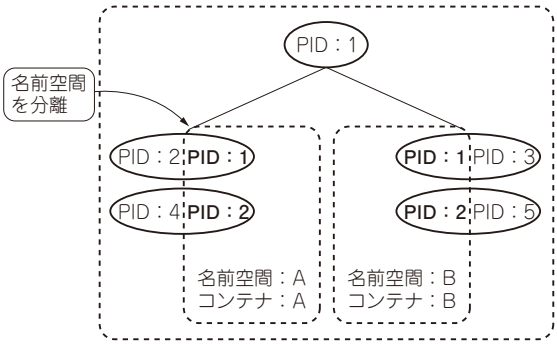
第3章 名前空間/コントロール・グループ/ストレージ・ドライバ/ネットワークへの参加

コンテナを実現するために使われている技術

土屋 健

表1　名前空間で分離できるリソース

名前空間	対象リソース	用　途
PID	PID，プロセス・ツリー情報	プロセスの分離
USER	UID/GID，権限などの管理情報	ユーザ・アカウントの分離
MOUNT	ファイル・システムのマウント情報	ファイルの保存領域の分離
UTS	ホスト名，ドメイン名	ホスト名の分離
NET	NICやネットワーク・スタック	ネットワーク・アドレス，デバイスなどの分離
IPC	System V IPC オブジェクト，POSIX メッセージ・キュー	プロセス間通信の分離



PID空間の分離の例
- 黒文字のPIDはホストで見えているプロセスID
- 太字のPIDがコンテナ内で見えているID
- コンテナAとコンテナBには同じPIDが存在するが実態は別のプロセス

図1　名前空間のイメージ

コンテナとは，OSレベルでアプリケーションの実行環境を隔離する技術です．

コンテナは次の複数の基盤技術を組み合わせて機能を実現しています．Linuxが標準で持つ，

(1) 名前空間（namespace）
(2) コントロール・グループ（cgroup）

という機能です．その他に持つ，

(3) コンテナに見せるストレージを提供する
(4) コンテナをネットワークに参加させる

機能も必要です．

この中で，(1)，(2) の機能はコンテナの基礎となるものですが，(3)，(4) のストレージやネットワークについてはさまざまな実現方法があり，ディストリビューションやDockerのバージョンによって使われる技術が変わったりします．

名前空間（namespace）

前に述べたように，コンテナとはホストとは隔離されたプログラムの実行環境です．

サーバのリソースには，

- CPU　- メモリ　- ファイル　- ネットワーク
- プロセス情報　- ユーザ情報

などがあります．

これらはグローバルなリソースで，みんなで共有されますので，これらを他の環境と隔離することで分離を実現します．CPUやメモリについてはOSレベルで分離されるように制御されていますので，他のリソースの分離を考えます．

リソースの分離のために導入されたのが名前空間（namespace）というものです．それぞれのリソースを，名前を付けたある領域に所属させることで，異なる名前を持つ領域間ではお互いが見えないように隔離する考え方です．

表1に名前空間で分離できるリソースを示します．

名前空間の分離のイメージを**図1**に示します．プロセス空間の分離のイメージ図になります．

コントロール・グループ（cgroup）

サーバ・リソースをコンテナ間で分離する仕組みは，主にソフトウェア・リソースの分離です．

それ以外にハードウェア・リソースの利用を制限する仕組みも必要です．同じハードウェアを共有するので，制限をかけないと一部のアプリケーションだけがリソースを食い潰してしまいます．