

1Y-1 UNIXを用いたMHS処理方式の一検討

山上俊彦 本村公太 春田勝彦

NTT 電気通信研究所

1. はじめに

オフィスでの電子メールの普及とともに、相互接続が重要な課題となっている。国際的な相互接続の標準となりつつあるMHSが、オフィスでの標準的電子メールとしても有望である。MHSを広く利用できるようにするためには移植性の高いOSの上で実現することが課題となる。ここではそのようなOSとしてUNIXを選び、UNIX上で郵政省告示予定の電子メール通信推奨通信方式(以下パソコンMHSと略す)を実現するための処理方式について述べる。

2. UNIXでのMHS実現の問題点

本稿では、ユーザ数が数百オーダの事業所においてテキストメールを提供するMHSシステムを対象として処理方式を検討する。UNIXは移植性の高い標準OSの1つとしてミニコン、ワークステーションに普及しつつある。UNIXの柔軟なファイルシステムと豊富なファイル管理ユーティリティは電

子メールのようなアプリケーションの実現には有効であると考えられる。反面、UNIXでのMHSシステム実現には表1のような問題点がある。以下、処理方式についての検討を報告する。

3. 処理方式

3.1 プロセス構成

ノードにおいては、複数の処理要求を並列に処理する必要がある。複数プロセスの実現方法はいろいろあるが、次のような観点により実現する。

(1) 拡張性

X.409コーディングの他にX.430のキャラクタコーディングが将来的に採用されることを考えると、コーディング処理部分(レイヤ6処理部分)を切り分けて実現することが望ましい。

またMHS P3プロトコルが下位サーバとして利用するRTS(Reliable Transfer Server)は上位にP3以外のアプリケーションを持つことができる。この部分も独立に実現することが望ましい。

またメディア変換処理を行なう部分はシステムのグレードアップとともに拡張し、マルチメディア化に順次対応する必要がある。

(2) 性能

各ユーザに応答性能を保証するため、ノードと端末との通信がそれぞれ干渉することなく処理される必要がある。

また、複数プロセスの並列実行で実現するとした場合には、利用者のパスワード情報などが適切にメモリ上にロードされ、すべての関係プロセスで共有され、必要なファイル入出力を最小とすることが望ましい。

資源管理の効率化のため、RTSを実現する部分、コーディング処理を行う部分を独立したプロセスとし、端末コマンドを受け付ける部分は1プロセスで利用者情報を一括管理し、複数端末からの要求をイベント駆動で受け付けるものとする。さらにファイル管理プロセスを別に設けて、ファイル入出力においても外の端末のサービスが実行できる処理方式が適当である。図1にプロセス構成を示す。

表1 UNIXでのMHSシステム実現の問題点

項番	項目	概要
1	並列なプロセスの実現とプロセス間通信	MHSモデルに基づく処理要素のUNIXプロセスとしての実現方法
2	リカバリの実現 (ファイル書き込み タイミング問題、 リカバリポイント と二重化)	UNIXのファイルシステムで書き込み時の書き込み完了保証がないことへの対処
3	ファイル管理方式	処理情報とUNIXのファイルシステム上での実現方法 消去などクリティカルな部分の処理
4	異常処理	UNIXシステムコールエラー検出時のプロトコル上の処理
5	処理実現の契機	処理契機がインプリメントマターのサービス項目の実現方法
6	最大値制限	MHSの仕様では規定されない値のインプリメント制限(時刻指定の範囲、MTA個数など)

*UNIXはベル研究所の登録商標である。

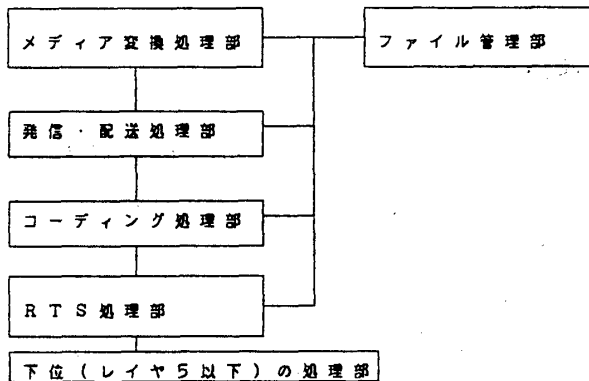


図1 プロセス構成

3.2 リカバリ方式

メールシステムにおいてはメッセージの消失を防ぐことが重要である。UNIXのファイルシステムは決してフォールトトレラントではないので、MHSの実現における対策として次のようなものが考えられる。

- ・ファイルの2重化
- ・管理ファイルの2重化
- ・ディスクデバイスファイルを利用した特殊なファイルシステムの構築
- ・変更履歴の収集

ファイル入出力の増加がシステム性能のネックとなるので、事業所用小規模MHSシステムでは管理情報の2重化が適当である。

MHSではプロトコル上、確認レスポンスを返した時点でメッセージの保証を行う。一方ファイルへの、メッセージの書き込み、管理情報の書き込みはシステム性能のネックとなる。最小の書き込みでリカバリを行うため、発信時エンベロープ情報の書き込み、配送時エンベロープ情報の読みだしを2点のリカバリポイントとして設定し、この前後でのシステム異常はこの2点から復旧し、復旧に当たっては2重配送を許すフェイルセーフ処理方式が望ましい。

3.3 ファイル管理

MHSにおけるメッセージ管理は受付イベント識別子、配送イベント識別子、検索識別子に基づいて行われる。この場合、次の2つの方法がある。

- ・識別子テーブルを作成する
- ・ファイル名を識別子と対応付けて作成する

事業所用メールシステムにおいては、検索識別子を個別管理することにより、識別子をユーザにとって自然な形で提供する方式が望ましい。このようにすることにより端末からのアクセスを可能とする。

一方システム内部でのみ管理する配送中のメッセージについてはイベント識別子をそのままファイル名とし、必要ならば拡張識別子を付加することにより実現する方式とする。これにより大きな管理テーブルの排他制御が不用になり、サービス管理のユーティリティの作成が容易になる。

3.4 消去方式

複数の通信回線にたいしてサービスを行い、それを独立なプロセスで実現した場合、メールボックスにおけるメッセージの出力と消去の競合に関するメッセージ管理方式が問題となる。

- ・読みだしと消去の競合

メールボックスメッセージに関しては同一利用者名による2重ログインの禁止によって対応できる。

- ・通知情報の消去

読みだし後のどの時点で読みだしたメッセージ（特に読みだし後、消去することになっている通知類）をシステムが消去するかが問題となる。下位レイヤがデータ転送の確認を上位に通知しない場合でもシステムが動作するためには次の3つの方法がある。

- ・次の正常なコマンド送信を待つて消去する
- ・ユーザのログオフを待つて消去する
- ・ローカルプロトコルでデータ送信完了を確認するシステムのパフォーマンスを確保するために正常すべてのコマンドを監視するのではなく、ログオフを契機に消去を行う方式では、消去までのタイムラグにおいて消去しても空きエリアが増えないが、処理が簡単で消去要求時にも応答性能が劣化しない。サービス要求と処理能力とのトレードオフで判断することになる。

4. むすび

MHSシステムを汎用OS UNIXの上で構築する上で特に拡張性及び汎用化を考慮した構成法について、検討し、報告した。

参考文献

- [1] CCITT MHS 勧告 X.400シリーズ
- [2] 郵政省告示（予定） 電子メール通信推奨通信方式（案）