

分散環境でのファイル名前空間とユニオンディレクトリについて

4H-5 岡坂 史紀 中野 隆裕 清水 謙多郎 芦原 評

電気通信大学情報工学科

1. はじめに

同じディレクトリパス名で複数のファイルシステム内のファイルアクセスを可能にするユニオンディレクトリの機構は、ソフトウェア開発環境の構築にあたって 3DFS[2]や Translucent FS[3]として実現された。Plan 9[1]もこの機構を採用し分散環境に適用している。

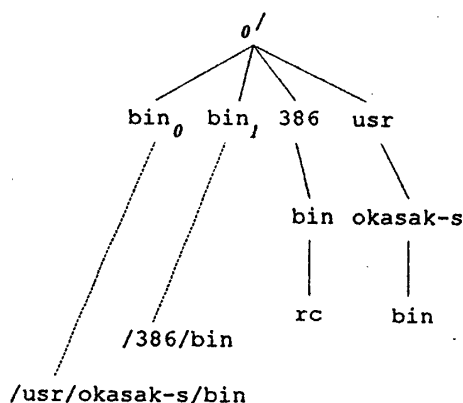


図1 名前空間の例

図1にユニオンディレクトリを含むファイル名前空間の例を示す。/bin はふたつのディレクトリ/386/bin と/usr/okasak-s/bin のユニオンになっている(ディレクトリ名の右下の数字の意味は次節で述べる)。利用者はどちらのディレクトリにあるファイルも/bin によってアクセスすることができる。ユニオンディレクトリは、Plan 9を例にすれば、次のようにコマンド行から入力することによって構成できる。

```
% bind /386/bin /bin
% bind -b /usr/okasak-s/bin /bin
```

“File Name Space with Union Directories in Distributed Environment”

Shiki Okasaka, Takahiro Nakano, Kentaro Shimizu, Hyo Ashihara

The University of Electro-Communications  
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182, Japan

本発表では、ユニオンディレクトリのパス名評価の困難な点を示しながら、筆者らが開発中の007およびAct 8[4]オペレーティングシステムで採用しているパス名評価のアルゴリズムとその実装について報告する。

2. パス名評価

ユニオンディレクトリのある名前空間では、同じ絶対パス名をもつファイルが複数存在し得る。与えられたパス名に対応するファイルを名前空間から選択するための基本規則を次のように定める：

規則 1. 与えられた絶対パス名から選択されるファイルは、ファイルツリーを深さ優先順で探索し最初にマッチしたファイルである。

木の深さ優先探索では、根から再帰的にディレクトリ項目を左から右へ子を探索するものとする。

ユニオンディレクトリ内の任意のファイルを指定するには、次のようなパス名表記を用いる：

$${}_0/e_{1n_1}/e_{2n_2}/\dots/e_{nn_n}$$

$e_{in_i}$  はユニオンディレクトリ  $e_i$  において左から  $n_i+1$  番目のディレクトリを指す。指示子  $n_0$  は、左から  $n_0+1$  番目のルートディレクトリを指す。指示子  $n_i$  を省略した場合は、暗黙にもっとも左に位置するディレクトリを指すものとする。

2.1. “..”による親ディレクトリの評価

ユニオンディレクトリのある名前空間では、親ディレクトリの概念も異なる。例えば、 ${}_0/bin_1/rc_0/..$  は  ${}_0/bin_1$  ではなく  ${}_0/bin_0$  に評価されなければならない(図1参照)。

絶対パス名のすべてのパス要素の指示子が 0 の場合、そのディレクトリは「名前空間の手前にある」という。パス名評価の2番目の規則は：

規則 2. “..”で移動したら、手前にあるもっとも近くのディレクトリまで戻り、評価しなおす。

手前にあるもっとも近くのディレクトリへ戻るには、ディレクトリが名前空間の手前に来るまで次の手順を繰り返す:

- 1) 作業ディレクトリがユニオンディレクトリで、かつ、指示子が0でなければ、指示子0のディレクトリへ移動する。
- 2) そうでなければ、作業ディレクトリの親へ移動する。

この手順を実行すると未評価のパス要素が残る場合がある。例えば、`_0/bin_1/rc_0/..rc`はいったん`_0/bin_0`に評価され、“rc”を残す。その後、規則 1により、`_0/bin_1/rc_0`に評価される。無駄に見えるが、もしここで`/user/okasak-s/bin/rc`が作成されていれば、`_0/bin_1/rc_0/..rc`は`_0/bin_0/rc_0`に評価されなければならない。

## 2.2. 相対パスによるファイル名評価

プロセスの現ディレクトリからの相対パスによるファイルの指定は便利な機能であるが、やはり注意が必要である。例えば、プロセスの現ディレクトリが`_0/bin_1/rc_0`であると仮定する。現ディレクトリが`_0/bin_1/rc_0`へ移った後でディレクトリ`_0/bin_0/rc_0`が作成されていて、パス名を`_0/bin_1/rc_0`から評価した場合、規則 1 が破られる可能性がある。パス名評価の 3 番目の規則は:

規則 3. 相対パス名を評価するときは、まず、手前にあるもっとも近くのディレクトリまで戻ってから評価を行う。

この例では、パス名評価の作業ディレクトリはいったん`_0/bin_0`に戻り、“rc”を残す。続いて、“rc”と相対パス名の連結が`_0/bin_0`から評価される。もし`_0/bin_0/rc_0`が存在していれば、現ディレクトリもそちらに移動する。

## 3. 実装

ユニオンディレクトリは簡単なマウント表に

よって管理することができる。マウント表は以下の情報を含んでいる。

- マウント地点のものと作業ハンドル
  - マウントされた作業ハンドルの線形リスト
- 作業ハンドルはファイルもしくはディレクトリへのポインタであり、さらに最後に通過したマウント地点へのポインタを保持している。どのディレクトリからも、作業ハンドル→マウント地点→マウント地点のものと作業ハンドル→...という順序で最終的には名前空間のルートに行き着く。これによって、ディレクトリが名前空間の手前にあるかどうか調べたり、“..”で論理的な親ディレクトリへ戻ることが可能になる。なお、ユニオンディレクトリの名前変更はマウント地点のものと作業ハンドルに対して行われる。

利用者が名前空間の手前にあるディレクトリ内で作業している場合、ファイルを開くコストはユニオンディレクトリがないシステムと比べて特に差はない。ディレクトリの一覧を見る場合には名前空間で同じ絶対パス名をもつディレクトリすべてにアクセスが行われる。

## 4. まとめ

本稿ではユニオンディレクトリのある名前空間においてパス名を評価するための 3 つの規則とその実装方法について述べた。

## 参考文献

- [1] D. Presotto, R. Pike, K. Thompson, H. Trickey, “Plan 9, A Distributed System,” *Proc. of the Spring 1991 EurOpen Conf.*, May, 1991.
- [2] D. Korn, E. Krell, “A New Dimension for the Unix® File System,” *Software-Practice and Experience*, Vol. 20(S1), June, 1990.
- [3] D. Hendriks, “A filesystem for software development”, *Proc. of the Summer 1990 USENIX Conf.*
- [4] 岡坂, 清水, “対称型マルチプロセッサ用実時間カーネルの設計と実現について,” 第 50 回情処全大 2H-5, 1995.