rtshell入門

Geoffrey Biggs
(独)産業技術総合研究所
知能システム研究部門
統合知能研究グループ

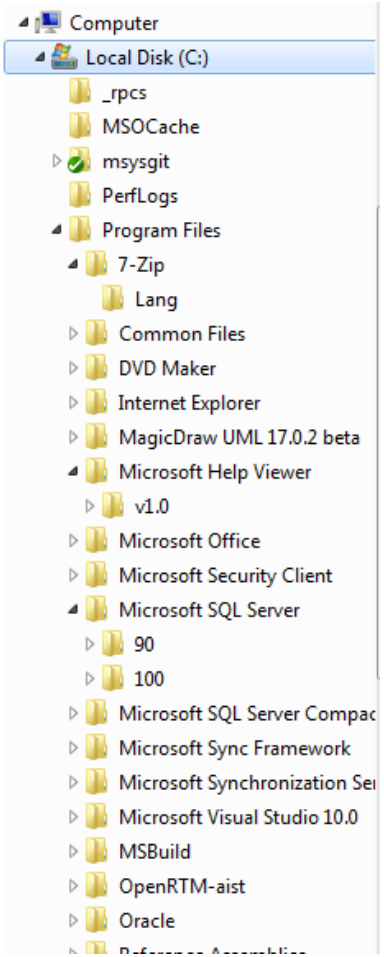
rtshellって何?

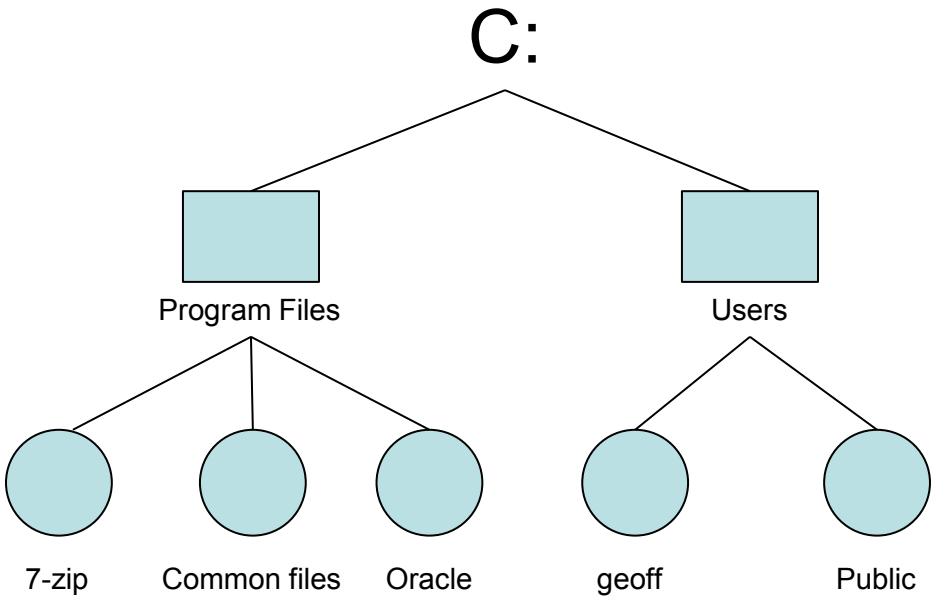
- コマンドラインで個々のRTコンポーネントやRTシステムを制御するツール
- RTSystemEditorと同等の機能を持つ
- テストやデバッギング用のコマンドを持つ
- RTSystemEditorが使えない場合に有効である
  - 少リソース
  - GUIが使えなくて、ネットワークもない

rtshellの特徴

- コマンドラインやシェルスクリプトでRTコンポーネントを接続したりアクティベートしたりすることが可能
- Windows、Linux、BSD、及びMacOS Xなどで利用可能
- RTシステム全体を一つのコマンドで起動、終了することが可能
- コマンドラインからrtprintとrtinjectでコンポーネントのテストを簡単に行うことができる
- コンポーネントが送るデータを記録し、再生する

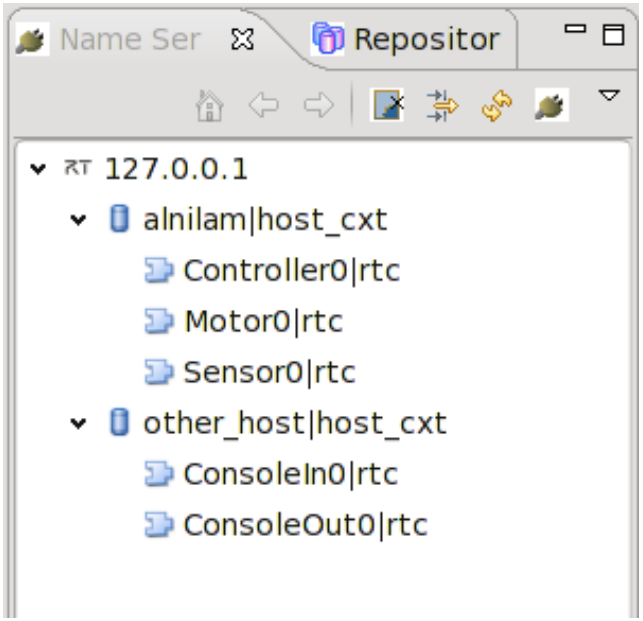
rtshellのバーチャルファイルシステム

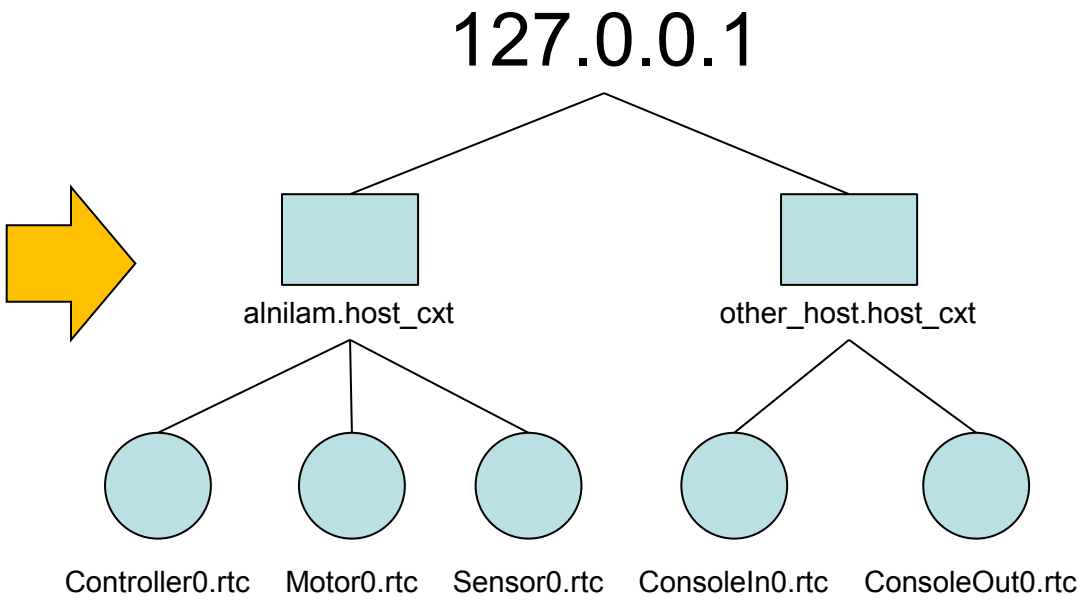




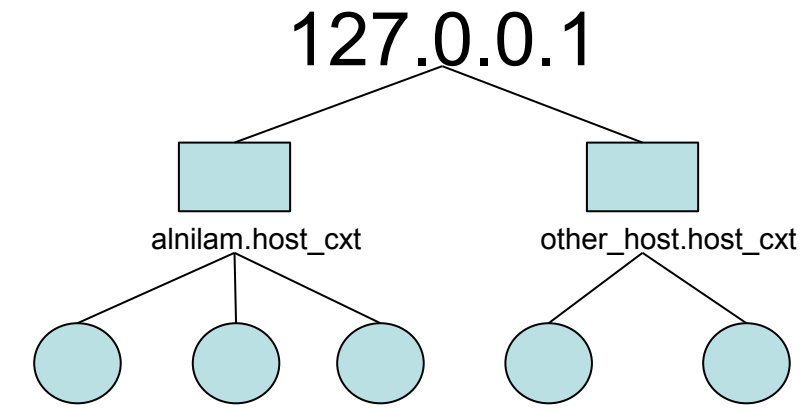
rtshellのバーチャルファイルシステム

- ファイルシステムのツリーと同じようにRTC等を見る





rtshellのバーチャルファイルシステム




```
geoff ~ $ rtls
localhost/
geoff ~ $ rtcwd localhost/
geoff ~ $ rtls
alnilam.host_cxt/  other_host.host_cxt/
geoff ~ $ rtcwd alnilam.host_cxt/
geoff ~ $ rtls
Controller0.rtc  Motor0.rtc  Sensor0.rtc
geoff ~ $ rtcwd ..
geoff ~ $ rtls
alnilam.host_cxt/  other_host.host_cxt/
geoff ~ $ rtcwd other_host.host_cxt/
geoff ~ $ rtls
ConsoleIn0.rtc  ConsoleOut0.rtc
geoff ~ $ rtpwd
/localhost/other_host.host_cxt/
```

バーチャルファイルシステムの管理

- rtcwd で現在のワーキングダイレクトリを変更する
- rtls で現在のワーキングダイレクトリの内容を表示する
- 迷ったらrtpwd でワーキングダイレクトリの名を表示する

```
geoff ~ $ rtls
localhost/
geoff ~ $ rtcwd localhost/
geoff ~ $ rtls
alnilam.host_cxt/  other_host.host_cxt/
geoff ~ $ rtcwd alnilam.host_cxt/
geoff ~ $ rtls
Controller0.rtc  Motor0.rtc  Sensor0.rtc
geoff ~ $ rtcwd ..
geoff ~ $ rtls
alnilam.host_cxt/  other_host.host_cxt/
geoff ~ $ rtcwd other_host.host_cxt/
geoff ~ $ rtls
ConsoleIn0.rtc  ConsoleOut0.rtc
geoff ~ $ rtpwd
/localhost/other_host.host_cxt/
```

コンポーネントライフサイクル変更

- rtact でコンポーネントをactivate する

```
$ rtact ConsoleIn0.rtc
```

- rtdeact でコンポーネントをdeactivate する
- rtreset でエラーステートにあるRTコンポーネントをリセットする
- コンポーネントの現在ステートをrtls -l で表示する

```
$ watch -n 1 rtls -lR
```

(Windowsで利用できない)

ポート接続

- rtcon で接続を作る
- rtdis で接続を切る
- IDを使ったら、単独な接続を管理できる

```
$ rtcon ConsoleIn0.rtc:out ConsoleOut0.rtc:in
      -i my_connection

$ rtcon ConsoleIn0.rtc:out ConsoleOut0.rtc:in
      -i other_connection

$ rtdis ConsoleIn0.rtc:out -i my_connection
```

コンポーネント情報

- rtcat でコンポーネントのプロパティ等を表示する
  - -l オプションを使って情報管理する

```
$ rtcat -l ConsoleIn0.rtc
```

- 大きなツリーならrtfind でコンポーネント等を探す
  - 名でフィルター
  - 種類でフィルター

```
$ rtfind -iname Con* -type cz .
```

ゾンビー!

- ゾンビーコンポーネントは邪魔になる
- rtdel で消す
- 「ツリーがなくなった!」
  - rtdel で何でも消せる!
  - -z オプションを使ったら、ゾンビーだけが消せる

コンフィグレーションパラメータ

- rtconf でコンフィグレーションパラメータを管理する
- パラメータの表示
- パラメータの編集

```
$ rtconf ConfigSample0.rtc -l

$ rtconf ConfigSample0.rtc -s mode0
     set int_param0 54321

$ rtconf ConfigSample0.rtc -l -s mode0
```

マネージャ

- ファイルシステムでマネージャはダイレクトリ
- rtcwd、rtls 等は使える
- rtcat でマネージャの詳しい情報を表示する
- rtmgr でコンポーネントを作ったり消したりする
- rtexit でコンポーネントの終了する

```
$ rtmgr manager.mgr/ -l
     ./rtc/SeqIn.so:SeqInInit

$ rtmgr manager.mgr/ -c SeqIn

$ rtexit SequenceInComponent0.rtc
```

マネージャでコンポジットコンポーネント

- rtcomp でコンポジットコンポーネントを作る(得に便利!)
  - 外部ポートの設定する
  - インスタンス名設定する

```
$ rtcomp manager.mgr:DriveUnit
      -a Motor0.rtc
      -a Controller0.rtc:in
      -a Sensor0.rtc:out
```

システム管理

- rtshell で全システムを管理する
  - RTSProfile ファイルを使う
  - RTSystemEditor もRTSProfile を使う
- rtresurrect で復元する
- rtstart で起動させる
- rtstop で停止する
- rtteardown で消す

```
$ rtresurrect my_system.rtsys
$ rtstart my_system.rtsys
```

システムのインスペクション

- システムを起動したら確認したい
  - rtcheck でライブなシステムをRTSProfile ファイルと比べる
- rtstodot でRTSProfile のモデルを可視化する
  - rtcryo と組み合わせたらライブなシステムを可視化する
  - （Windowsで利用できない）

コンポーネントのデバッギング

- rtshell でコンポーネントが送信し受信するデータのインスペクション
  - rtprint でアウトポートからのデータを表示する
  - rtinject でインポートにデータを送る
- データロギングでテスト

```
$ rtprint /localhost/ConsoleIn0.rtc:out

$ rtinject /localhost/ConsoleOut0.rtc:in
     -c 'RTC.TimedLong({time}, 42)'
```

データログ作り

- rtlog でコンポーネントが送るデータをファイルに記録する
- オフラインでrtlog によってファイルからのデータを再生する
  - ハードウェアなしでアルゴリズムのテスト

```
$ rtlog -f log.rtlog
    /localhost/ConsoleIn0.rtc:out.numbers

$ rtlog -f log.rtlog -p
    /localhost/ConsoleOut0.rtc:in.numbers
```

ドキュメント

- 全コマンドで -h オプションによってヘルプを取得

```
geoff ~ $ rtinject -h
Usage: rtinject [options] <path1>:<port1> [<path2>:<port2>...]
Write a constant value to one or more ports.

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -c CONST, --const=CONST
                        The constant value to send, as a Python expression. If
                        not specified, values will be read from standard in.
  -m MODULES, --mod=MODULES
                        Extra modules to import. If automatic module loading
                        struggles with your data types, try listing the
                        modules here. The module and its __POA partner will be
                        imported.
  -n MAX, --number=MAX  Specify the number of times to write to the port.
                        [Default: 1]
  -p PATHS, --path=PATHS
                        Extra module search paths to add to the PYTHONPATH.
  -r RATE, --rate=RATE  Specify the rate in Hertz at which to emit data.
                        [Default: 1.0]
  -t TIMEOUT, --timeout=TIMEOUT
                        Write data for this many seconds, then stop. This
                        option overrides --number. [Default: -1]
  -v, --verbose         Output verbose information. [Default: False]
```

ドキュメント

- UNIX型では深い説明（使い方の例を含む）がmanページに

```
RTINJECT(1)                                                    User commands
                                RTINJECT(1)



NAME
       rtinject - inject data into ports

SYNOPSIS
       rtinject [options] <path1:port1> [path2:port2...]

DESCRIPTION
       Writes constant values to one or more ports. By default, the value is written once. Options ar
e available to write a set number of
       times, or write regularly for a specified length of time.

       A connection will be made to the port using the default connection settings compatible with th
e port.

OPTIONS
       -c CONST, --const=CONST
              The constant value to send, as a Python expression. If not specified, values will be re
ad from stdin. Any occurrences of {time}
              in the constant expression will be replaced with the current time.

       -m MODULES, --mod=MODULES
              Extra modules to import. If automatic module loading struggles with the constant's data
 types, try listing the modules here.
              The module and its __POA partner will be imported.

       -n MAX, --number=MAX
lines 1-26
```

```
$ man rtinject
```

ドキュメント

- Windows型では深い説明がhtmlページにある

http://www.openrtm.org/pub/OpenRTM-aist/tools/rtshell/3.0/ja/rtshell.html

チュートリアル

- openrtm.orgにチュートリアルがある
- rtshell によるRTシステムの管理

http://openrtm.org/openrtm/ja/node/5014/

- rtshell でコンポーネントデータの保存・再生

http://openrtm.org/openrtm/ja/node/5015/

- Youtube にスクリーンキャストがある

http://www.youtube.com/user/OpenRTM

まとめ

- rtshell で開発効率をあげる
- RTSystemEditor と組み合わせたら全部の力を得る
- 説明書に参照してください
  - 全部のコマンドで -h オプションによってヘルプを取得
  - man ページには深い説明がある
  - (Windows の場合はHTML として提供する)