

# トロンプロジェクトと オープンソースソフトウェア

2003年11月19日

高田田広章広 章

名古屋大学大学院 情報科学研究科 教授

NPO法人 TOPPERSプロジェクト 会長

Email: [hiro@ertl.jp](mailto:hiro@ertl.jp) URL: <http://www.ertl.jp/~hiro/>

## 講演の内容

### トロンプロジェクトと組み込みシステム

- ▶ トロンプロジェクトとITRON仕様
- ▶ 自己紹介 ~ 今日の講演の立場
- ▶ 組み込みシステムとは？

### TOPPERSプロジェクト

### 組み込みシステムにおけるオープンソースソフトウェア

- ▶ オープンソースソフトウェア利用の現状
- ▶ オープンソースのビジネスモデル
- ▶ 組み込みシステムにおけるライセンスの問題
- ▶ TOPPERSライセンスとT-Kernelのライセンス条件

### オープンソースの開発における知的財産権の扱い

### まとめ

## トロンプロジェクトとは？

- ▶ ユビキタスコンピューティング社会における基盤技術の構築を目指す産学共同のプロジェクト  
プロジェクトリーダー：坂村 健 (東京大学教授)  
プロジェクトの開始：1984年
- ▶ “オープンアーキテクチャ”

### 各方面への取り組み (主なもの)

- ▶ ITRON (組込みシステム向けOS)
- ▶ BTRON (パソコン/ワークステーション向けOS)
- ▶ CTRON (通信/サーバ向けOS)
- ▶ トロン仕様マイクロプロセッサ
- ▶ トロンHMI (Human-Machine Interface)
- ▶ T-Engine (組込みシステム向け開発プラットフォーム)
- ▶ ユビキタスIDとeTRON

## トロンプロジェクトにおけるオープンの意味

- ▶ トロンプロジェクトでいう“オープンアーキテクチャ”は、トロンプロジェクトで標準化した仕様に従って、誰でも自由に製品を作り、販売・利用できるという意味
  - μITRON4.0仕様書より
  - μITRON4.0仕様は、オープンな仕様である。誰でも自由に、μITRON4.0仕様に準拠したソフトウェアを開発・使用・配布・販売することができる。トロン協会ITRON部会へのライセンス料の支払いや届け出は必要ない。
- ▶ 仕様に準拠して実装されたソフトウェアはオープンとは限らない
  - ▶ “オープンソースソフトウェア”とはオープンの意味が異なる
  - ▶ 実際、数多くのトロン仕様準拠のOSが販売されている

## ITRON仕様とは？

- ▶ トロンプロジェクトにおいて標準化を推進してきた組み込みシステム用のリアルタイムOSとそれに関連する仕様
- ▶ 誰でも自由に実装できるオープンな仕様

### ITRON仕様の歴史と現状

- ▶ 1984年のプロジェクト開始からこれまでの約20年間に、4世代のITRONカーネル仕様を策定・公開
- ▶  $\mu$ ITRON4.0仕様は、現世代のリアルタイムカーネル技術の範囲では、極めて完成度の高い仕様に
- ▶ メジャーな組み込みシステム用プロセッサのほとんどすべてに、ITRON仕様のRTOSが実装されている
- ▶ 国内で最も広く使われているRTOS仕様 (デザイン数ベースでのシェアが40~50%)。特にコンシューマ機器の分野において、シェアが大きい

## 自己紹介

### 1988年～1997年 東京大学 理学部 坂村研究室

- ▶ 坂村健教授の元で，大学院生～助手として，トロンプロジェクト (主にITRON) 関連の研究・開発・標準化に従事
- ▶ ITRON仕様OSを開発し，オープンソースとして配付

### 1997年～2003年 豊橋技術科学大学

- ▶ ITRONカーネル仕様の最新版である  $\mu$ ITRON4.0仕様の策定の中心となる (1999年6月に仕様書を公開)
- ▶ オープンソースのITRON仕様OSであるTOPPERS/JSPカーネルを開発・公開 (2000年11月に公開)．TOPPERSプロジェクトを開始

### 2003年～ 名古屋大学大学院 情報科学研究科

- ▶ NPO法人 TOPPERSプロジェクトを設立 (2003年9月)
- ▶ 組み込みリアルタイムシステムに関する研究に従事

## 組み込みシステム

### 組み込みシステムとは？

生活環境に入り込んだコンピュータ

- ▶ 各種の機器に組み込まれて，その制御を行うコンピュータシステムのこと

例) 自動車 (20 ~ 70 のマイクロプロセッサ)

携帯電話, 家電 (テレビ, デジタルカメラ, ...)

ロボット, 工作機械, ゲーム機 などなど

### 組み込みシステム開発の動向と課題

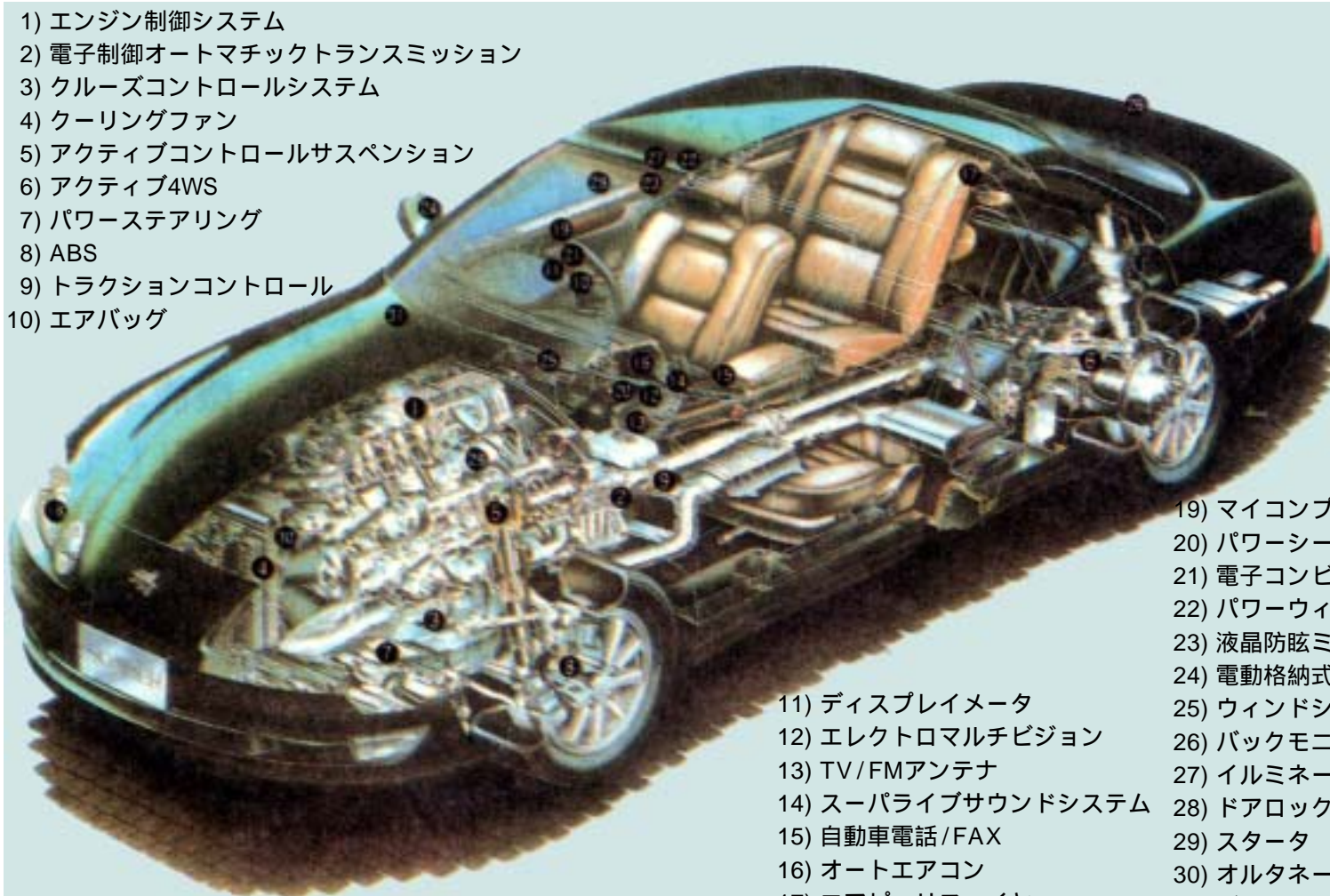
- ▶ 組み込みシステムの適用分野が拡大
- ▶ 従来からの組み込みシステムの大規模化・複雑化
- ▶ 開発期間の短縮やコストダウンに対する強い要求



- ▶ 組み込みシステムの開発技術が大きな課題に

## 例) 自動車内の組み込みシステム

- 1) エンジン制御システム
- 2) 電子制御オートマチックトランスミッション
- 3) クルーズコントロールシステム
- 4) クーリングファン
- 5) アクティブコントロールサスペンション
- 6) アクティブ4WS
- 7) パワーステアリング
- 8) ABS
- 9) トラクションコントロール
- 10) エアバッグ



- 11) ディスプレイメータ
- 12) エレクトロマルチビジョン
- 13) TV/FMアンテナ
- 14) スーパライブサウンドシステム
- 15) 自動車電話/FAX
- 16) オートエアコン
- 17) エアピュリファイヤ
- 18) コンライト
- 19) マイコンプリセットステアリング
- 20) パワーシート
- 21) 電子コンビネーションスイッチ
- 22) パワーウィンドウ
- 23) 液晶防眩ミラー
- 24) 電動格納式ミラー
- 25) ウィンドシールドワイパ
- 26) バックモニター
- 27) イルミネーテッドエントリー
- 28) ドアロックコントロール
- 29) スタータ
- 30) オルタネータ
- 31) バッテリ
- 32) ワイヤハーネス

資料提供: トヨタ自動車



## 組み込みシステムの特性

### 専用化されたシステム

- ▶ システム全体が一つの目的に専用化して設計される

### 厳しいリソース制約

- ▶ コストダウン要請 ← 大量生産品で顕著
- ▶ 低消費電力，動作環境 (温度条件など)，軽量化

### 高い信頼性

- ▶ システムの誤動作が機器の誤動作に直結 (無保証というわけにはいかない．PL法の対象に含まれる)
- ▶ システム改修に高いコストがかかる
- ! 機器の信頼性に対するユーザの期待

### リアルタイム性

- ▶ 制御対象の機器の定める時間要件に従って動作することが必要 (単に速ければよいわけではない！)

## 組み込みシステム開発の動向とITRON

### 大規模・複雑化する組み込みシステムとITRON仕様

- ▶ ITRON仕様OSのシェアの高いコンシューマ向けの「小さい」機器でも，ソフトウェアは大規模・複雑に
- ▶ ITRON仕様を，大規模・複雑なソフトウェアにも適するように発展させる必要性

### ITRON仕様からの新しい展開

- ▶ 弱い標準化の発展 (従来の標準化の延長)
  - ▶  $\mu$ ITRON4.0仕様 保護機能拡張
  - ▶ デバイスドライバ設計ガイドライン / 標準化
- ▶ T-Engineプロジェクト <http://www.t-engine.org/>
  - ▶ ハードウェアまで含めたより強い標準化
- ▶ TOPPERSプロジェクト <http://www.toppers.jp/>
  - ▶ オープンソースソフトウェア化による展開

## TOPPERSプロジェクト

TOPPERS = **T**oyohashi **O**pen **P**latform for  
**E**mbedded and **R**eal-Time **S**ystems



### プロジェクトの活動内容

- ▶ ITRON仕様の技術開発成果をベースとして、組み込みシステム構築の基盤となる各種のオープンソースソフトウェアを開発するとともに、その利用技術を提供

*組み込みシステム分野において、Linux のような位置付けとなるOSの構築を目指す！*

### プロジェクトの推進主体

- ▶ 産学官の団体と個人が参加する産学官連携プロジェクト
- ▶ 2003年9月にNPO法人として組織化
- ▶ それ以前は、豊橋技術科学大学 名古屋大学 高田研究室を中心とする任意団体として活動

## TOPPERSプロジェクトの狙い

**!** 日本の主要産業分野で重要な役割を果たしている組み込みシステム分野で、日本独自のITRON仕様の技術を維持・発展させていきたい

### 現世代のリアルタイムOSの決定版の構築

- !** 約20年間に渡るITRON仕様の技術開発成果をベースに
- ▶ ITRON仕様の標準的なオープンソース実装を用意することで、企業の開発投資をより先端的なソフトウェア部品や開発環境の開発に向ける
  - ▶ ITRON仕様の実装が絞られることで、ソフトウェアの移植性が向上し、それらにサポートが集中する
- ↓
- ▶ ITRON仕様がかかえる「過剰な重複投資」と「過剰な多様性」の問題が解決 (または軽減)

## 次世代のリアルタイムOS技術の開発

- ▶ 組み込みシステムの要求に合致し，ITRONの良さを継承した，次世代のリアルタイムOS技術を開発する
  - Linuxと類似のOSをもう1つ作っても意味がない！**
  - ▶ 汎用OS向けに開発された技術をそのまま導入するのではなく，組み込みシステムに向けた技術を開発する
- ▶ オープンソースソフトウェア化により，産学官の力を結集することが可能に
- ▶ 標準仕様を策定 (ITRON仕様のアプローチ) するよりも，ソフトウェアを開発した方がスピードが速い

## 組み込みシステム技術者の育成

- ▶ オープンソースソフトウェアを活用した教材の提供や，教育の場を設けることで，組み込みシステム技術者の育成に貢献する

## これまでの主な開発成果

### TOPPERS/JSPカーネル

- ▶  $\mu$ ITRON4.0仕様のスタンダードプロファイルに準拠したリアルタイムOS
- ▶ 2000年11月に公開．現在 Release 1.3

### IIMPカーネル

- ▶ JSPカーネルをベースに，メモリ保護機能を追加したリアルタイムOS． $\mu$ ITRON4.0/PX仕様に準拠

### IDLカーネル

- ▶ IIMPカーネルをベースに，モジュールの動的なダウンロード機能を追加したOSと関連ソフトウェア

### TINET

- ▶ ITRON TCP/IP API仕様に準拠したコンパクトなTCP/IPプロトコルスタック．JSPカーネル上で動作

## TOPPERS/JSPカーネルの利用事例

- ▶ 松下電器産業のカラオケマイク「DO!KARAOKE」(商品型番: SY-MK7)



パナソニック SD カラオケマイク「SY-MK7-S」  
デュエットマイク「SY-DK7-S」  
(2003年2月 松下電器)

## 組み込みシステムにおけるオープンソースソフトウェア

### オープンソースソフトウェア利用の現状

- ▶ 開発ツールとランタイムソフトウェア (OS, ミドルウェアなど) とで若干事情が異なる
  - ! 組み込みシステムに限ったことではない

### 開発ツール

- ▶ 開発ツールは最終製品には組み込まれないので, ライセンス条件や知的財産権に関する問題は, それほどシビアではない
- ▶ GNU開発ツール (GCC, GDB など) は, 組み込みシステム向けの有力な開発環境の一つとして, 以前から広く利用されている



## ランタイムソフトウェア (OS, ミドルウェアなど)

- ▶ ラインタイム (実行時) ソフトウェアは最終製品に組み込まれることから, ライセンス条件や知的財産権に関する問題がシビア (後述)
- ▶ Linux の組み込みシステムへの適用が広がる
  - ▶ Emblix (Japan Embedded Linux Consortium) が2000年6月に発足. 多くの会員が集まる
  - ▶ 組み込み Linux をビジネスにする会社が増える (ベンチャ企業も生まれる)
  - ▶ この1~2年に, 製品への適用事例が急速に広がる (シェアが10%強に)
- ▶ その他のオープンソースOS (ITRON仕様も含む) の適用事例も徐々に出つつある
  - ▶ シェア的にはまだ大きくない (5%弱程度)

## オープンソースのビジネスモデル

! 組み込みシステム分野では，オープンソースソフトウェアをベースにしたビジネスモデルが成立しやすい

例) 旧Cygus Solutions (Red Hat が吸収) のビジネスモデル

- ▶ GNU開発ツールに対するサポートビジネス
  - ▶ バグがあった場合には， $n$ 日以内に対応します
  - ▶ 世界中に分散したエンジニアで，分散サポート体制
    - ➔ 実は，売上げの半分に満たない
- ▶ GNU開発ツールのポーティングビジネス
  - ▶ 新しいプロセッサ $X$ のためのGNU開発ツール一式 (GCC, GAS, GDB, シミュレータ, ライブラリ) を開発します
    - ← 組み込みシステムにおけるプラットフォームの多様性

## 組み込みシステム分野におけるライセンスの問題

### ライセンス条件の重要性

- ▶ ライセンス条件が，ソフトウェアの採用しやすさを決める大きな要因であるのは言うまでもない
- ▶ さらに，周辺ビジネスとして許されるビジネスモデルの枠を決めるものである

### ビジネスモデルを縛るルール

### 組み込みシステムの事情

- ▶ 組み込みシステムにおいては，コンピュータのハードウェアやOSは，(直接的には) 製品の価値に影響しない
  - ▶ 「ITRON仕様OSを使っているデジタルカメラが欲しい」ということは (普通は) ない
- ▶ どんなプロセッサやOSを用いているか (組み込みシステムの内部) は，公表されないのが普通

## GPL (GNU General Public License)

- ▶ GPLのプログラムとリンクしたプログラムのソースコードは、GPLで公開する義務がある
- ▶ 組み込みシステムの内部は公表したくない。まして、ソースコードは出したくない
  - ▶ 知的財産権の流出 (競合会社に使われてしまうおそれ)
  - ▶ ノウハウの漏洩
  - ▶ 知的財産権侵害が見つかりやすい!?
- ▶ オープンソースソフトウェア提供者側にとっては、ただ乗りされにくいという利点のあるライセンス条件
- ▶ ライセンス条件の曖昧性が最大の問題
  - ▶ GPLプログラムと動的にリンクしたものは公開義務なし? (GPL汚染される範囲はどこまでか?) 組み込みシステムでは、全体を1つに静的リンクする場合も

## BSDライセンス

- ▶ GPLよりはるかに自由に利用できるライセンス条件
- ▶ ただし、バイナリ形式で再配布する場合に、製品マニュアル等にソフトウェアについている著作権文言や無保証規定を転載する必要があり面倒
- ▶ 旧版のBSDライセンスは、宣伝条項 (製品の宣伝資料にも使っていることを書かなければならない) があり、さらに面倒
- ▶ 例えば、自動車のマニュアルにBSDライセンスの文言を載せることは可能か？
  - ▶ 載せること自身の害はない (小さい字にすればOK? 秘密にしたいという問題は残るが...)
  - ▶ 載せるようにする社内調整は容易ではない
    - ➔ 採用担当者としては敬遠したい

## TOPPERSライセンス

### ライセンス条件設定にあたっての基本的な考え方

- ▶ 組み込みシステム開発に広く利用してほしい
    - ▶ プロジェクトの狙いを達成するためには、広く利用していただくことが必要
    - ▶ 組み込みシステムの事情を考慮したライセンス条件
  - ▶ 周辺ビジネスが成立しやすいようにしたい
    - ▶ 産業振興と技術発展がプロジェクトの最終目的
  - ▶ プロジェクトの成果を把握したい
    - ▶ 公的資金を使って開発している以上、説明責任 (アカウンタビリティ) がある
    - ▶ 成果のアピールが、他での採用や予算獲得につながる
- !** ITRON仕様からの反省

## アプローチ

- ▶ TOPPERSプロジェクトで独自に開発したソフトウェアには，独自のライセンス条件を設定する
- ▶ 機器に組み込む場合には，利用の報告さえしてもらえれば，自由に使ってよいものとする
  - “ レポートウェア ”
    - ▶ 松下電器のカラオケマイクの適用例は，この報告義務に基づいて報告があったもの
    - ▶ 要請があれば，報告内容の守秘には応じる
- ▶ 派生したソフトウェアをオープンにする義務は課さない
  - ただ乗りOK
- ▶ GNUソフトウェアとリンクして使うことを許すように，GPLとのデュアルライセンスに (報告義務が，GPLコンパチブルでないという指摘)

## ライセンス文言の構成

- ▶ ソフトウェアの名称と著作者の著作権表示
- ▶ 利用条件本体 (次のスライドに掲載)
- ▶ 無保証規定

## 指摘されている弱点

- ▶ 報告義務が難しいという指摘
  - ▶ TOPPERSの開発成果物をベースにしたリアルタイムOSをユーザに提供する場合に，報告義務が理解してもらいにくい
  - ▶ 説明義務のあるリアルタイムOSしか持っていないと，ビジネス上で不利になる可能性も !?

結局は...

**開発者側の利益と利用者側の利益のトレードオフ**



上記著作権者は、以下の (1) ~ (4) の条件が、Free Software Foundation に  
よって公表されている GNU General Public License の Version 2 に記述され  
ている条件を満たす場合に限り、本ソフトウェア（本ソフトウェアを改変した  
ものを含む。以下同じ）を使用・複製・改変・再配布（以下、利用と呼ぶ）す  
ることを無償で許諾する。

- (1) 本ソフトウェアをソースコードの形で利用する場合には、上記の著作権表  
示、この利用条件および下記の無保証規定が、そのままの形でソースコー  
ド中に含まれていること。
- (2) 本ソフトウェアを、ライブラリ形式など、他のソフトウェア開発に使用で  
きる形で再配布する場合には、再配布に伴うドキュメント（利用者マニユ  
アルなど）に、上記の著作権表示、この利用条件および下記の無保証規定  
を掲載すること。
- (3) 本ソフトウェアを、機器に組み込むなど、他のソフトウェア開発に使用で  
きない形で再配布する場合には、次のいずれかの条件を満たすこと。
  - (a) 再配布に伴うドキュメント（利用者マニュアルなど）に、上記の著作権表  
示、この利用条件および下記の無保証規定を掲載すること。
  - (b) 再配布の形態を、別に定める方法によって、TOPPERSプロジェクトに報  
告すること。
- (4) 本ソフトウェアの利用により直接的または間接的に生じるいかなる損害か  
らも、上記著作権者およびTOPPERSプロジェクトを免責すること。

## T-Kernelのライセンス条件

### T-Kernelとは？

- ▶ T-Engineプロジェクトで開発しているリアルタイムOS
- ▶ T-Engineアーキテクチャの重要な一部分 (T-Kernelを使わないものは、T-Engineとは呼ばない)

### T-Kernelのソースコード公開

- ▶ 年内にもソースコードが公開される予定
- ▶ 現時点ではライセンス文言が公表されていない

### ライセンス条件設定にあたっての基本的な考え方

- ▶ T-kernelの目標はミドルウェアの流通促進
- ▶ 組み込みシステムに適したライセンスとする
- ▶ T-kernelはオープンソースソフトウェアであると同時に標準化規約の一部である

## アプローチ (まだ固まっていない)

- ▶ シングルソースは堅持する (ミドルウェアの流通性を確保するためには、派生物が出まわるのは望ましくない)
- ▶ 変更部分を公開する必要はない (組込みシステムでは最適化が不可欠である。自社内で利用するために変更するのは可能)
- ▶ バイナリのみでの配布も可能
- ▶ クローズソフトウェアとの共存も可能
- ▶ 公式版は T-Engine Forum からリリース

このページと前のページの内容は、越塚登氏 (東京大学助教授, YRPユビキタスネットワークング研究所副所長) からの聞き取り調査によるものです。

## オープンソースの開発における知的財産権の扱い

! ライセンス条件と並んで、オープンソースに関わる大きな法的課題

**SCOによる対Linux訴訟**

### 知的財産権に関する問題の本質

- ▶ ボランティア開発者の力を活用できることは、オープンソースの重要な利点の1つである
- ▶ オープンソースソフトウェアが、第3者の知的財産権を侵害していないことを保証することは極めて困難な課題
- ▶ 知的財産権を侵害を完全に排除しようとする、ボランティア開発者の貢献を期待することは難しくなる
- ▶ とはいえ、利用者のためには何らかの対策が必要

ここでも結局は...

**開発者側の利益と利用者側の利益のトレードオフ**

## FSF (Free Software Foundation) のモデル

- ▶ FSFは、GNUソフトウェア (GNU開発ツール, GNU Emacs などなど) を管理している団体
- ▶ コントリビュータ (自分の開発したソフトウェアをGNUソフトウェアに入れたい開発者) は、FSFに著作権を譲渡する
  - ▶ FSFが権利関係を一元管理できるようにするのが目的
  - ▶ その際に開発者は、知る限りにおいて他者の知的財産権を侵害していないことを誓約する
  - ▶ 開発者の雇用者 (いる場合) の同意書も求める
  - ! ただし、現在このプロセスは公開されておらず、詳細は不正確かもしれない
- ▶ ソフトウェアが第3者の知的財産権を侵害しないための仕組みがある。ただし、開発者側にやや厳しい面がある

## Linuxカーネルのモデル

- ▶ Linuxカーネルの公式版は，コア開発チームが管理
- ▶ コントリビュータの開発したソフトウェアは，コア開発チームの手を経て，公式版に取り込まれる
  - ▶ コア開発チームは，技術的な内容はチェックしているが，知的財産権についてはチェックしていない
    - SCOによる対Linux訴訟のような事態を防ぐ仕組みがない **!黒かどうかは別**
- ▶ コントリビュータは，著作権を譲渡しない(ソフトウェアについている著作権表示が増えていく)
  - ▶ 譲渡することを宣言する書面はない

このページと前のページの内容は，新部裕氏(産業技術総合研究所，FSIJ 理事長)からの聞き取り調査によるものです。

## TOPPERSプロジェクトのモデル

### 基本的な考え方

- ▶ 開発者の貢献しやすさと利用者の使いやすさを折衷させる
- ▶ なるべく簡便なプロセスにする (ソフトウェアをコントリビュートする度にサインするのは避ける)

### アプローチ

- ▶ コントリビュータは、プロジェクトの会員に限定する。プロジェクト内で会員の行動規範を定める「開発成果物の知的財産権に関する規則」を制定
- ▶ 著作権 (侵害していることが自覚できる) と工業所有権 (知らずに侵害する場合がある) を区別して考える
- ▶ コントリビュータは、著作権を譲渡しない (ソフトウェアについている著作権表示が増えていく)

## 規則の関連部分 (重要な部分のみを抜粋)

第2条 TOPPERSプロジェクトの会員（以下、「会員」という。）が開発し、次のいずれかに該当するソフトウェアを、TOPPERSプロジェクトの開発成果物（以下、「開発成果物」という。）と呼ぶ。

(1) TOPPERSプロジェクトの開発計画の一環で開発したソフトウェア

(2) TOPPERSプロジェクトが、それを開発した会員より、開発成果物として取り扱う旨の合意を得たソフトウェア

第5条 会員は、自らが開発する開発成果物において、他者の著作権を侵害してはならない。

第6条 会員は、自らが開発する開発成果物に、自らが所有する工業所有権（特許権など）を利用する場合には、TOPPERSプロジェクトにその旨を通知するとともに、開発成果物を利用する場合に限って、当該工業所有権の実施を無償で許諾しなければならない。

第7条 会員は、開発成果物が何らかの知的財産権を侵害していることを発見した場合には、TOPPERSプロジェクトに直ちにその旨を報告しなければならない。



## 規則のその他の内容

- ▶ プロジェクトの開発成果物と公式リリースの定義
- ▶ 知的財産権の帰属
  - ▶ 開発成果物の知的財産権は，開発者に帰属．プロジェクトに権利が移ることはない
- ▶ 開発成果物と公式リリースの利用条件
  - ▶ 開発成果物の利用条件は，可能な限りTOPPERSライセンスとする
  - ▶ 公式リリースの利用条件は，TOPPERSライセンスとする
- ▶ 早期リリース
  - ▶ 会員は 版・ 版を入手可能

## おわりに

### まとめ

- ▶ トロンプロジェクトにおけるオープンの意味を説明
- ▶ トロン関連のオープンソースソフトウェア開発プロジェクトであるTOPPESについて紹介．オープンソースプロジェクト運営における考慮点をケーススタディ
- ▶ 組み込みシステム分野におけるライセンスの問題点を議論
- ▶ オープンソースの開発における知的財産権の問題について議論

### 2つのメッセージ

- ▶ GNU GPLが重要なのは言うまでもないが、それだけがオープンソースライセンスではない
- ▶ ライセンスも重要だが、オープンソースの開発における知的財産権も大きい問題になってきた