

バックエンド方式によるUNIX*: その構成と通信方式

6P-3

岡本 利夫 今井 徹 前田 賢一
(株)東芝 総合研究所

1. はじめに

ここでは、2つのプロセッサをバックエンド方式で結合したシステムに関して報告する。報告者らはAI言語を高速に実行することを主な目的として、AIワークステーションを研究開発してきた[1]。これは、独自に開発したAIプロセッサ(AIP)をエンジニアリングワークステーション(EWS)にバックエンド方式で接続したものである。この方式の利点は、実行時間がかかる推論処理のところのみAIPで高速に実行することであり、その他の入出力等の本質でない処理はホストとなるEWSで行なう。すなわち、AIプログラムの開発環境の向上と実行速度の向上を同時に狙ったものである。

AIPは、ハードウェア構成として、仮想記憶、割り込み処理、例外処理、メモリ管理機能等、1人前のプロセッサとしての機能を持っており、命令もRISC風の汎用命令をサポートしている。

報告者らは、このAIPを土台とし、基本ソフトウェア(OS)を研究しており、その一環としてUNIXの移植を行なった。今回は、その構成と通信方式に関して報告する。

2. システム概要

今回、UNIXを移植したシステムの全体構成を図1に示す。以下ハードウェア構成、ソフトウェア構成に分けて説明する。

2.1 ハードウェア構成

AIPは、ボード3枚が最小構成で、当社のEWSであるAS3000に、バックエンド方式で結合されている。両者はVMEバスによってインターフェー

スされている。AS側からはAIPのイニシャライズ、プログラムのローディングや、実行制御を行なえる。また、AIPの内部状態を観察したり、変更したりできる。

デバッガやI/Oは、この機能を用いて、VME空間を介して処理を行なっている。

2.2 ソフトウェア構成

AIPには周辺装置が直接接続されていない。UNIXをAIP上に移植するには最低限、次の2つの周辺装置が必要となる。

1. コンソール
2. ディスク
(ルートファイルシステム及びスワップ)

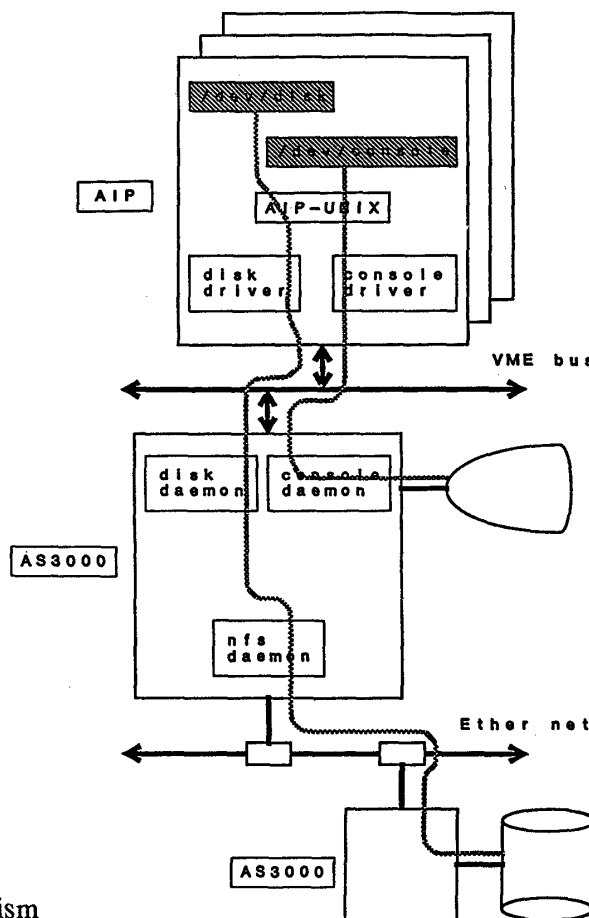


図1 AIP-UNIXの全体構成

* UNIXはATTが開発しライセンスしているOSです

A UNIX System using Backend connection Mechanism
Toshio OKAMOTO, Toru IMAI, Ken-ichi MAEDA
TOSHIBA Corporation

したがって、本システムではVMEバスを介してEWSが上記の2つの周辺装置と見えるように構成されている。コンソールとディスクそれぞれの専用のデーモンプロセスがEWS上に常駐しており、それらがサービスを提供する。

又、デバックには"AIP-dbx"とよぶ、クロス・シンボリックデバッガを用意し、これを用いて効率的に移植を行なった[2]。

3. I/O 方式

2章で説明した通り、AIPには自前の周辺装置がないので、EWSが代行する。現在はディスクとコンソールの2つの周辺装置のみをサポートしている。両者ともVME空間上でデータを受け渡し、割込信号で制御する。

この方式の利点として以下のことが挙げられる。

1. デバイスドライバーが簡単になる
2. デバイスドライバーのデバックが簡単になる
3. ルートファイルシステムを作るのが楽である
4. カーネルデバックが簡単になる

デバックモード(OS開発初期)と通常モードではI/O方式の実現法をかえている。デバックモードでは、デバッガ"AIP-dbx"が両方の周辺装置の機能も合わせて提供する。

通常モードでは、ディスク用、コンソール用のそれぞれのデーモンプロセスをEWS上に動かしておき、それがサービスを行なう。(図2)

以下、ディスクI/Oを例に取って説明する。

AIP-UNIX側とEWS上のAIPディスクデーモン側とのインタフェースは、デバイスドライバーレベルで行なう。

AIP-UNIX内で、ディスクにデータを書きに行く場合、デバイスドライバーのディスクインタフェース部へ制御が渡る。ここでAIP側では、AIP-EWS間インタフェース用のコミュニケーション領域(VME空間上)へ、ディスクへの読み書きに必要なアドレス・サイズ等のデータを積み、割込信号を用いてEWS側へリクエストする。EWS側のディスクデーモンは、この割込みによっ

てEWSのカーネルを経て起動され、EWS側に用意してあるAIP用のルートディレクトリのファイルまたはAIP用のスワップエリアのファイルにアクセスする。

結果は逆ルートでAIP側へ伝える。

ディスクデーモンとコンソールデーモンは、互いに非同期で動いている。そこで、互いに割込みをかけあって制御している。AIP側のメモリはVME空間からすべて見えるので、コミュニケーションエリアには制御情報のみを置き、実際のデータはAS側からアクセスする。

5. おわりに

AIP上のUNIXについて、全体構成、I/O方式を説明した。

今後の研究方向としては、第一にソケット・デバイスをサポートし、NFS、X等の標準的ツールの移植を行なう。また、現在、本アーキテクチャをもとにAIチップを作製しており[3]、これを利用してMach, Spriteに代表される分散・並列OSの研究に発展させていく予定である。

参考文献

- [1] 岡本、他:「バックエンド方式によりプロセッサを結合したAIワークステーション「WINE」の構成法」、電学論C,108,721 (1988.9)
- [2] 今井、他:「AIP-dbxの実現法」、情報処理学会第37回全国大会、4K-6,1988
- [3] 皆川、他:「AIチップ(IP704)のアーキテクチャ、情報処理学会第38回全国大会、4U-1,1989

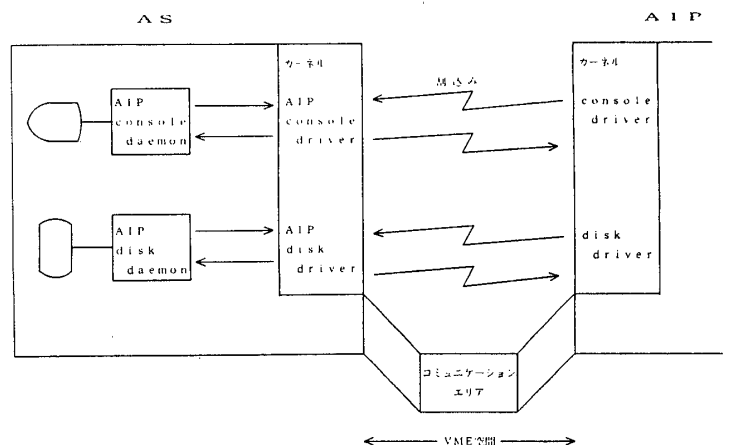


図2 I/O 構成