Hello, World!

An introduction to programming NoTA services and applications

Glossary

Term	Meaning
AN	Application Node. Application nodes are parts of a NoTA (sub)system that can interact with other nodes, but do not offer services of their own.
SN	Service Node. Just as application nodes can be thought of as clients, service nodes are analogous to servers. Service nodes can accept connections from application nodes or other service nodes, and can also initiate contact with other service nodes.
SIS	Service Interface Specification. A description of the services offered by a service node.
RM	Resource Manager. In each NoTA network, one subsystem must be specified as the resource manager. The resource manager handles tasks like keeping track of active services.
IA	Interconnect Address. Each subsystem within a NoTA network has an Interconnect Address. This address is used to coordinate communication between subsystems, and is not visible to the application developer.
HIN, H_IN	High Interconnect. The top layer of the interconnect stack, which communicates with the nodes in the network.
LIN, L_IN	Low Interconnect. The bottom part of the interconnect stack, responsible for communicating with the physical network.
HIF	High Interconnect Interface. The part of H_IN visible to application developers.
SID	Service Identifier. A unique identifier for each service node that is registered on the network.

Contents

1 Introduction

1.1 Abstract

This example demonstrates the basic model of NoTA services and applications under Interconnect 3. The service offers the capability to print the string sent to it to standard output. There are two versions of the service included, one that uses blocking calls and one that uses asynchronous IO multiplexing. For each program, there are two different versions: program and program_sp. The first uses the Interconnect Daemon, and the second runs in single process mode (for more information, see the documentation for Interconnect 3). Both are functionally identical, so no distinction will be made between the two in the rest of this document.

1.2 License

2 Installation

2.1 Prerequisites

This procedure assumes that you have a working Interconnect 3 installation. The program is available as a binary debian package and as a source tarball.

2.2 Compilation

To compile the program from source, use:

```
./autogen.sh

make
```

2.3 Installing

If you compiled the program from source, run

```
make install
```

You can also install the debian package:

```
dpkg -i nota3-hello-world_0.1.0-1.deb
```

2.4 Removal

As usual, the program can be removed by issuing either the command

```
make uninstall
```

Or, if you're using the packet manager:

```
dpkg -r nota3-hello-world
```

3 Use

3.1 Setting up

As with all NoTA programs, one part of your network must be set to act as Resource Manager. Please note that the Resource Manager should be started before any other devices/subsystems are connected.

3.2 Running the program

Once your network is up and running, it's time to introduce some service nodes to it. Launch either example service:

```
nota3_hello_world_blocking_sn
```

or

```
nota3_hello_world_nonblocking_sn
```

With the service registered on the network, try connecting to it with the example application node:

```
nota3_hello_world_an
```

If the connection is made successfully, you will be asked for a string to send to the service node.
Congratulations – you've just set up your first NoTA service network!

An example run of the Hello, world application

```
nota3_hello_world_an
Connecting to service '8'
Connected socket 8 to service '8'.
Enter string to send (255 characters maximum):
Hello, world of NoTA!
Send successful.
Would you like to ask the server to shut down ("y" for yes)?
y
Closing socket 8.
```

The same from the service side (blocking version)

```
nota3_hello_world_blocking_sn
Activating service with SID '8'
        Service active.
Waiting for connection...
        done - connected to socket 7.
Waiting for messages...
String received: Hello, world of NoTA!


Waiting for messages...
Quitting.
Deactivating service with SID '8'
        done.
```