

マイクロカーネルLavenderにおけるネームサーバの構築

2F-2

青木 英郎 † 佐脇 秀登 † 毛利 公一 † 大久保 英嗣 ††
 †立命館大学大学院理工学研究科 ††立命館大学理工学部情報学科

1 はじめに

我々は、機能拡張が容易で、かつ、プロセス間手続き呼出しのオーバーヘッドが少ないマイクロカーネルLavenderを開発している。Lavender[1]では、カーネルとシステムサーバが、ユーザに対してさまざまな資源を提供する。しかし、資源は、その種類によって単純な数値や複雑なデータ構造によって表わされる種々の識別子を使用するため、ユーザには扱いにくい。また、識別子の値は、あらかじめ定められる静的な値ではなく、資源の割当てを行ったときにシステムから与えられる動的な値である。このため、プロセスが資源を利用するには、その識別子を特定しなければならない。しかし、プロセス自体が特定することは困難である。この問題は、資源をユーザの定義した名前で扱うための名前管理機構を用意することで解決できる[2]。プロセスに資源を公開するために、名前管理機構へ識別子と資源の名前を登録する。資源を利用するプロセスは、利用する資源の名前を知っていれば、名前管理機構に問い合わせを行うことで、識別子の特定が可能になる。Lavenderでは、ネームサーバが名前管理機構の役割を果たす。

以下、本稿では、2章でLavenderの概要、3章でネームサーバの構成、4章で名前空間の利用について述べ、5章でまとめを述べる。

2 Lavenderの概要

マイクロカーネルLavenderは、メモリ、プロセス、スレッド、プロセス間通信、割込みに関する機能を持つ。その他の機能およびカーネルが持つ機能の拡張は、システムサーバとして実現する。マイクロカーネル方式のOSでは、その構造からプロセス間の協調作業が頻繁に発生し、システムの性能低下を招くという問題がある。この問題を軽減するために、Lavenderは、プロセス間の手続き呼出しを効率的に行う機構を備えている。

Lavenderは、モジュールという機構を持ち、異なるアドレス空間に存在するメモリ資源を共有できる。プロセスは、モジュールを用いて資源を共有することによって、オーバーヘッドの伴うメッセージ通信を用いることなく資

源を参照することが可能になる。

この効率化を行うためには、資源の特定とシステム情報の取得を必要とする。Lavenderでは、ネームサーバに格納された資源の情報が、これらの機構の実現に使用される。

3 ネームサーバの構成

ネームサーバは、初期化部、通信部、要求解析部、要求実行部、データベースから構成されている(図1参照)。初期化部は、ネームサーバ起動時に実行され、名前管理に必要なデータベースを作成し、通信ポートを確保する。通信部は、メッセージ通信でネームサーバを用いる際に呼び出され、メッセージの送受信を行う。要求解析部は、通信部が取得した要求を解析し、要求を実行するための手続きを呼び出す。要求実行部は、データベースを操作する関数群から構成されている。要求実行部は、Lavenderのモジュール機構によって、直接呼び出すことが可能である。また、分散環境では、以上の構成の他に位置透過性実現部が加わる。これは、一定の時間ごとに他ノードと通信を行い、大域的な名前空間の管理を行う。

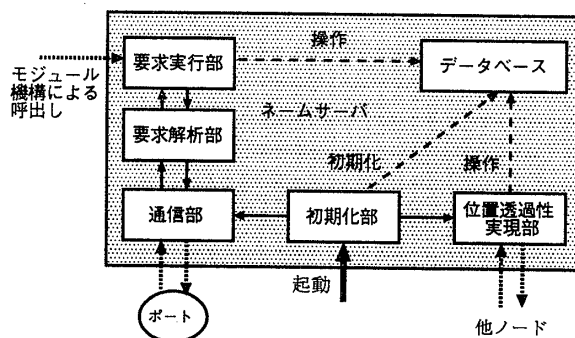


図1: ネームサーバの構成

Lavenderのネームサーバは、以下に示す4つの特徴を持っている。

(1) 名前空間の提供

ネームサーバは、名前空間と呼ばれる、階層を持った名前を使用して名前管理を行う。ネームサーバには、資源に関する情報とそれを指し示す名前が登録される。情報は、数値でも複雑なデータ構造でもよく、ネームサーバはその内容に関知しない。

システムサーバは、ネームサーバの名前空間を使用して資源の名前付けを行うことができる。システムサーバ

An Implementation of Name Server on Lavender Micro Kernel
 Hideo Aoki †, Hideto Sawaki †, Koichi Mouri † and Eiji Okubo ††
 †Graduate School of Science and Engineering, Ritsumeikan University
 ††Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

は、独自に名前管理機構を実装する必要がないため、開発の効率化が期待できる。

(2) 名前管理方式

ネームサーバは、3階層の名前管理方式[3]を用いて名前管理を行う。本方式では、外部名、固有識別子、物理的実体を用いて管理する。ネームサーバは、外部名から固有識別子に変換するネーミング情報、固有識別子から物理的実体を導くロケータ情報、固有識別子から物理的実体を導くロケータ情報を管理する。外部名は、ユーザが資源につけた名前、物理的実体は資源の情報である。固有識別子は、資源の一意性を保証するためにネームサーバ内で用いる。

(3) 位置透過性の保証

分散環境では、資源の位置透過性が問題となる。Lavenderでは、ネームサーバが資源の位置透過性を提供する。各ノードに配置されたネームサーバが協調することで、システム上で一つの大域的な名前空間をつくる。資源は、名前に位置情報を含むことなく、システム内のどの位置からも特定することができる。

(4) サービスの効率的な実行

ネームサーバの機能を用いるには、モジュール機構による呼出しと通信ポートを用いたメッセージ通信の2つの方法がある。これらの2つの方法は、ネームサーバを利用する頻度によって使い分ける。ネームサーバを頻繁に呼び出すシステムサーバは、モジュールを使用する。モジュールは、最初の呼出しにはメモリ資源の共有操作が発生するのでオーバーヘッドが大きい、2回目以降の呼出しは手続き呼出しとなるため高速に実行できる。ネームサーバに一度問い合わせただけでよいプロセスは、メッセージ通信を利用する。

4 名前空間の利用

Lavenderでは、プロセスとプロセスグループを基本とした名前空間の利用を行う。プロセスグループは、協調作業を必要とするプロセスの論理的な集まりである。プロセスの名前空間には、プロセスが所有している資源のなかで、他のプロセスやプロセスグループに公開する資源が登録される。プロセスグループの名前空間には、グループに所属するプロセスの資源や、グループとして確保した資源が登録される。また、名前空間全体は、カーネルやシステムサーバが用いるシステム名前空間と、ユーザプロセスが用いるユーザ名前空間がある。このポリシーに基づく名前空間の利用例を図2に示す。図2は、ユーザ名前空間に3つのプロセスと2つのプロセスグループが存在する。プロセスグループ p-group1は、プロセス proc1と proc2が所属しており、プロセスグループ p-group2は、プロセス proc1と proc3が所属している。このときプロセスグループの名前空間には、所属しているプロセスの名前空間が加えられ、所属しているプロセスが持つ資源の参照が可能となっている。

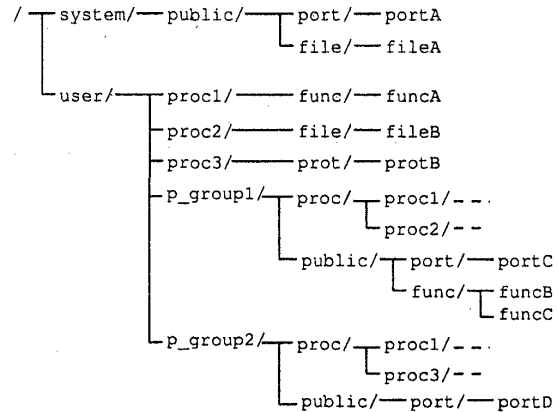


図 2: 名前空間

名前空間は、作成時に参照に関する設定が行われる。これにより、必要のないプロセスが名前空間を参照するのを防ぐ。プロセスグループの名前空間は、グループに属さないプロセスが共有資源の名前を参照できないよう設定される。システムのサービスを呼び出すための情報は、どのプロセスからも参照できるように設定される。図2においては、/system/publicがそれにあたる。

5 おわりに

本論文では、Lavenderにおけるネームサーバの構築について述べた。ネームサーバは、Lavenderが提供する資源を名前で扱う手段をユーザに与える。また、プロセス間手続き呼出しのオーバーヘッドを軽減させる種々の機構を運用するための情報を保持する。資源を提供するシステムサーバは、ネームサーバを用いることで、資源への名前付けと位置透過性の保証を行うことができるため、開発の効率化が期待できる。また、本稿では、ネームサーバで用いる名前空間の利用法についても提案した。現在、ネームサーバは、プロトタイプの実装が行われている。今後の課題としては、障害透過性、分散環境時のスケラビリティに対する検討が挙げられる。

参考文献

- [1] 芝 公仁, 佐脇 秀登, 豊岡 明, 毛利 公一, 大久保 英嗣, “マイクロカーネルLavenderの構成,” 情報処理学会研究報告 97-OS-75, pp. 7-12, 情報処理学会 (1997).
- [2] Sanjay R. Radia, Michael N. Nelson, and Michael L. Powell, “The Spring Name Service,” Sun Microsystems Laboratories Technical Report SMLI-93-16, October(1993).
- [3] 清水謙多郎, 前川守, 芦原評, “分散 OS Galaxy,” 情報処理, Vol. 36, No. 8, pp. 702-707 (1995).