

# 第18回 TOPPERS 開発者会議 開催レポート

2017年10月15日(日)、16(月)、17日(火)に、おんやど恵(神奈川県湯河原町)にて、第18回 TOPPERS 開発者会議を開催しました。TOPPERS 開発者会議は、TOPPERS プロジェクトの開発成果物の開発、利用に携わる方(非会員でも参加可能)が集まり、2泊3日の合宿形式で集中的に議論、開発する会議です。初の湯河原での開催となり、今年も活発な議論が展開されました。本レポートでは、写真を交えながら、その魅力をたっぷりお伝えします。

## ゲストトーク：



eSOL 株式会社エンベデッドプロダクツ事業部技術部部長の坂本裕和氏をお招きし、eSOL 社が開発しているマルチ・メニーコア向け RTOS の開発経緯や仕組みについてお話を伺いました。TOPPERS プロジェクトでは、マルチコア向け RTOS である TOPPERS/FMP カーネルを公開しており、またメニーコア向け RTOS を研究、開発中です。

講演の前半では、RTOS が動作するハードウェアアーキテクチャを整理するために、マルチコア、メニーコアの定義とアーキテクチャ例をご紹介いただきました。マルチコアは、「プロセッシングコアが～8個程度で、各コアのキャッシュ間で一貫性を保つキャッシュコヒーレンシが維持できるアーキテクチャ」、メニーコアは「プロセッシングコアが数10～数千個程度で、コア間のキャッシュコヒーレンシが実現されていないアーキテクチャ」という定義を紹介されました。

後半では、RTOS の設計・実装について紹介していただきました。マルチコア向けの RTOS は、キャッシュコヒーレンシ、全コアがアクセスできる共有メモリの存在を前提とするので、シングル・コア用 RTOS を拡張する形で実現しています。CPU ごとに、タスクを固定するか、他のコアへのマイグレーションを許可するかどうかを選択できる仕様となっており、この点は、タスクマイグレーション API を用意している TOPPERS/FMP カーネルの仕様とは異なる考え方です。メニーコア向け RTOS は、キャッシュコヒーレンシ、共有メモリの存在を前提としていないので、分散マイクロカーネル設計を採用しています。この設計では、コア間で通信する際は、すべてメッセージパッシングで実行されます。OS の高機能なサービス(例えば、POSIX のセマフォ/Mutex)は、クライアント・サーバ型のメッセージ通信で実現されており、サーバに対して、各タスクはメッセージを送信することで OS の機能を利用します。各コアの OS

(マイクロカーネルやライブラリ) で実現されるサービスと、クライアント・サーバ型メッセージ通信で実現されるサービスを、最適化してアプリケーションに提供します。

質疑では、特にメニーコア向け RTOS について、メッセージパッシングによる処理オーバーヘッド、リアルタイム性の実現、サーバの配置方法など、RTOS としての性能を維持できるかどうか、スケーラビリティはどうかという観点で議論が進みました。加えて、マルチ・メニーコア向けのアプリケーション開発は、シングル・コア向けとは大きく異なります。メニーコア向け RTOS をさらに普及するためには、アプリケーション開発支援ツールを含めた総合的な技術開発が必要との考えで一致しました。組込みシステム向けのメニーコアアーキテクチャはまだ発展段階であり、今後も、様々なアーキテクチャ、製品が登場することが予想されます。TOPPERS プロジェクトでも引き続き、マルチ・メニーコア向けの RTOS 開発を進めてまいります。

## メニーコア OS

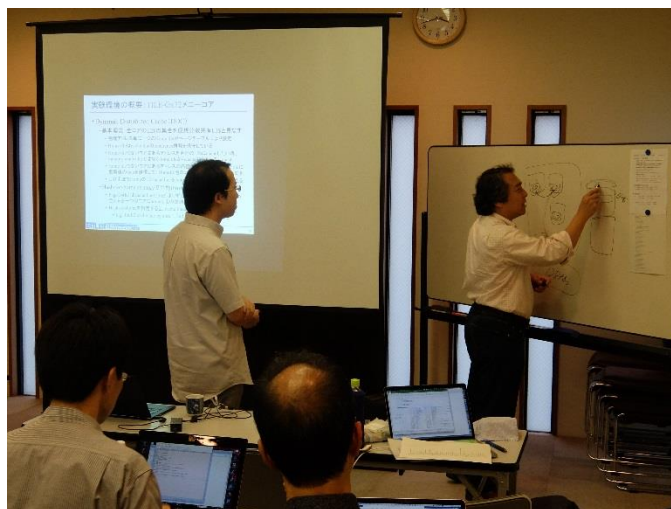
メイン議論 1 として「メニーコア OS」について、名古屋大学の李さんから発表がありました。メニーコアプロセッサが組込みシステムに浸透し始め、それよりもコア数が多いマルチコアについても、高性能の並列アプリケーション（例えば自動運転）向けに導入されると予想され、組込みシステムで使われる RTOS がメニーコアにも適しているか検証した話をされました。

コア数に応じて、10 個以下のマルチコア、十数個から百個程度の中規模メニーコア、数百以上の大規模メニーコアがあり、マルチコアでは共有バス、中規模メニーコアでは NoC (Network on Chip) を使用し、各コアが見るメモリの内容を同じにする（コヒーレンスを取る）仕組みを持たせている。コヒーレンスが取れていることで、従来の並列アプリケーションの性能向上が望めるが、大規模メニーコアではコヒーレンスを諦め、違ったアーキテクチャを使うため、従来のアプリケーションでは対応できず、違った性能の上げ方が必要と説明がありました。

中規模メニーコアまでは、Linux を使ったマルチコア向けの最適化で成果が上がっています。RTOS は Linux よりオーバーヘッドが少ないので、中規模メニーコア向けに最適化することで、Linux より高い性能を出せる可能性があると言われました。

まとめとして、中規模メニーコア向けに、最適化を行った FMP と Linux とは近い性能が出せ、FMP ではおおむね少し高い性能が出ることが示されました。RTOS も最適化することで中規模メニーコアにも使用できることが分かりました。

参加者からの意見として、カーネルが小さい方がキャッシュの占める割合は小さく性能に有利ではないかという指摘や、RTOS の特徴であるリアルタイム性を評価できるベンチマークがあると、より有利な結果を示せるのではという指摘がありました。



## ASP3 カーネルの環境構築とポーティング方法

TECS WG の主査である大山氏による ASP3 カーネルで使用されている TECS 技術の話について TOPPERS カンファレンスの資料を引用しながら説明をしてもらいました。

### ●TECS コンポーネントとカーネルの関係

コンポーネントを用いない場合は、TOPPERS/ASP までと同様に、サービスコールや syslog 関数を直接呼び出すことも可能です。

サービスコールに対応するコンポーネントはサービスコールに対するラッパーとみなすことが出来ます。システムログ機能のコンポーネントは、コンポーネントからの呼び出しだけでなく、C 言語の syslog 関数としても呼び出せるように実装されています。

カーネルから見た場合、コンポーネントからの呼び出しとコンポーネント以外からのサービスコールの直接の呼び出しは区別されません。どちらも同じサービスコールの呼び出しとして処理されます。

#### ●TECS コンポーネントのうれしいところ

TECS ジェネレータは、CDL で定義されたシグニチャ定義、セルタイプ定義、セル組み上げ記述等に基づき、コンポーネントを実行するためのC言語ソースコード、セルタイプコードの（C言語の）スケルトンだけでなく、Makefile（自動生成したソースコード間の依存関係を含む）、静的API記述ファイルなど、コンパイルに必要なファイルをすべて生成します。

### Autoware チュートリアル

最近自動運転などで注目を浴びている Autoware やその中核の技術である mROS に関する議論も開発者会議で取り上げました。

大阪大学の安積卓也さんより Autoware のチュートリアルということで、参加者には事前に GitHub 上にある Autoware のインストールを準備する連絡がありました。持ち込み PC 参加者の中にはインストールパッケージの依存関係でうまくセットアップできない人もいましたが、参加者全員が Autoware を体験することができました。

講義編では Autoware の説明ということで Autoware の中核である、自己位置推定に関する説明してもらいました。カメラなどの画像データだけでなく、予定運転を行うためには地図とその経路さらに自車情報が大切だということがわかりました。また位置測定のために準天頂衛星が重要なわけも分かりやすいスライドで説明してもらいました。

演習編ではインストールした Autoware に予め演習用に用意した地図データとドライビングの位置データファイルを渡してもらい、実際に Autoware での操作を体験しました。マシンのスペックによりませんが、仮想環境上でも動作を確認できることもわかり専用の PC を用意しなくても試すことができました。地図データを変更した場合にどのような挙動になるのかと言った質問がでました。Autoware は GitHub などで公開されていますので、みなさんも試してみたいはいかがでしょうか。

<https://github.com/CPFL/Autoware>



### mROS に関する議論

まず始めに mROS の紹介を行った後、課題について技術討論を実施する予定でしたが、参加者の関心度が高かったこともあり、前半から質疑応答が非常に活発に行われました。

ROS (Robot Operating System) は、機能をノードという単位で表現して設計開発を進めるロボットソフトウェア開発支援フレームワークです。（OS ではなく）Linux のミドルウェアであり、Publish/Subscribe 方式によるノード間の通信プロトコル・通信レイヤを提供しています。ただし、ROS は Linux (Ubuntu) が無いと動かないのが難点となります。

mROS は、Linux を搭載できないような低消費電力のマイコン上でも ROS が使えるように開発された軽量実行環境です。mROS はマイコンの中でもミッドレンジ、RTOS がきびきび動くような環境をターゲットにしています。特別会員団体である京都大学高木研究室にて開発が進められており、ソースコードは GitHub にて公開されています。

<https://github.com/tlk-emb/mROS/>

mROS はアプリケーション作成の負荷を減らすための配慮もなされています。ROS ノードは ROS API を直接記述して作成することができ、ROS マスタの動作するホスト PC とネイティブな ROS の通信プロトコルでデータをやり取りすることができます。このため、数多く公開されている既存の ROS パッケージをそのまま組み込みマイコン上に移植して実行できるようになる可能性を秘めています。参加者が TOPPERS

に詳しい人たちということもあり、活発な議論がなされました。特に、RTOSを使う意義や、mROS内のデータ通信・同期の仕様を中心とした実装方法について重点的に検討が行われました。また、このセッション内での要望を受け、開発者会議内でのハッカソンでは、mROSの実装対象となっているGR-PEACH用ASP+mbedパッケージのmbedライブラリの最新版への追従、ASP3のGR-LYCHEE対応も実装されました。

mROSについて詳しくは以下資料が公開されていますのでそちらをご覧ください。mROSをGR-PEACH上で動作させる方法も記載されています。

<https://github.com/tlk-emb/mROS/blob/master/doc/mROS.pdf>

また、積極的にユーザ企業を募集して開発方針を定めていきたいとのことでしたので、ご興味のある方はぜひ **mROS 開発メンバ** ([emb@lab3.kuis.kyoto-u.ac.jp](mailto:emb@lab3.kuis.kyoto-u.ac.jp)) にお問い合わせください。



## WGの現状が分かるプロジェクトアップデートセッション

TOPPERSでは幾つかのワーキンググループがあり、開発者会議ではTECS、教育、ホームネットワークの3グループからの発表がありました。

### ■TECS

組込みシステムに適したコンポーネントシステムであるTECS (TOPPERS Embedded Component System)の開発、応用を行っています。現在公開されているものとして、効率よく、フラグメントが起きにくいアロケータ (TLSFアロケータ)のTECSコンポーネント化、TOPPERS/HRP2への対応、VMを1タスク1VMとして複数VMが実行可能なmrubyのコンポーネント化、TCW for TECS (軽量GUIツールキット)の開発、TECSリファレンスマニュアルがあります。

開発中のものとしては、TINETのTECSコンポーネント化、TOPPERS/HRP3への対応、実行時にTECSコンポーネント関連情報を取得できる仕組みであるTECSInfoが報告されました。

### ■教育

講座形態では大規模な組込みプラットフォーム教育は無理だと判断し、実際にエバレーションボードに使ってみたいアプリを載せ、自ら改良することにより技術者育成を図る (アプリ作成からデバイスドライバ開発まで) ことにしました。Media Player (JPEG表示、MP3演奏、SDカード、USBメモリ)をアプリとして実装した高速コンピュータ設計の例として「CQ出版 Interface2016年12月号」に掲載されました。

今年はSTM32F401RE nucleo、STM32446TR nuvltro-64のTOPPERS BASE PLATFORM対応版基礎1, 2セミナーを開催しました。さらにSPIドライバを使い、ArduinoカラーLCDシールド表示、SPI-SDカードファイルシステム対応、ADCドライバでJoy Stick操作を行う新基礎3講座を開発中です。

また、Intel社Cyclone V SoC-FPGAをコアに使用したterasIC社のDE0-Nano-SoCボードを対象としたFMPカーネルをベースとした組込みプラットフォームTOPPERS BASE PLATFORM (CV)を開発中です。

ブレッドボードに回路を組み、デバイスドライバを開発する基礎ハードウェア設計の実験セミナーを行いました、この教材はサイトから入手可能です。

### ■ホームネットワーク

家電にもネットワークの機能が求められるようになり、ECHONET Lite (通信プロトコル) 搭載機器 (HEMS、エアコン、冷蔵庫、蓄電システムなど) も増えてきています。さらにそれらをWebサービスと連携させるには、HTTPS通信、ファイルシステム、スクリプト言語が必要になります。

WG では TOPPERS/ECNL (ECHONET Lite 通信ミドルウェア) 周辺の開発を行っています。現在までに mruby 向け ECNL プロトタイプツール (機器オブジェクトのスケルトン生成)、TOPPERS/SSP 版 (依存部は RL78) を開発しました。ECNL プロトタイプツールは Web アプリです。ビジュアル言語を用いて設計し、mruby へコンパイルし、機器上の mruby 実行環境で実行します。FileSystem、Socket を実装し、Contributed Software として ([https://dev.toppers.jp/trac\\_user/contrib/wiki/EcnlProtoTool](https://dev.toppers.jp/trac_user/contrib/wiki/EcnlProtoTool)) で公開しています。実装には Socket API が含まれない newlib ではなく、Linux 用静的リンク向け標準 C ライブラリ musl を採用しました。さらに標準 C タスクを使って、OpenSSL や WolfSSL、HTTPS クライアント、DNS クライアントなどが動くように開発を進めています。

OSC (Open Source Conference) などでもデモの展示をしています。

## TOPPERS の今後の活動戦略

食後議論では、TOPPERS の今後の活動戦略について議論しました。

TOPPERS 開発者会議の参加者を増やしたいというゴールに向けて、誰に向けて何をすれば良いかを議論しました。開発を手伝ってくれそうな上級者、ユーザになってもらえそうな初心者などターゲットをいくつかのグループに分類し、それぞれ対策を検討しました。同時に、新規ユーザ開拓を目的として OSC で実施している TOPPERS セミナーの内容見直しを行い、ワーク形式に変更することになりました。使用するボードについても活発に議論がなされ、所持している人が多く、また他の用途にも使えることから raspberry pi を利用するのが最適という結論になりました。

今後の活動として、OSC で実施する raspberry pi を利用したハンズオンの準備、さらに TOPPERS を知ってもらうための新しい試みとしてクラウドファンディング (Crowdfunding) が追加されました。これらが実施された際にはぜひともご参加くださいますようお願いいたします。

## Maker Fair 対策

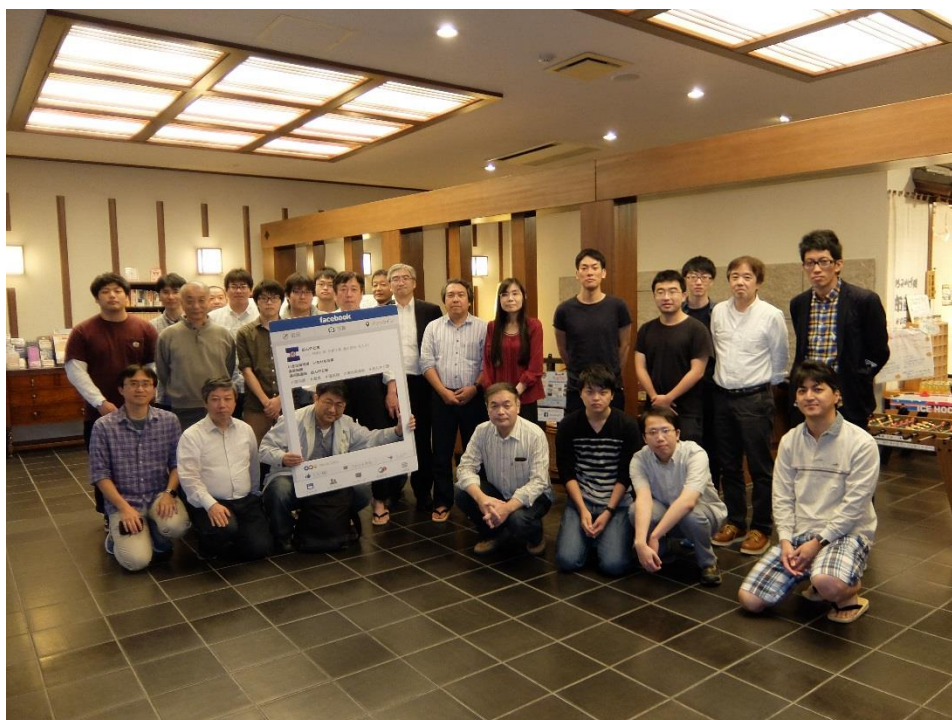
2日目の食後議論では、Maker Faire Tokyo への出展に向けた検討や議論を行いました。

TOPPERS プロジェクトでは、開発者会議実行委員会のメンバが中心となって、全国各地で開催されているオープンソースカンファレンスへの出展を行っています。オープンソースコミュニティにおける TOPPERS プロジェクトの知名度向上や普及を目的としており、これまで一定の成果を挙げてきました。この活動の次なる一歩として、Maker Faire Tokyo への出展を検討しています。

議論では、まず、Maker Faire Tokyo に来場参加したことがある方から、イベントの開催風景や印象に残った展示物などを紹介いただき、YouTube にアップされている動画も見ながら、開発者会議の参加者で情報を共有しました。応答速度などなにかを極めた展示や、使えるものかどうかより作っていることが楽しいことが分かる展示などが来場者の興味を特に惹いていたそうです。実際にもものづくりを行うワークショップを実施しているところもありました。また、来場者はお子様連れの方が多く、子供向けに見て分かる展示やブースが良いのではないかという意見が出ました。

TOPPERS プロジェクトの強みはものづくりのプラットフォームにあります。これを活かしながら Maker Faire Tokyo の来場者が楽しく思える展示を開発していきたいところです。今後、今回の開発者会議に参加された方が有志となって、出展に向けてさらなる検討を進めていきます。Maker Faire Tokyo 2018 に TOPPERS プロジェクトが出展できたら、ぜひともブースにお立ち寄りください！





### ハッカソンや開発者会議の感想を最後に発表

クロージングでは、開発者会議へ参加した感想や、各自で取り組んだハッカソンの作業成果などについて、参加者から発表を行っていただきました。学生の参加も多く、まとまった時間をこのハッカソンにあてて研究ができるという感想をいただきました。また TOPPERS 会員ではない方からは、TOPPERS の内部で動いている方の活動を知ることができて、自分も何か活動に参加できることを探していきたいとコメントもいただきました。

#### 開発者会議実行委員会

高田光隆 (名古屋大学/実行委員長)  
伊藤良樹 ((株) ヴィッツ)  
伊予田健敏 (創価大学)  
岡山直樹 (アイシン・コムクルーズ(株))  
小川清 (名古屋市工業研究所)  
小南靖雄 (個人会員)  
高瀬英希 (京都大学)  
長島宏明 (コアーズ)  
平橋航 (東芝情報システム(株))  
堀武司 (北海道立総合研究機構工業試験場)  
松原豊 (名古屋大学)  
山根ゆりえ (特別会員)

#### NPO 法人 TOPPERS プロジェクト

〒103-0011 東京都中央区日本橋大伝馬町 6-7 住長第 2 ビル 3F  
(一社) 組込みシステム技術協会内

TEL & FAX: 03-5643-5166 Email: devconf@toppers.jp

Facebook : <http://www.facebook.com/toppersproject>

Copyright (C) 2000 - 2017 by TOPPERS Project, Inc. All Rights Reserved.

※ “TOPPERS” および TOPPERS プロジェクトのロゴは、TOPPERS プロジェクトの登録商標です。