

A9N Manual

Version 0.0.1

Contents

1. Introduction	3
1.1. about A9N	3
2. Capability Node	4
2.1. Introduction	4
2.2. Node API	4
2.2.1. copy	4
2.2.2. move	4
2.2.3. remove	4
2.2.4. revoke	4
3. Generic	5
3.1. Introduction	5
3.2. Generic API	5
3.2.1. convert	5

1. Introduction

1.1. about A9N

A9N は, HAL を用いて移植容易性を実現する, Capability-Based Microkernel です. ユーザーからのカーネルに対するすべての操作は, Capability と呼ばれるアクセス権を用いて実行されます. 基本的に, ユーザーからのカーネル呼び出しであるシステムコールは `ipc()` メカニズムを用いて,

```
ipc(capability_descriptor, args ... )
```

のように行われます.

しかしながら, この `ipc()` を直接呼び出す形式は最も低レベルなものであるため, `liba9n` ライブラリを用いた呼び出しを使用することが推奨されます.

例えば, Generic Capability に対する Convert 操作には,

```
common::error convert(  
    generic_descriptor,  
    type,  
    size,  
    count,  
    node_descriptor,  
    node_depth,  
    node_index  
)
```

のようなライブラリ関数が用意されます.

同様に, 他すべての Capability に対する操作へ, ラッパーであるライブラリ関数が用意されます.

2. Capability Node

2.1. Introduction

Capability Node は, Capability を格納するためのコンテナとして機能する Capability です.

Node のスロット数は作成時に規定されますが, 複数階層の Node を使用することにより, ユーザー側で拡張することが可能です.

Node 内の Capability は Descriptor によって, 以下のように Addressing されます :

- Node 内の Radix Bits から, Index に使用する Bit を決定します
- Descriptor から Index を取り出し, 子ノードを取得します
- 子ノードに対して, Descriptor を使い切るか, 終端に到達するまで再帰的に探索を行います

Node と Descriptor は Page Table と Virtual Address のような構造をしています.

Node 自体を指定したい場合など, Descriptor の残りに関わらず探索を途中で打ち切りたい場合, Node Depth を設定することで探索の上限を設けることが可能です.

2.2. Node API

2.2.1. **copy**

2.2.2. **move**

2.2.3. **remove**

2.2.4. **revoke**

3. Generic

3.1. Introduction

Generic は、メモリを抽象化する Capability です。

A9N カーネルはヒープを持たないため、カーネルオブジェクトのようなシステム内で使用するメタデータのメモリは、ユーザーが明示的に割り当てる必要があります。

生の物理メモリをユーザーに直接使用させるのはセキュリティ上のリスクが発生するため、`convert()`メカニズムを用いて安全な割当ポリシーを実現します。`convert()`は対象 Generic を切り出し、カーネルオブジェクトを作成します。作成したオブジェクトは親 Generic の Dependency Node に登録され、初期化処理などに使用されます。

3.2. Generic API

3.2.1. `convert`

```
common::error convert(
    library::capability::capability_descriptor generic_descriptor,
    library::capability::capability_type      type,
    library::common::word                     size,
    library::common::word                     count,
    library::capability::capability_descriptor node_descriptor,
    library::common::word                     node_depth,
    library::common::word                     node_index,
)
```

name	description
<code>generic_descriptor</code>	対象 Generic への Descriptor
<code>type</code>	作成する Capability の Type
<code>size</code>	作成する Capability の Size • Capability が固定長の場合、値は無視される
<code>count</code>	作成する Capability の個数
<code>node_descriptor</code>	格納先 Node への Descriptor

node_depth	格納先 Node を探索する深さ
node_index	格納先 Node の Slot