

協調型仮想計算機のための二重 OS シェル

白石光隆* 新城靖†,‡ 五明将幸† 佐藤聡† 中井央§ 板野肯三†,‡

筑波大学† 科学技術振興機構‡

1 はじめに

協調型仮想計算機とはゲスト OS (Operating System) とホスト OS を隔離せず、ゲスト OS からホスト OS の資源を使う仮想計算機である [1]。本稿では、その協調型仮想計算機上で動作する二重 OS プログラムの 1 つとして、2 つの OS の資源を扱うことが可能なシェルについて述べる。このシェルを二重 OS シェルと呼ぶ。

2 二重 OS シェル

協調型仮想計算機ではゲスト OS からホスト OS のシステムコールやライブラリルーチンを呼び出すことが可能である。二重 OS シェルはこの機能により、ホスト OS およびゲスト OS の資源同時利用を実現する。二重 OS シェルの動作を図 1 に示す。二重 OS シェルはゲスト OS 側で動作する。

二重 OS シェルは以下の機能を提供する。

- ゲスト OS 上でのコマンドの実行
- ホスト OS 上でのコマンドの実行
- ゲスト OS の環境の設定
- ホスト OS の環境の設定
- 入出力の切り換え
 - 標準入出力 (キーボード、ディスプレイ)
 - ゲスト OS 側のファイル
 - ホスト OS 側のファイル
 - ゲスト OS 内のパイプ
 - ホスト OS 内のパイプ
 - ゲスト OS とホスト OS 間のパイプ

二重 OS シェルの文法では、ホストの資源を利用したい場合はコマンドやファイルの先頭に記号"!"を付加する。以下に例を示す。

```
cd パス名 1 ; !cd パス名 2
```

このコマンドを実行すると、ゲスト OS 側とホスト OS 側のカレントワーキングディレクトリがそれぞれパス名 1、パス名 2 の示す場所に切り替わる。

ゲスト OS 上からホスト OS の環境変数を変える場合、以下のコマンドを実行する。

```
!export 環境変数名 = 値
```

ゲスト OS でのコマンドの出力をホスト OS のコマンド入力

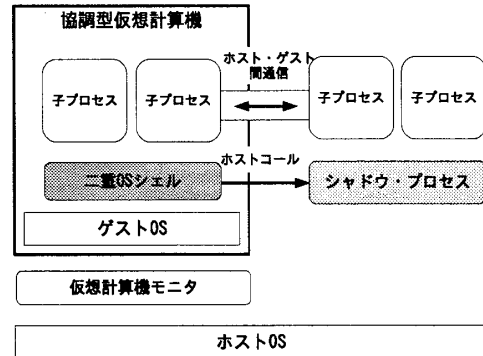


図 1 二重 OS シェルの動作

にすることもできる。

```
a2ps -o guestprog.c | !lpr
```

このコマンドを実行すると、ゲスト OS 側にあるファイル guestprog.c が a2ps コマンドによってポストスクリプトに変換される。出力結果はホスト・ゲスト間にまたがるパイプを通じてホスト OS 側に送られる。ホスト OS 側では受けとった結果を lpr コマンドによってプリンタに出力する。ホスト OS 側のファイルを使用して、ゲスト OS 上でコマンドを実行することもできる。

```
cat < !hostfile.txt
```

このコマンドを実行すると、ホスト OS 側のファイル hostfile.txt の内容が表示される。

3 実装

ユーザが二重 OS シェルを起動すると、二重 OS シェルは自分に対応したシャドウ・プロセスをホスト OS 上に作成する。ユーザはゲスト OS 側から二重 OS シェルにコマンドを与える。二重 OS シェルは特に何も指定されなかった場合、コマンドを自分自身 (ゲスト OS) で実行する。ユーザがコマンドをホスト OS で実行することを指示した場合、二重 OS シェルはホストコールを用いてシャドウ・プロセスに送り、シャドウ・プロセスはこれを実行する。

ホスト OS 側のシャドウ・プロセスはゲスト OS 側からのホストコールにより、指示を受け取る。受け取ったシャドウ・プロセスは指示されたコマンドを実行する。二重 OS シェルのインスタンスごとに対応したシャドウ・プロセスを生成することで、ホスト OS 側は複数の二重 OS シェルからのホストコールに対応できる。

二重 OS シェルはホスト・ゲスト間にまたがったパイプ処理を、協調型仮想計算機が提供しているホスト・ゲスト間通

Dual OS Shell for Host OS call on Cooperative Virtual Machine
Mitsutaka SHIRAISHI, Yasushi SHINJO, Masayuki GOMYO,
Akira SATO, Hisashi NAKAI, Kouzo ITANO

† University of Tsukuba

‡ Japan Science and Technology Agency

信で実現する。これにより、二重 OS シェルはゲスト OS 上のプロセスとホスト OS 上のプロセスを協調させて実行することが可能となる。

4 RPCによる実装

協調型仮想計算機は実装途中のため、現段階では使用できない。本研究ではRPC(Remote Procedure Call)および、socket APIによるTCP/IP通信によって、二重 OS シェルの動作に必要な協調型仮想計算機の機能を代替する。

4.1 ディスプレイサーバとキーボードサーバ

ディスプレイサーバはゲスト OS 上で動作し、ホスト OS 上で実行されたコマンドの結果を受け取り、画面に出力するサーバである。キーボードサーバはゲスト OS 上で動作し、ユーザからの入力をホスト OS 上に送るサーバである。例として次のコマンド"!wc"を実行することを考える。二重 OS シェルはwcコマンドの標準入力を、TCP/IPでキーボードサーバに接続する。また、標準出力をコンソールサーバに接続する。

4.2 RPCサーバ

RPCサーバはゲスト OS から送られる要求に従ってホスト OS のシステムコールを実行する。RPCでは、ゲスト OS 側からホスト OS 側へのシステムコールの回数とプロセス ID の送信、ホスト OS 側からゲスト OS 側へのシステムコール実行の成否を送信する。現段階においてRPCで行うシステムコールを表1に示す。

コマンド"`command1 | !command2`"を実行する例を用いて、二重 OS シェルの動作を説明する。最初にゲスト OS 側からホスト OS 側に、RPCでforkの実行を指示する。ホスト OS 側ではforkによって、シャドウ・プロセスを作成する。ゲスト OS 側でcommand1を実行するための子プロセスを作成する。これを子プロセス1とする。子プロセス1はsocket、bind、getsockname、listenを実行する。これにより子プロセス1は通信確立に使用するポート番号を取得する。子プロセス1はポート番号を親プロセスに渡した後、acceptを実行し通信要求が来るまで待機状態になる。

ゲスト OS 側の親プロセスは最初、RPCによってホスト OS 側のシャドウ・プロセスにforkの実行を指示する。これによりホスト OS 上でcommand2を実行するための子プロセスが作成される。この子プロセスを子プロセス2とする。

子プロセス2を作成した後、ゲスト OS 側の親は子プロセス1からポート番号を受け取るまで待機する。ポート番号を受け取ったら、RPCによってホスト OS 側のシャドウ・プロセスに、ポート番号と通信要求の実行指示を送る。ホスト OS 側ではシャドウ・プロセスがこれらを受け取り、子プロセス2に対してsocket、connectによる接続要求を行うことを指

示する。子プロセス2は子プロセス1に対して接続要求を行う。待機していた子プロセス1がこれを受けとることで、子プロセス1と2の間でプロセス間通信が確立される。

この後、子プロセス1および2で標準入出力の切り替えを行う。子プロセス2の標準入出力切り替えは、ゲスト OS 側の親プロセスがRPCによって指示する。子プロセス2の標準出力はディスプレイサーバに接続する。入出力切り替えが終了したら、子プロセス1はcommand1を、子プロセス2はcommand2を実行する。

5 二重 OS シェルの利用方法

二重 OS シェルを利用することでゲスト OS の管理が容易になる。例として、二重 OS シェルではホスト OS 上でゲスト OS のファイル内容をプリンタに出力できることがあげられる。これにより、ゲスト OS ではプリンタの設定が必要なくなる。同様に、電子メールの設定などが不要になるなどの点もあげられる。このような点から、二重 OS シェルはゲスト OS の管理が容易になる。

6 関連研究

別のコンピュータでコマンドを実行する方法として、RSH(Remote Shell)、SSH(Secure Shell)、onコマンドがあげられる。これらはリモートホスト上でコマンド実行する機能を提供するが、onコマンドはさらにユーザの起動側のカレントワーキングディレクトリと環境変数を保持することができる[2]。

二重 OS シェルはRSH、SSH、onコマンドと比べて、次の点が異なる。

- ネットワークを介さず、実機内でホスト OS の資源にアクセスする。
- 構文に関して、ホスト名を指定する必要がない。また、ホスト・ゲスト OS 両方で同じシェルの文法が利用できる。
- ホスト・ゲスト間でのパイプを提供する。
- 二重 OS シェルはゲスト OS、ホスト OS それぞれの実行環境であるカレントワーキングディレクトリと環境変数を保持することができる。

7 おわりに

本稿では協調型仮想計算機上で動作する二重 OS シェルについて述べた。二重 OS シェルは協調型仮想計算機の機能を利用することで、ゲスト OS とホスト OS 両方の資源を扱うことが可能になる。今後は協調型仮想計算機上にて実際に二重 OS シェルを動作させ、その機能を評価する。

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] Sun Microsystems. *Solaris 10 Reference Manual Collection*, 2007.

表1 RPCで行うシステムコールの一覧

使用目的	システムコール
シャドウ・プロセス生成	fork
ホスト・ゲスト間通信	socket,connect,bind getsockname,listen,accept
標準入出力切り替え	open,close,dup2
コマンド実行	exec