

3 ROS

3.1 ROSとは？

- 以下の文書を抜粋したURL

<http://rtm-ros-robotics.googlecode.com/svn/wiki/20110420-rtmros-okada.pdf>

- 2007年Switchyard Stanford AI研M.Quigley
- 2008年WillowGarage社がサポート開始,Brian Gerkey (Stage/Player) がWG社に参画
- ロボットアプリケーションを作成するソフトウェア開発者のためのライブラリとツール。ハードウェア抽象化
- <http://www.ros.org/news/2010/11/happy-3rd-anniversary-ros.html>
- <http://www.willowgarage.com/>

- 以下の文書を抜粋したURL

<http://www.ros.org/wiki/ja/ROS/Introduction>

ROSはあなたのロボットのための、オープン・ソースのメタオペレーティング・システムです。ROSはあなたがオペレーティングシステムに望んでいたであろう、ハードウェア抽象化や低レベルデバイス制御・よく使われる組み込み関数・プロセス間通信・パッケージ管理の機能を持っています。さらにROSは多様なコンピュータ間を横断して適用したり、ビルドしたり、記述したり、実行したりするコードのためのツールやライブラリも提供します。「ROS」はPlayerやYARP・CARMEN・Orca・MOOS・Microsoft Robotics Studioといった、“ロボットフレームワーク”といくつかの点で似ています。

ROSのランタイム“graph”はROSコミュニケーション・インフラを用いて接合されたプロセス同士の、ゆるいPeer-to-Peerネットワークで成り立っています。ROSには、サービスごしの同期RPC形式の通信やトピックごしの非同期データ・ストリーミング・パラメータ サーバ上のデータ・ストレージといった、幾つかの異なるタイプの通信方法が実装されています。これらについてはROS のコンセプトに詳細な説明があります。ROSはリアルタイムのフレームワークではありませんが、ROSにはリアルタイムのコードを含めることができます。Willow GarageのPR2ロボットはpr2_etherCATと言うシステムを使用していますが、これはリアルタイム・プロセスへROSのメッセージを送受信するものです。さらにROSはOrocosのリアルタイム・ツールキットとシームレスに統合します。

代表的なROSのスタックリストの紹介も含む

<http://www.ros.org/wiki/StackList>

3.2 ROSのプログラミングモデル

T・B・D

3.3 ROSを使ってみよう！

3.3.1 インストール

1 ROS本体のインストール

- Linux

<http://www.ros.org/wiki/electric/Installation/Ubuntu>

より抜粋

ROS本体、Robot向け汎用パッケージ、PR2向けパッケージなど公開されているパッケージをソースでインストールする。まずは以下の様にaptのリポジトリを登録し、

```
sudo apt-get install build-essential python-yaml cmake subversion wget python-setuptools git-core mercurial aptitude
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu `lsb_release -cs` main" > /etc/apt/sources.list.d/ros-latest.list'
wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
```

次に以下の作業を行う。

```
sudo apt-get update
sudo apt-get install python-setuptools
sudo easy_install -U rosinstall
sudo apt-get install ros-electric-wg-pr2-apps ros-electric-pr2-apps ros-electric-pr2-desktop ros-electric-openni-kinect
```

2 ROS外部パッケージのインストール

```
rosinstall ~/prog/rtm-ros-robotics /opt/ros/electric http://rtm-ros-robotics.googlecode.com/svn/tags/latest/agentsystem_ros_tutorials/rtm-ros-robotics.rosinstall
```

3 ROS関連コマンド

最低限 `roscd` , `roscdep` , `rosmake` , `roslisp` を知っていればよい。例えば,

```
roscdep install euslisp
rosmake euslisp
```

とすると, euslisp
パッケージに必要なパッケージをダウンロード&インストールし, さらにコンパイルする。

また, euslispパッケージにあるディレクトリに移動したければ

```
roscd euslisp
```

とすればよい。また, euslispパッケージ以下のirteusglという実行ファイルを実行したければ

```
roslisp irteusgl
```

とする。引数も渡せるので,

```
roslisp irteusgl irteus/demo/demo.l "(crank-motion)"
roslisp irteusgl models/irt-all-robots.l "(make-all-robots)"
roslisp irteusgl models/irt-all-objects.l "(make-all-objects)"
```

等としてサンプルを試すことが出来る。crank-motionはリターンキーを打つと止まり、irteusgl\$のプロンプトが出る。プロンプトに対して、exitを入れるとeuslispが終了する。

より詳しい情報は <http://www.ros.org/wiki/ROS/Tutorials> へ, また,

いざとなったら [ROS CheatSheet](#) を参考にすると助けになる。

ほとんどのunixコマンドは先頭にrosをつけたツールが存在する(rosls, roscp等)。いろいろと調べるとよいが, もしかしたらその先は [奥深い](#) ので要注意。

インストールができれば, [サンプルプログラム](#) を試してみよう

3.3.2 サンプルプログラム

- Hello World

T. B. D

http://code.google.com/p/rtm-ros-robotics/wiki/ROS_Example