をおこなうシステムの概略図。

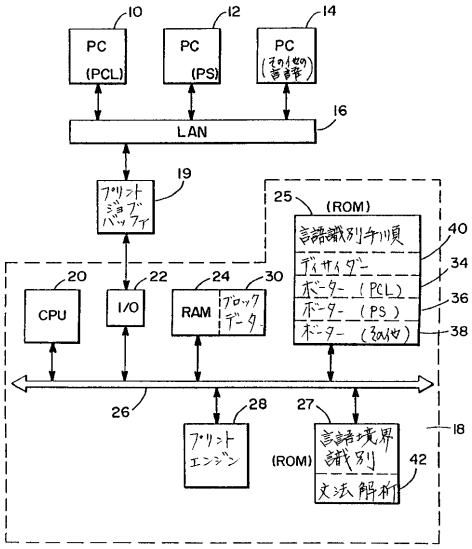
【図２】第１図のシステムの制御言語の境界判定の動作を説明するフローチャート。

【図３】第２図に加えて、文法解析の動作を説明するフローチャート。

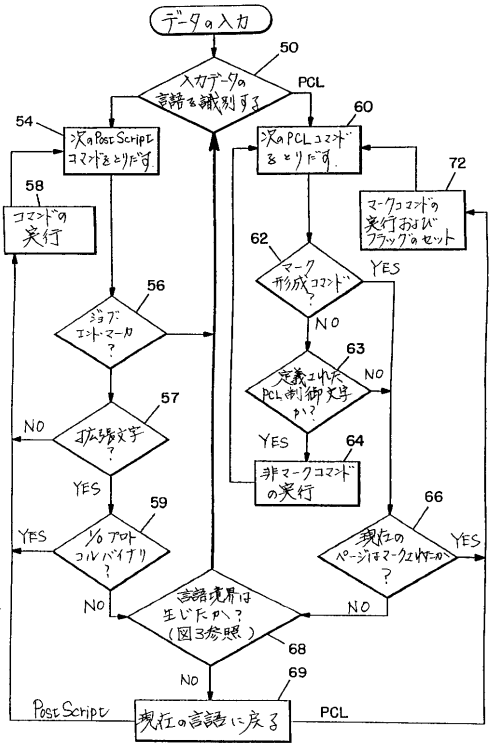
【符号の説明】

１０、１２、１４：ＰＣ
１６：ＬＡＮ
１９：プリント・ジョブ・バッファ
２０：ＣＰＵ
２２：入出力装置
２４：ＲＡＭ
２５、２７：ＲＯＭ
２８：プリント・エンジン

【図１】



【図２】



【図３】

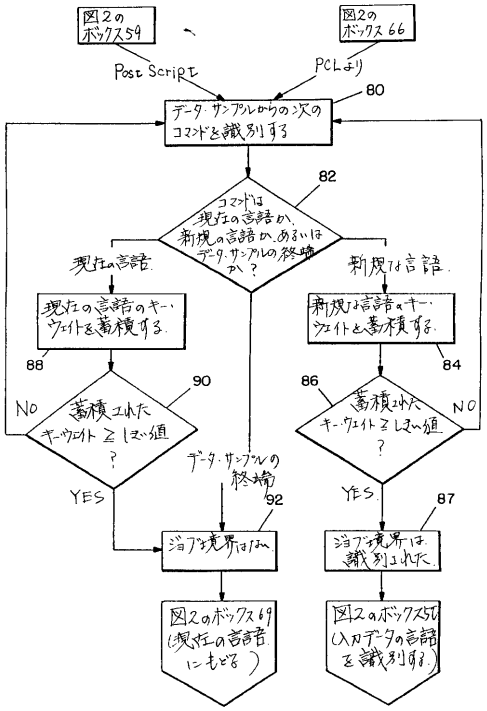


図2のボックス59
図2のボックス66
Post Script
PCLより
80
データ・サンプルからの次のコマンドを識別する
82
コマンドは現在の言語か、新規の言語か、あるいはデータ・サンプルの終端か？
現在の言語
新規な言語
現在の言語のキー・ウェイトを蓄積する
新規な言語のキー・ウェイトを蓄積する
88
84
90
86
蓄積されたキー・ウェイト≧しきい値？
NO
YES
データ・サンプルの終端
92
87
ジョブ境界はない
ジョブ境界は識別された
図2のボックス69（現在の言語にもどる）
図2のボックス56（入力データの言語を識別する）

フロントページの続き

(72)発明者　ダニエル・エル・ジョージ
アメリカ合衆国アイダホ州ボイジー・クリー・ウェイ　４７８５
(72)発明者　トーマス・ティー・パース
アメリカ国アイダホ州ボイジー・ラムロッド・ロード・ダブリュー　１２０８３
(72)発明者　マイケル・イー・スローン
アメリカ国アイダホ州ボイジー・クリー・ウェイ　５０２６

審査官　須藤　竜也

(56)参考文献　特開昭６３－２１６７５６（ＪＰ，Ａ）
特開平２－９９３７７（ＪＰ，Ａ）
特開昭６４－６４０２９（ＪＰ，Ａ）
特開昭４－２３０５１４（ＪＰ，Ａ）
特開平５－３３４０２４（ＪＰ，Ａ）
特開平５－１０４８２５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F　3/12
G06F 13/10