

協調型仮想計算機におけるホスト OS のファイルへのアクセス

白石 光隆^{†1} 新城 靖^{†1} 五明 将幸^{†1}
佐藤 聡^{†1} 中井 央^{†2} 板野 肯三^{†1}

1. はじめに

1 台の PC(Personal Computer) にて、複数の OS(Operationg System) のアプリケーションを動作させる方法として、Type II の仮想計算機 (Virtual Machine, VM) を利用する方法がある。従来の仮想計算機は実機や他の VM と相互干渉せず、独立して動作することを目的としている。これに対し、**協調型仮想計算機^{1),2)}** は独立して動作するのではなく、ゲスト OS とホスト OS が協調して動作する VM である。これにより、ゲスト OS 側からネットワーク処理を介さずホスト OS のプログラムとファイルを利用でき、さらに両 OS の機能を組み合わせた処理が可能となる。

ゲスト OS 側で動作するプログラムは協調型仮想計算機が提供するホストコールを用いて、ホスト OS のプログラムを実行したり、ファイル操作を行うことができる。しかし、ホストコールを利用するためには、ゲスト OS 側のプログラムを修正する必要がある。ホスト OS 側のプログラムとファイル利用の度に、ゲスト OS 側のプログラムを逐一修正するのは手間がかかる。本稿では、この問題を解決する機能として、ホスト OS 側のファイル操作を行うためのファイルシステムである **hostfs** とホスト OS 側のプログラムを実行するためのシステムソフトウェアである **二重 OS シェル** について述べる。

2. 協調型仮想計算機

協調型仮想計算機はゲスト OS 側のプロセスに対し、次の 3 つの機能を提供する。

ホストコール: これは、ゲスト OS 側からホスト OS 側のシステムコールを実行するためのインタフェースである。ゲスト OS 側のプロセスは、ホストコールを利用してホスト OS 側のファイル操作とプログラムの実行を行う。

シャドウ・プロセス: ホストコールが行われるとホスト OS 側で生成され、実際にシステムコールの実行を行うプロセスである。シャドウ・プロセスはカレントワーキングディレクトリなど、ホスト OS 側における実行コンテキストの保持を行う。

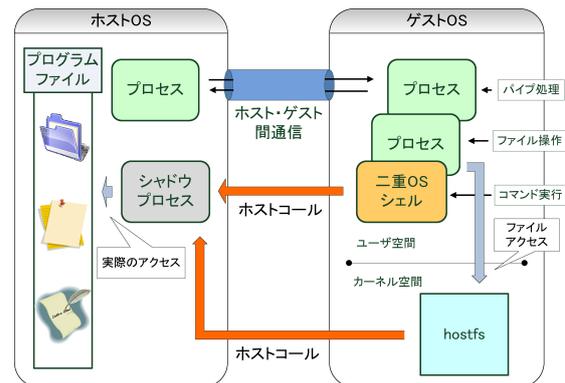


図 1 ホスト OS のプログラム・ファイル利用における制御の流れ

ホスト・ゲスト間通信: ホスト・ゲスト OS 間にまたがる、UNIX のパイプと同様のプロセス間通信機能である。

本稿では、これらの機能を用いて **hostfs** と **二重 OS シェル** を実現する。

3. hostfs

利用者はホストコールを使用することで、ホスト OS 側のファイル进行操作することができる。しかし、ゲスト OS 側のプログラムの実行結果をホスト OS 側のファイルに出力する場合、ホストコールを使用して **open** システムコールなどをホスト OS 側で実行するように、ゲスト OS 側のプログラムを修正する必要がある。これに対し、ホスト OS 側のファイルシステムをゲスト OS のディレクトリ木構造から見えるようにする機能があれば、ゲスト OS 側のプログラムの修正を行う必要なくゲスト OS 側のプログラムとホスト OS 側のファイルを組み合わせられた処理ができる。このような機能を提供するファイルシステムとして、UML(User Mode Linux)⁵⁾ の **hostfs**(Host Filesystem) がある。本研究でも同様に **hostfs** を提供する (図 1)。

本 **hostfs** は、ホスト OS 側のファイルシステムをゲスト OS 側にマウントするファイルシステムである。利用者はマウントポイント以下に存在するホスト OS 側のファイルを指定することで、ホスト OS 側のファイルを開いて読み書きしたり、リダイレクションを利

^{†1} 筑波大学システム情報工学研究科
^{†2} 筑波大学図書館情報メディア研究科

用できるようになる。

```
例: cat guest.txt >> /mnt/home/hoge/host.txt
```

上記の例では、利用者がマウントポイント/mnt からホスト OS 側のファイル/home/hoge/host.txt に対し、ゲスト OS 側のファイル内容を追加している。

hostfs は FUSE(Filesystem in Userspace)⁴⁾ を用いて実装している。FUSE を利用すれば、ファイルシステムの内部やカーネルモジュールプログラミングの知識を必要とせず、ファイルシステムを開発できる。ファイルシステムを開発する際、FUSE のカーネルモジュールから呼び出される関数を実装している。この実装する関数にて、ホストコールを使用することでホスト OS 側のファイルを操作する。ホスト OS 側のファイル情報を取得する場合、関数 getattr にてホストコールを実行し、stat システムコールをホスト OS 側で実行する。これにより、ホスト OS 側のファイル情報を取得した後、ゲスト OS 側にこれを渡す。

4. 二重 OS シェル

利用者がホストコールを利用するためには、ゲスト OS 側のプログラムを修正する必要がある。しかし、ゲスト OS 側のプログラムを逐一修正するのは手間がかかる。そこで本研究では、ゲスト OS 側のシェルのみを修正することで、他のゲスト OS 側のプログラムの修正を行わずに、ホスト OS 側のプログラムを実行することを可能にする。このホストコールを用いたシェルの二重 OS シェルと呼ぶ (図 1)。二重 OS シェルは、ゲスト OS 側の利用者がゲスト・ホスト両 OS 側のプログラムを実行・制御するためのシステムソフトウェアである。

ホスト OS 側にてコマンドを実行する場合は、コマンド `hrun` を使用する。コマンド `hrun` は二重 OS シェルの組み込みコマンドである。引数として与えられたコマンドをホスト OS 側にて実行するコマンドである。

```
例: hrun ls -l
```

上記のコマンド例の場合、コマンド `"ls -l"` がホスト OS 側で実行される。`hrun` を使用しない場合、コマンドはゲスト OS 側にて実行される。

二重 OS シェルは、ゲスト・ホスト両 OS 側にカレントワーキングディレクトリを保持する。ホスト OS 側のカレントワーキングディレクトリの移動には、コマンド `hcd` を使用する。

```
例: hcd ../../workspace
```

上記の例では、ホスト OS 側のカレントワーキングディレクトリを相対パス指定で移動している。

利用者はホスト・ゲスト間通信を利用することで、ホスト・ゲスト間に跨がるパイプ処理が可能である。

```
例: grep hoge guest.txt | hrun lpr
```

上記の例では、ゲスト OS 側のファイルをゲスト OS

側で操作し、その出力をホスト OS 側に渡して印刷している。

現在、二重 OS シェルは `bash`(Bourne-Again Shell)³⁾ を拡張して実装している。また、ホスト OS 側でのコマンド実行はホストコールを使用している。コマンド `hrun`、及びコマンド `hcd` は `bash` の組み込みコマンドの 1 つとして実装している。`hrun` はホストコールによって、ホスト OS 側でシステムコール `exec` を実行することでコマンド実行を実現する。`hcd` でも同様に、ホスト OS 側でシステムコール `chdir` を実行することで、ホスト OS 側のカレントワーキングディレクトリの移動を実現する。

5. 関連研究

UML における `hostfs` は本稿にて提案した `hostfs` と同様に、ゲスト OS からホスト OS 側のファイルを操作するための機能である。しかし、対応している OS に関して UML は Linux のみでしか動作しない。これに対し、今回提案した `hostfs` は FUSE にて実装しているため、FUSE に対応する OS ならばどれでも動作する。

6. おわりに

本稿では、協調型仮想計算機にてホスト OS のプログラムとファイルを利用するための機能である `hostfs` と二重 OS シェルについて述べた。今後は提案した機能を実装し、その機能と性能を評価する。

参考文献

- 1) 五明将幸, 新城靖, 白石光隆, 板野肯三, 佐藤聡, 中井央. 協調型仮想計算機におけるホスト呼び出し機構. 情報処理学会第 70 回全国大会 (2007).
- 2) 白石光隆, 新城靖, 五明将幸, 板野肯三, 佐藤聡, 中井央: 協調型仮想計算機のための二重 OS シェル, 情報処理学会第 70 回全国大会 (2007).
- 3) Chet Ramey. Bash Reference Manual-Reference Documentation for Bash Edition 3.2, for Bash Version 3.2. (2006).
- 4) Miklos Szeredi. File system in user space (FUSE). <http://fuse.sourceforge.net/>.(2008/11/3)
- 5) Jeff Diked. User-mode Linux, Atlanta Linux Showcase Proceedings of the 5th annual Linux Showcase & Conference, Vol.5, pp.2 (2001).