7T−8

ｕｎｉｘ及びＳＣＳＩインタフェースを用いた端末の実現例

橋口　俊彦
（ＮＴＴデータ通信株式会社）
山口　雄一郎
（沖電気工業株式会社）

１．はじめに

近年、電算システムの処理分散化及び高機能化が進み、端末にはより多くの周辺装置（プリンタ、文字認識装置、ＬＡＮ、ＦＤＤ、カードリーダ等）を制御する能力が求められている。また、各周辺装置の処理状態や媒体情報等を的確かつタイムリーに取得し、処理することが必要である。加えて、処理の多重化を図り、電算システムトータルのスループットを向上させることも重要である。

こうした要請に応えるため、専用端末の世界では専用ＯＳ（モニタ）を利用するものが大半を占めている。これは、専用ＯＳが汎用のものに比べ、一般的にコンパクトであり、周辺装置の特性を活かしやすく、かつリアルタイムな処理に適しているためと考えられる。

しかし、端末のプログラム規模が飛躍的に増大しつつある状況で専用ＯＳを使用することは、生産物の他システムへの流用が困難になること及び使用するＯＳに精通したプログラマが限られることなどで問題がある。

よって、汎用ＯＳ及び標準的なインタフェースを用いて端末を構築することとし、それぞれが持つ規格の範囲内で端末を効率良く使用するための問題点及びその対策について検討、対処したのでその実現方法について提案する。

２．端末に適用するための課題

図１に本検討内容を適用したシステムでの業務の流れを、図２に端末装置のハードウェア概要を示す。本端末には、既に業界で標準ＯＳと認められつつあるｕｎｉｘと周辺装置インタフェースとして汎用性が高く、双方向通信が可能で、転送速度も比較的高速に属するＳＣＳＩを用いている。本端末装置の構成で快適な業務処理を実現するため以下の項目を条件として設定した。

①各装置からの情報（故障状態、スイッチ・媒体のセット等）をリアルタイムに取得できること。
②ＯＳ、インタフェースプロトコル等には、極力手を入れない。また、装置及び業務処理の拡張性を十分に考慮すること。
③各装置をできる限り並列動作させ周辺装置の処理能力を最大限に発揮させること。

この項目を実現するための主要な課題として以下の２点が上がり、これに対する解決策を検討した。

①ｕｎｉｘ等のＯＳが処理依頼（システムコール）を受けると原則として処理が完了するまで上位プログラムに応答を返却しないこと
②ＳＣＳＩインタフェースでの双方向転送(*)におけるＯＳ及び上位プログラムとのデータ受渡方法

注(*) 双方向転送例：
ⅰ プリンタへの印字データ転送とプリンタで非同期に発生した事象のＷＳへの通知
ⅱ 文字読み取り装置からの文字情報取得とプリンタへの印字データ転送

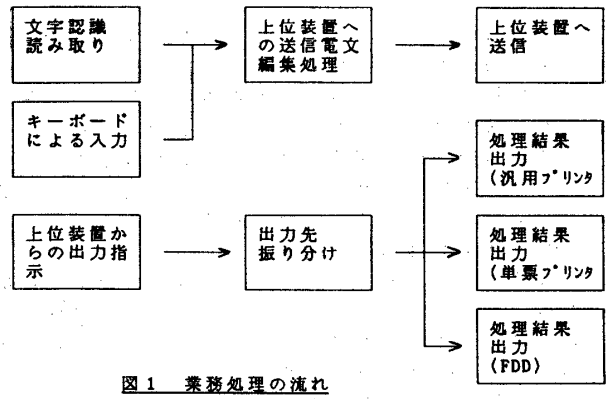

図１　業務処理の流れ

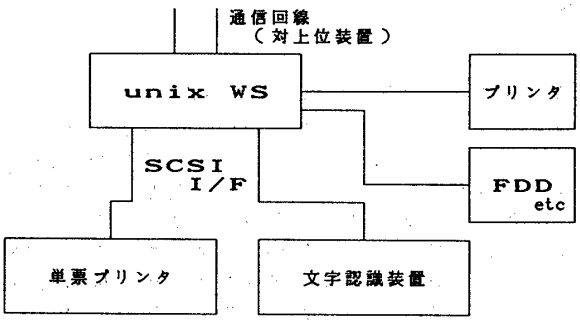

図２　端末ハードウェア構成

An Implementation of unix™ and SCSI to the terminal system
Toshihiko HASHIGUCHI , Yuichiro YAMAGUCHI
NTT DATA COMMUNICATIONS SYSTEMS CORPORATION. ,
Oki Electric Industry Co.,Ltd.

3．業務処理プログラムの構造

課題の①項に関する対処として、ＯＳは通常、比較的単純なイベント処理の解決策として、ソフトウェア割り込み機能（unixでいうsignal等）を提供するが、これだけでは以下の２点が解決できない。

①ソフトウェア割り込みによってＯＳから得られる情報は非常に少ない。これに対処するため、通常、業務処理プログラムが装置の変化に対応できる状態になった時点で、改めて装置情報をセンスすることが多い。そのため、処理の遅延が生じ、業務処理プログラムが装置状態にうまく対応できなくなる事象が発生する。

②業務処理実行中、非同期に割り込みルーティンに移行するため、ユーザに提供するシステムとして十分な品質を保証するには、業務処理実行中のあらゆるタイミングで対処されていることを確認する必要がある。

①については、ＯＳとのインタフェースを追加すること（例えば、ＯＳで割り込み発生時業務処理プログラムの領域にセンス情報を埋め込む等）が考えられるが、これでは汎用ＯＳを利用するメリットが薄らいでしまう。

そのため、本端末プログラムでは業務処理部と装置管理部をプロセス構成上分離した。その上で、装置管理部が常時イベント取得用のシステムコールを発行することとし、上記のようなインタフェースの特例を設けることなく、非同期に発生する情報を取得する方式を採用した。本処理により、イベント発生時に極小時間で装置状態を取得することが可能となり、装置とのずれを最小限にとどめている。

②については、業務処理部と装置管理部とのインタフェースをメッセージ方式による送受に統一することで対処している。すなわち割り込みのタイミングを、常に業務処理部のアイドル時のみとすることにより細かいタイミングでの処理抜けを最小限に抑えることができる。また、このように統一されたインタフェースによるプログラム構成を取ることによって、以下の利点が生まれる。

①他システムへの流用が容易
②ソフトウェアオーバヘッドを防止
③安定した品質を提供

4．ＳＣＳＩインタフェースでの実現方法

ＳＣＳＩインタフェースを用いての双方向通信としては、双方が共にイニシェータとなり情報転送が必要な時に相手をターゲットとして起呼する構成（マルチイニシェータ構成）が一般的である。

しかし、この方式はＳＣＳＩの特性を活かしているものの、ＯＳが業務処理プログラムから情報取得要求や非同期イベントの取得要求を受けていないときでも、周辺装置からの情報を受け取る必要があるため、ＯＳなどのインプリメントが複雑になるという欠点がある。この簡易な対処として、ＯＳ内部に十分なバッファ領域を持つことも考えられるが、これは作成するソフトウェアのフレキシビリティなどに問題が生じる。なぜならバッファとして必要な領域はその時のシステム負荷や使用方法などによって大きく変動する要素であり、種々の条件による領域の変更をＯＳレベルで必要とするためである。

よって、本端末装置はＷＳのみイニシェータとする、いわゆるシングルイニシェータ構成とし、周辺装置はリコネクト処理によってのみＷＳへ再接続する形態を取った。また、装置管理部が常時イベント取得用のシステムコールを発行しているため、データ受渡に関する処理は同期を取りつつ動作することができる。

ところで、上述の構成にする場合、装置への処理依頼と装置からの情報取得をうまく両立させる手段が必要となる。この対策として本システムでは、ＳＣＳＩのＬＵＮ機能（論理ユニット）を用いている。すなわち、１装置に対して処理依頼受付部と非同期情報通知処理部を別に用意し、各々独立した論理ユニットとして用意するものである。

また、これに伴い装置管理部を装置の状態監視を行なうプロセスと装置へ処理依頼を行なうプロセスに分割することとした。これによって装置の状態監視処理を単純化することができ、高負荷の状態変化に対応することが可能となった。

処理イメージを図３に示す。

5．まとめ

現在、本端末を用いたシステムは実際にユーザへのサービスを開始している。この運用において、本端末は十分な多重処理を行っており、実運用上問題ないものと考えている。

今後、この構成を他のＯＳ上でも適用し、さらに検証を深めていく予定である。

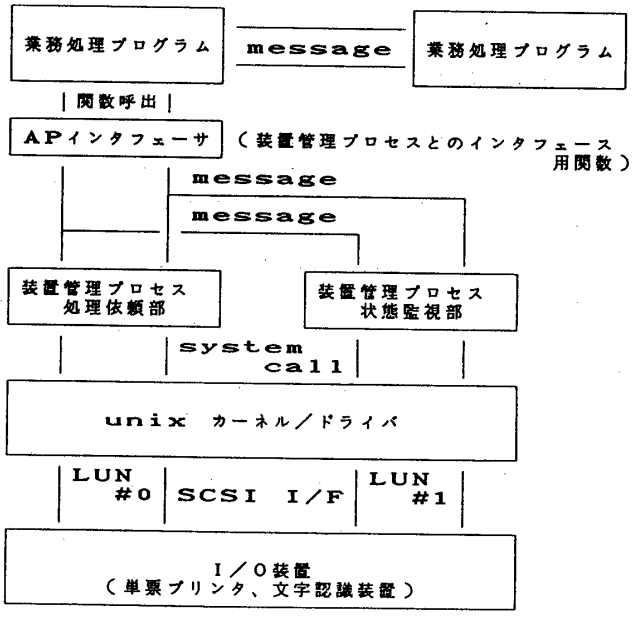

図３　ＡＰ／ＳＣＳＩインタフェース概要