

ユーザレベル OS のための軽量 VM に対する ネットワーク機能の追加

榮樂 英樹

1 背景

ユーザレベル OS とは、本来は実機上で動作するような OS (Operating System) を、他の OS 上でユーザ・プロセスとして動作可能にしたものである。現在までに部分的なハードウェアのエミュレーションを行う仮想計算機 (軽量 VM) と機械語命令の静的な書き換えによりユーザレベル OS を実現する方法を提案し、実現してきた。実際に NetBSD、および、FreeBSD が Linux 上で動作している。この時、元の OS に対する修正は、アドレス定数や CPU タイプの判定などごく単純なものであった。

2 目的

このプロジェクトの第 1 の目標は、ユーザレベル OS からネットワーク機能を利用可能にすることである。これにより、ユーザレベル OS 上でネットワーク・アプリケーションやサーバを動作させることが可能になる。さらに、第 2 の目標は、性能を改善することである。その他に、既存のソースコードを見直し、ドキュメントを整備することで、オープンソースとして公開し、日本発の仮想計算機ソフトウェアとして広く世界に進出することをめざす。

3 開発の内容

開発したソフトウェアの動作環境を以下に示す。

CPU: Pentium 以降またはその互換 CPU

メモリ: 128MB 以上

ホスト OS (Linux 版): Linux 2.4.18 以降が動作している Linux ディストリビューション

ホスト OS (NetBSD 版): NetBSD 1.6.1。カーネルにパッチを当てて再構築が必要。

3.1 ネットワーク機能の追加

従来のユーザレベル OS でネットワーク機能を利用可能にするには次の 2 つの問題を解決する必要があった。

1. イーサネット等のネットワーク機器をエミュレートする
2. ネットワーク・フレームを、実際の物理的なネットワークに流す

前者は、比較的簡単である。たとえば、User-mode Linux には、擬似的なイーサネット・デバイスのデバイス・ドライバが組み込まれている。外部には、スイッチングハブをエミュレートするような別プログラムがある。しかし、この機能を利用して外部のネットワークとは通信をすることができない。

外部のネットワークとの通信を可能にするには、ユーザレベル OS が出力したネットワーク・フレームを、実際の物理的なネットワークに流す必要がある。User-mode Linux では、この機能も外部のプログラムで提供している。ただし、このプログラムは、Linux が提供している TUN/TAP (tunneling/tapping) と呼ばれる機能に依存している。

この TUN/TAP という機能は、ホスト OS の通信を妨害したり、また、他のコンピュータへの攻撃に利用することができる。したがって、この機能を利用するには、特権利用者 (root) の権限に限定する必要がある。

本軽量 VM では、従来の方法とは異なったアプローチでユーザレベル OS から外部のネットワークをアクセス可能にする。概略を以下に示す (図 1)。

- 送信
- ユーザレベル OS から出力されたネットワーク・フレームを解析し、データリンク層、IP 層、および、TCP 層のヘッダを取り除き、もとの通信データ部分だけを取り出す。
 - 元の通信データを通常のソケット・インタフェースを使ってホスト OS に渡す。
- 送信
- ホスト OS から通常のソケット・インタフェースを通じて通信データを受け取る。
 - 通信データに対して、TCP 層、IP 層、データリンク層などのヘッダを付加し、ネットワーク・フレームを生成し、ユーザレベル OS へ渡す。

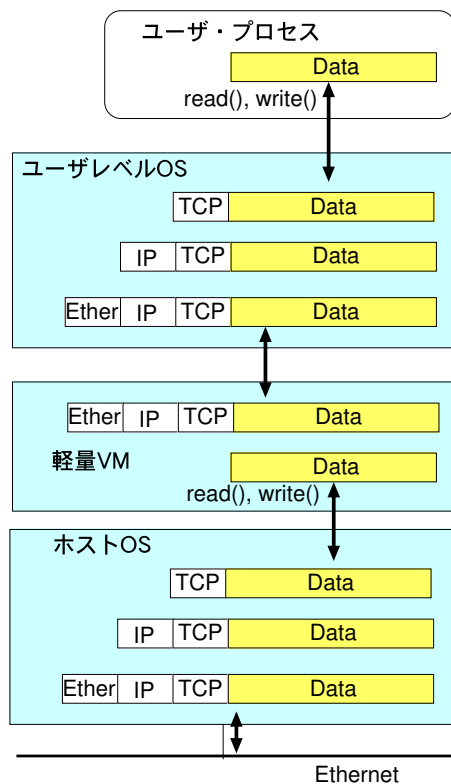


図 1: 軽量 VM における TCP/IP 通信

3.2 性能向上

現在の実現では、MMU の機能をエミュレートする時に頻繁に `mmap()` システム・コールや `munmap()` システム・コールを呼び出す必要がある。この部分を見直し、高速化を計る。場合によっては、高速化のためにカーネルに機能を追加することも検討する。

3.3 ソースプログラムの整理

最後に、ソース・プログラムの見直しを行い、拡張しやすいように書き換える。現在の実装では、短い演習期間内に動作させることを最優先にコーディングを行ったので、デバッグ用のコードや既に不要になったコードがプログラム・ソースに大量に残されている。これを整理し、プログラムの見直しを行うことで、他人にも理解しやすいものにする。そして、広くオープンソースとして公開し、世界中の開発者の支援を得ながらさらなる機能拡張をめざす。

3.4 他の OS への移植

軽量 VM は、もともと Linux 用に開発したものであった。これを、NetBSD で実行できるようにした。この際、NetBSD 1.6.1 には、軽量 VM を実行するのに必要なプロセストレースの機能が足りなかったため、Linux の PTRACE_SYSCALL と互換性のある機能と、特殊なレジスタを読み取るための機能を追加した。

4 従来の技術（または機能）との相違

機械語命令をコンパイル時に書き換えるという点に新規性がある。NetBSD をユーザレベル OS として実行した例は、他には存在しない。

従来の仮想計算機やユーザレベル OS では、ネットワーク機能を実現するために、カーネル・モジュール、または、特権利用者の権限が必要であった。この提案では、カーネル・モジュールも特権利用者の権限もなしにネットワーク機能を実現する所に新規性がある。

5 期待される効果

本システムのようなユーザレベル OS は、既存のオペレーティング・システムの開発にも 2 つの意味で影響を与えると予想される。まず、カーネル自身の開発がユーザレベル OS により加速される。もう 1 つは、ユーザレベル OS の実行を高速化するために、カーネルに支援機能が組み込まれていくことが予想される。

6 普及（または活用）の見通し

プロジェクトの終了後は、オープン・ソース・ソフトウェアとして公開する。機械語命令を書き換えるという手法は、IA-32 以外のアーキテクチャでも利用可能である。よって、ソース・コードを公開した場合、他のアーキテクチャへの本手法が適応され、他のアーキテクチャでも類似の軽量 VM が開発されると思われる。

7 開発者名（所属、e-mail アドレス）

榮樂 英樹（筑波大学第三学群情報学類、 hdk@coins.tsukuba.ac.jp、 hdk.2@users.sourceforge.net）

（参考）開発者 URL

<http://lilyvm.sourceforge.net> で、開発中の軽量 VM 「LilyVM」 の公開を行っている。