半導体が創り出す未来社会 ③『マイコン・システム LSI』オープンソースを活用した共創開発の実像

先端 IT エレクトロニクス技術が支える未来（Full ver.)

宗像尚郎

ルネサスエレクトロニクス株式会社
ハイパフォーマンスコンピュート ソリューショングループ
シニアダイレクタ

2023-12-20

自己紹介 (Who am I)

総合半導体メーカーで デバイス制御用の SW ソリューション開発 に従事しています

- ルネサスエレクトロニクス株式会社
  - 日立、三菱、NEC の半導体事業が統合、グローバルな半導体専業メーカーに発展
  - 自動車向け 大規模ソフトウエア基盤（OS、プラットフォーム）開発 に従事
  - 社内のオープンソース開発活動（Linux kernel 開発など）を管掌
- オープンソース開発コミュニティ（会社公認の社外活動）
  - The Linux Foundation 元理事（Board of Director）
  - AGL (Automotive Grade Linux) プロジェクト、yocto プロジェクト理事
  - COVESA（Connected Vehicle System Alliance）理事
  - 社外講師、講演、インターフェース誌などへの投稿など

半導体ビジネスにおいても「オープンソースの活用」は重要なテーマになっています

SOC を動かす為には、専用の SW が必要

SoC (= System on Chip) とは、どのような半導体なのか？

汎用プロセッサ vs. システムオンチップ

汎用プロセッサー

- マイクロプロセッサー（マイコン）
  - 8bit/16bit が主流、32bit も登場
  - 100pin 以下の比較的小規模なもの
  - 周辺機能を取り込んだ用途別設計
  - リアルタイム制御にも利用される
- 汎用プロセッサー (PC 用など)
  - 64bit が主流、1,000pin 規模
  - BIOS が HW 差異を吸収する
  - 汎用 OS（Windows/Linux 等）
  - コンパニオンチップと組み合わせ

システムオンチップ (SoC)

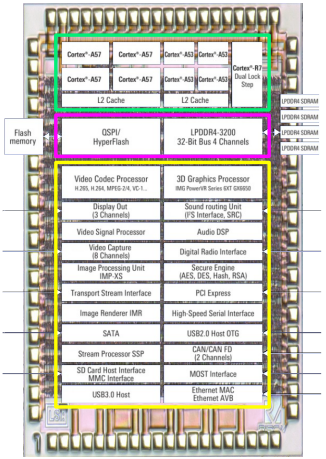
- 特定用途向け大規模集積回路
  - システムの大部分を内包
  - 専用回路（GPU、NPU 等）
- 台数が期待できる特定用途向け
  - スマートフォン、タブレット
  - デジタル家電（テレビなど）
  - 自動車（ナビ、運転支援）
- SoC には 専用の制御用 SWが必要
  - デバイスドライバー、起動手段

SoC は用途別の専用回路を実装したものなので、専用の制御用 SW が必要になる

SoC にはどのような機能が含まれるのか （自動車用の SoC の例）

システム回路全体が実装 された 大規模集積回路 (VLSI)

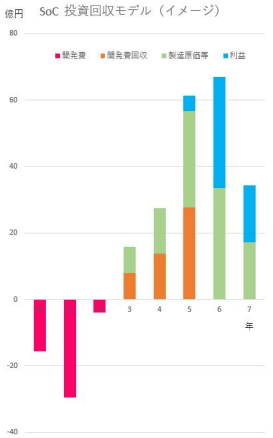
- プロセッサーコア [緑枠]
  - マルチコア（並列処理）
  - ヘテロコア（異種混合）
- メモリーインターフェース [マゼンダ]
  - DDR メモリーコントローラー
  - オンチップメモリー（キャッシュ RAM、ROM）
- 各種機能ブロック (IP=intellectual property) [黄色]
  - GPU（表示コントローラー）
  - NPU（AI　アクセラレータ）
  - USB、PCI、SD カード　など
- 受動部品（抵抗、コンデンサー　など）



SoC ビジネスの特性 （線径微細化に伴って、参入障壁は拡大傾向）

SoC の開発には 数年の開発期間 と 莫大な開発費用 が必要

- 設計開発期間
  - 回路規模の拡大により 設計検証期間は長期化 の傾向
  - 動作周波数高速化により 熱設計（パッケージ）が課題
  - 総開発費に占める SW の比率が拡大（現状、約４割）
- 製造期間
  - マスク数増大により ウエハー製造には一か月以上 かかる
  - 必要部材が多く、サプライチェイン問題 が顕在化した
  - 微細化により 受託製造メーカ（ファウンダリー）が台頭
- 投資回収
  - SoC 開発には 数十億円規模の初期開発投資が必要
  - 量産開始までの期間の投資を支える 資金調達力 が必要
  - 初期投資分を回収し利益を出す には、数量規模 が必要



何故半導体が「SWの開発力強化」を推進しているのだろうか？

競争軸が「コンポーネント」から「ソリューション」へシフトしている

ウィニング・コンビネーション

アナログ+パワー+組込みシステム+コネクティビティなど、ルネサスの横断的な製品ポートフォリオを組み合わせたトータルソリューションの数々がご覧になれます。


ワイヤレススマートロック


ユニバーサルNFC充電器


USB IO-Linkマスタ


自動車用ウィンドウコントロール


サラウンドビュー& AR-HUD用ビデオ出力拡張ソリューション


車載コックピット ハプティクスソリューション


接続型Androidクラスタ


AHLを備えたフルデジタルクラスタソリューション

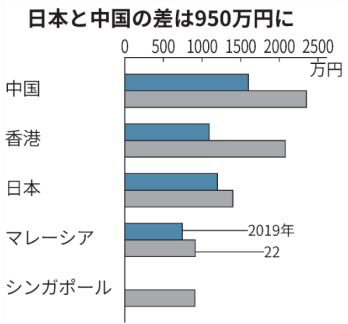
https://www.renesas.com/jp/ja

半導体メーカーの競争の主戦場が「システムソリューションの提案力」に移行している

世界の人材市場で「高度 IT 人材」の価値が高騰している

半導体業界も SW 人材の強化が喫緊の課題になっている

- Shift Left = SW の先行開発
- SW First = SW の再利用性の担保
- CICD (Continuous Integration, Continuous Delivery) = SW 開発 → 検証 → リリースの自動化
- Open Innovation = OSS の活用
- Embedded AI = デバイスにも AI 機能搭載
- Cloud Native = デバイスもサーバー連携が必須
- SW Update = 機能アップの為の頻繁な SW 更新
- Cyber Security = SW 脆弱性対策の義務化



(注)データサイエンティストの最高額。19年のシンガポールはなし
(出所)ヘイズ

https://career.nikkei.com/nikkei-pickup/002496/

半導体メーカーにおける「HW 開発」と「SW 開発」の違い

ハードウエア開発 → 高性能化、超複雑化

- 微細化により 論理規模が巨大化
  - 検証工数 も肥大化 → 効率化が課題
  - 発熱対策 が大きな課題
  - チップレット など実装面での革新
- 最先端プロセスの開発・製造コスト
  - 適用可能な製品群は限定される
- 性能競争
  - CPU 能力（並列化、高速化）
  - GPU や NPU (=AI) エンジン 性能
- RISC-V などオープン規格の潮流

ソフトウエア開発 → エンドレスゲーム化

- SW の規模が爆発的に拡大中
  - メカ制御 → SW 制御 の適用拡大
  - ネットワーク（クラウド連携 など）
  - AI による推論機能 の実装
- 組み込み機器も SW 更新 が必要
  - 機能追加 で製品の魅力を維持
  - 脆弱性対策 パッチの適用
  - HW 収束後も SW 開発は継続必至
- OSS の適用範囲 が拡大（Linux など）
  - 開発 コミュニティとの連携 も必要に

製品出荷後の SW 更新（機能強化、セキュリティ対策）費用の回収スキームは？

Open Source Software (=OSS) とは

オープンソースの 誕生 と 発展 の歴史

何故わざわざ「オープン」とか「フリー」なソフトと呼んでいるのか?

オープンソース ソフトウエア、フリー ソフトウエアの意味するもの

- 「オープン」＝文字通り開かれているという意味
  - ソースコード にアクセスでき、改造や再配布 が許されている
  - 開発プロセスが公開されていて、誰でも参加 できる
  - 著作権（後述）による縛りから免除 されている
- 「フリー」には「無料」と「自由」の意味があるが、ここでは「自由」の意味
  - ソフトウエアにおける「自由」とは何を意味するのか？
  - Richard Stallman は ３つの自由 が担保される必要があると主張した
  - Stallman の GNU Free Software 運動 が SW の共創開発という概念の起点

ソフトウエアは本来は「開かれていなくて」「不自由」な物なのであえて区別している

1983 年の Stallman に何がおこったのか

- 当時 Stallman は MIT のロボット研究所に所属
- FBI がハッカー集団 414 を逮捕、社会の脅威とされた
- ACM (Association for Computing Machinery) がアンチハッカー宣言、 一連の記事を掲載
- Stallman はハッカーの本質は違うと主張 するも著作権自体を否定していると誤解を受ける
- ACM 学会誌上で議論となり、MIT ラボを辞任
- 1985 年 3 月に GNU 宣言を発表

http://ergoemacs.org/emacs/i/Richard_Stallman_at_MIT_dancing_1970s.jpg

Stallman は 社会貢献を企図して「フリー ソフトウエア」 を提起した

フリー ソフトウエア（= 自由ソフトウエア）とは 4 つの自由 が担保されたもの

- どんな目的に対しても、プログラムを望むままに実行する自由 (第零の自由)。
- プログラムがどのように動作しているか 研究し、必要に応じて改造する自由 (第一の自由)。ソースコードへのアクセスは、この前提条件となります。
- ほかの人を助けられるよう、コピーを再配布する自由 (第二の自由)。
- 改変した版を他に配布する自由 (第三の自由)。これにより、変更がコミュニティ全体にとって利益となる機会を提供 できます。ソースコードへのアクセスは、この前提条件となります。
- https://www.gnu.org/philosophy/free-sw.ja.html

GNU Project Origin (GNU Manufesto)

GNU宣言

下記のGNU宣言はリチャード・ストールマンによって、1985年にGNUオペレーティング・システムの開発に支持を求めて、書かれたものです。1987年までは、開発を説明するのに少々更新されましたが、それからは、変更なしのままで置いておくのがよいでしょう。

その時から、よくある誤解について、そしてそれが異なる言い回しで避けられることを、わたしたちは学びました。1993年から脚注を付け足し、この点を明確にしました。

GNU/Linuxシステムをインストールしたい方には、わたしたちは100%自由ソフトウェアのGNU/Linuxディストリビューションのひとつを使うことをを推奨します。どのように貢献するかについては、http://www.gnu.org/helpをご覧ください。

GNUプロジェクトはソフトウェアのユーザの自由のキャンペーンである自由ソフトウェア運動の一部です。GNUを「オープンソース」の用語と関連付けるのは間違いです。この用語は1998に自由ソフトウェア運動の倫理的な価値に同意しない人々によって思いつかれたものです。かれらは同じフィールドで倫理とは無関係のアプローチを推進するのにこの用語を使っています。

https://www.gnu.org/gnu/manifesto.ja.html

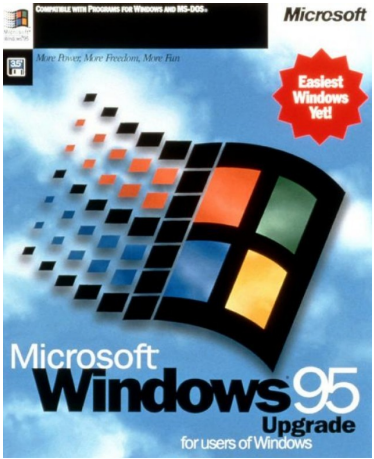
「情報化社会」は 1995 年にスタート していることに注目しよう

Windows95 に初めてインターネット接続機能（ブラウザー）がバンドル された

- 一般のユーザーが Web ページ を介したデジタル情報の取得を始める
- 同時に SAPM やコンピュータウィルス問題 が社会問題化
- Software Update の提供が始まった (Windows95 以降)
- 携帯音楽プレイヤー（iPod は 2001 年に登場）
- 仮想化技術 の普及（Xen は 2003 年、KVM は 2006 年に誕生）
- クラウドコンピューティング環境 の登場（オンラインストレージは 2006 年から）
- スマートフォン の登場（iPhone 第一世代は 2007 年に登場）
- サブスク方式のアプリ提供（Office365 は 2011 年に登場）

ブラウザーをめぐって、オープンソース運動が本格的に広がりをみせていく

ブラウザー対決（Internet Explorer 対 Netscape Navigator）



Netscape のコード公開が (anti MS としての) OSS 活性化を加速

c|net COVID-19 GIFT GUIDE BEST PRODUCTS REVIEWS NEWS HOW TO FINANCE HEALTH SMART HOME

Netscape sets source code free

After three months of anticipation, Netscape Communications releases the source code for its Communicator suite.

Janet Kornblum March 31, 1998 12:10 p.m. PT

After three months of anticipation, Netscape Communications today finally released the source code for its Communicator suite.

Netscape this morning unveiled the much-anticipated release with a teleconference featuring breathy executive statements touting the significance of the move. The company actually posted the approximately 8 megabytes of compressed Communicator 5.0 code at 10 a.m. PT to Mozilla.org, the site Netscape has set up to be the central clearinghouse for source code-related information.

Netscape CEO Jim Barksdale took this photo in France a month after the company released source code for its Communicator suite.

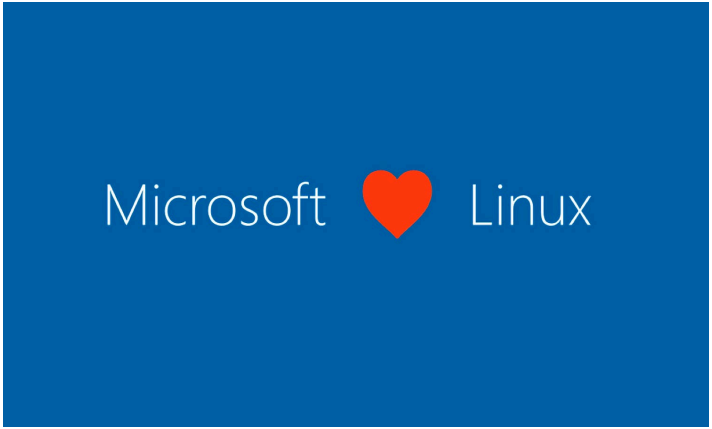
Alain BUU

https://www.cnet.com/news/netscape-sets-source-code-free/

この頃の Microsoft は、徹底したアンチ Linux の立場だった

https://gigazine.net/news/20070520_anti_linux_propaganda/

マイクロソフトは Linux が大好きだ (by CEO Satya Nadella, 2015)



https://www.microsoft.com/ja-jp/cloud-platform/Windows-Server-blog-Loves-Linux.aspx

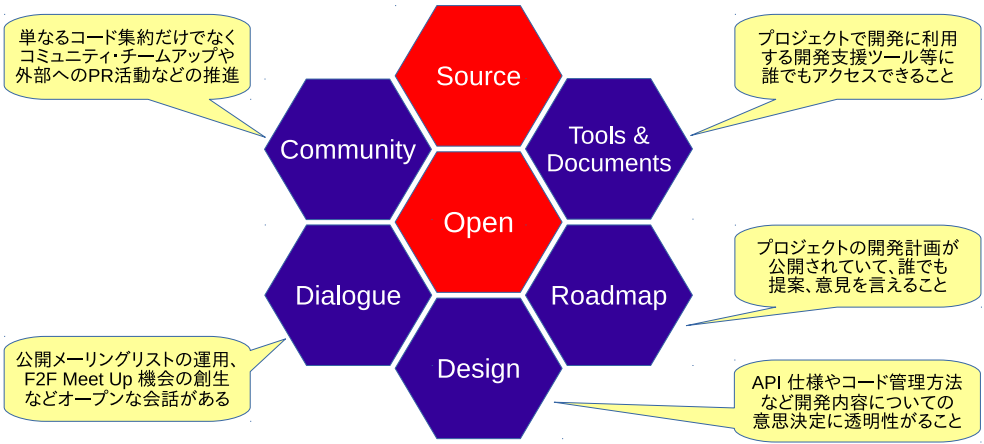
オープンソース開発コミュニティの実像

オープンソース開発コミュニティとは

オープンソースとの関わり方 には いくつかの段階 がある

- ユーザー（＝使うだけの人）
  - 無料で使えてうれしい、寄付 を要請されたけどどうしようかな
  - サポートとか、セキュリティ対策とか は大丈夫かな
- デベロッパー（＝開発に参加する人）
  - プログラムを改造してみたいけど、ソースコード はどこにあるのかな？
  - 機能追加やバグ対策の パッチを書いたけど、受け取ってもらえる のかな？
- プロジェクトメンバー（＝プロジェクトの成長を考える人）
  - 今後の リリース計画 をどうしよう？　ユーザーは何を望んでいるかな？
  - 中長期的にプロジェクトを 安定運営させためのリソース（人、金） をどうしよう？

コミュニティで影響力を行使するには コードの公開だけでは不十分

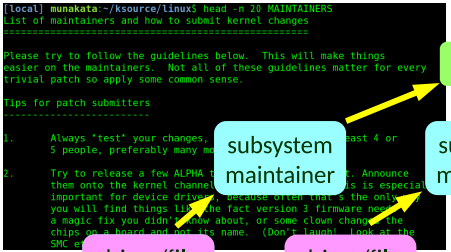


Linux kernel 統計情報（2023-10-29 リリースの kernel 6.6 実績)

項目	集計結果
ソースコード行数（コメント行を除く）	24,777,968 行
ver 6.5 → ver 6.6 の差分（増分）	163,028 行
ファイル数（ディレクトリーを除く）	82,189 個
sloccount によるソフトウエア価値の試算	$ 1,110,223,960（1兆6,653億円）
ver 6.6 の開発期間	63 日（9 週間）
ver 6.6 の開発に参加した開発者	1,978 人
今回はじめて開発に参加した開発者	249 人
追加された機能（= change set）	14,069 個
ver 6.6 に追加された価値 (sloccount 差分）	$ 7,668,754（11.5 億円）

https://lwn.net/Articles/948970/

OSS 開発プロセス（１）：階層化された開発コミュニティの構造



Linus Torvalds

Linux-next

subsystem maintainer

subsystem maintainer

subsystem maintainer

driver/file maintainer

driver/file maintainer

driver/file maintainer

driver/file maintainer

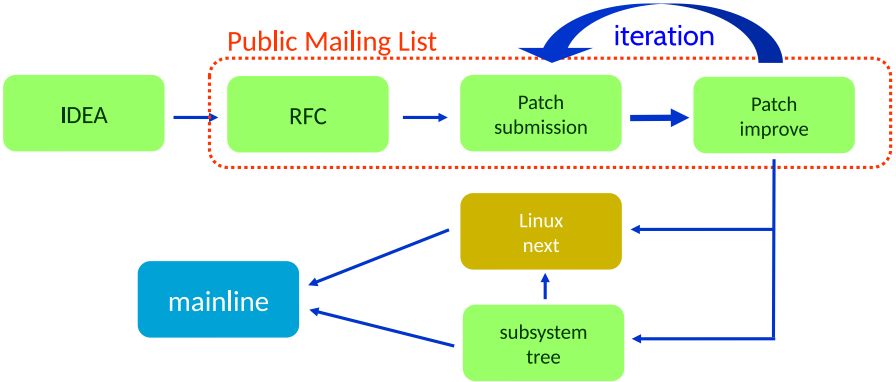
user/developer

user/developer

user/developer

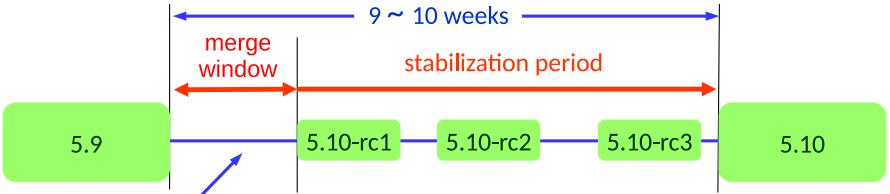
user/developer

OSS 開発プロセス（２）：公開 ML を使ったオープンな議論



公開のML 上で新提案をオープンに 検討→改善 を重ねる（イタレーション）、
どんな経験者でも数回の書き直しが要求される（顔パスは無い、皆で推敲して良いコードにする）

OSS 開発プロセス（３）：パッチのレビューとマージ



- Linux-next の段階で毎日 kernel build を実施して マージで問題が出ないことを確認しておく
- 一つ前のカーネルがリリースされると、Linux-next の コードを取り込む2週間のマージウィンドウが開く
- RC1 リリース後は新規マージを停止。安定化のため のシステム検証を行う。 通常は RC8 ~ 9 までいく

Linux next

ML discussion (previous page)

パッチレビューに合格して Linux-next に登録されてからマージが完了するまで 最大で19週 かかる

OSS のクオリティの源泉は 徹底したコードのピアレビュー（査読）

周期的なバージョン更新を確約することによって 締め切り間際での無茶を徹底排除

- オープンソースプロジェクトではコミュニティ開発者から投稿されたパッチを集約してプログラムを構築するので 低品質なコードが混入しないよう 入り口での制御 (コードレビュー) が重要な砦となる。
  - 組織的なコードレビューの仕組み (Gerrit など) があるか
  - メンテナーはタイムリーにパッチをレビューしているか、
  - レビュー内容に”透明性”や”中立性”があるか
  - メンテナーの技術バックうラウンド、スキルセットは適切か
  - 幅広く関連する別プロジェクトの開発者もレビューに参加しているか
- パッチレビューでは ベテラン開発者でも最低 3 回程度は書き直す のが普通

各界の識者による徹底したピアレビューによってサスティナブルなコードを生み出す

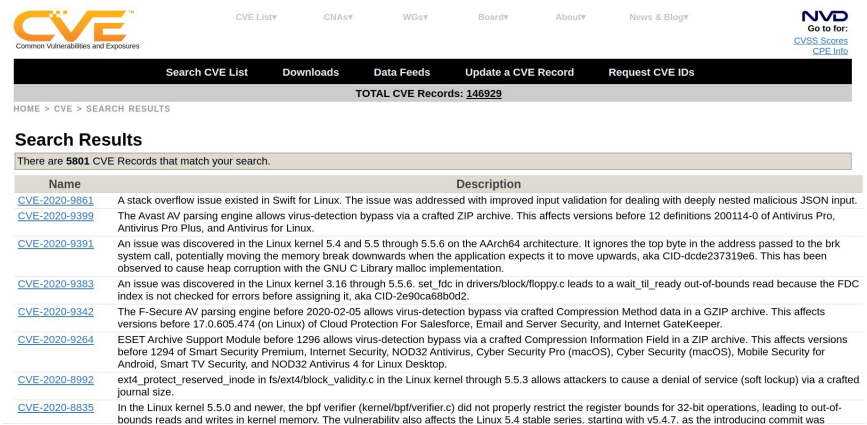
SW メンテナンス (バグレポート、対応状況管理、パッチリリース)

完全無欠なソフトウエアは無い、永遠にメンテナンスを続ける覚悟が必要

- プロジェクトにプログラムの 品質保全体制 (バグ報告 → 対策 → 来歴管理) が適切に運用できているかを確認する必要がある。実際の運用状況はまちまちである。
  - 公開バグ管理システム (=BTS) にバグ報告が集められているか
  - プロジェクト内でバグの原因究明と対策パッチ作成が行われているか
  - バグ修正パッチがタイムリーにリリースされているか
  - バージョンアップ時に過去バグの 累積的なリグレッションテスト の実施
- ローカルにバグ修正をおこなった場合、その内容をマスターコードにも反映しないと品質があがらない (= 将来同じ問題が再発する可能性がある)
- 組織的なバグ管理 (報告、履歴管理、リグレッションテスト等) 体制が必須

(会社にはとても嫌われるだろうが) SW 開発はエンドレスゲームと考える必要がある

脆弱性情報の公開 = Common Vulnerabilities Exposures (CVE)



CVE
Common Vulnerabilities and Exposures

CVE List▾ CNAs▾ WGs▾ Board▾ About▾ News & Blog▾

NVD
Go to for:
CVSS Scores
CPE Info

Search CVE List | Downloads | Data Feeds | Update a CVE Record | Request CVE IDs

TOTAL CVE Records: 146929

HOME > CVE > SEARCH RESULTS

Search Results

There are **5801** CVE Records that match your search.

Name	Description
CVE-2020-9861	A stack overflow issue existed in Swift for Linux. The issue was addressed with improved input validation for dealing with deeply nested malicious JSON input.
CVE-2020-9399	The Avast AV parsing engine allows virus-detection bypass via a crafted ZIP archive. This affects versions before 12 definitions 200114-0 of Antivirus Pro, Antivirus Pro Plus, and Antivirus for Linux.
CVE-2020-9391	An issue was discovered in the Linux kernel 5.4 and 5.5 through 5.5.6 on the AArch64 architecture. It ignores the top byte in the address passed to the brk system call, potentially moving the memory break downwards when the application expects it to move upwards, aka CID-dcde237319e6. This has been observed to cause heap corruption with the GNU C Library malloc implementation.
CVE-2020-9383	An issue was discovered in the Linux kernel 3.16 through 5.5.6. set_fdc in drivers/block/floppy.c leads to a wait_til_ready out-of-bounds read because the FDC index is not checked for errors before assigning it, aka CID-2e90ca68b0d2.
CVE-2020-9342	The F-Secure AV parsing engine before 2020-02-05 allows virus-detection bypass via crafted Compression Method data in a GZIP archive. This affects versions before 17.0.605.474 (on Linux) of Cloud Protection For Salesforce, Email and Server Security, and Internet GateKeeper.
CVE-2020-9264	ESET Archive Support Module before 1296 allows virus-detection bypass via a crafted Compression Information Field in a ZIP archive. This affects versions before 1294 of Smart Security Premium, Internet Security, NOD32 Antivirus, Cyber Security Pro (macOS), Cyber Security (macOS), Mobile Security for Android, Smart TV Security, and NOD32 Antivirus 4 for Linux Desktop.
CVE-2020-8992	ext4_protect_reserved_inode in fs/ext4/block_validity.c in the Linux kernel through 5.5.3 allows attackers to cause a denial of service (soft lockup) via a crafted journal size.
CVE-2020-8835	In the Linux kernel 5.5.0 and newer, the bpf verifier (kernel/bpf/verifier.c) did not properly restrict the register bounds for 32-bit operations, leading to out-of-bounds reads and writes in kernel memory. The vulnerability also affects the Linux 5.4 stable series, starting with v5.4.7, as the introducing commit was

https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Linux

サイバーセキュリティ等の脆弱性への対応

サイバーセキュリティ対応は SW 提供者の社会的責任 となってきている

- 新たに発見された ソフトウエア脆弱性 (＝セキュリティホール) に対して、タイムリーに対策パッチを提供するための組織的な取り組みが必要
- ソフトウエア脆弱性開示 (CVE) とのリンクした活動が行われているか
- リスクが公開されてから対策パッチが出るまでの時間が重要 (ゼロディ攻撃)
- 広く利用されているソフトウエアであってもコミュニティでの開発が既に終了していては脆弱性に対処できないケースもあった（OpenSSL など）
- サイバーセキュリティリスクなど重大なソフトウエア脆弱性が公開された時には 速やかに対策パッチを提供 することが求められる

多くの開発コミュニティが脆弱性対策に真摯に取り組んでいるが、支援も必要である

オープンソースの拠り所となる ライセンス を理解する

コンピュータプログラムは著作物 であり複製や改造には制限がある

著作権 (著作財産権、著作者人格権) の適用対象となるもの

- 言語 (論文、小説、脚本、詩歌、俳句、講演など)
- 音楽 (楽曲及び楽曲を伴う歌詞)
- 美術 (絵画、版画、彫刻、漫画、書、舞台装置など)
- 建築 (芸術的な建造物 (設計図は図形の著作物)
- 地図や図形 (地図と学術的な図面、図表、模型など)
- 映画 (劇場用／テレビ、ビデオソフト、ゲームソフト)
- 写真、グラビアなど
- コンピュータ・プログラム

著作権 (コピーライト) は明らかにオープンソース活動とは相いれない権利を主張する

著作財産権（= 狭義の著作権）: 対価の請求権 で第三者への譲渡も可能

著作権（財産権）

複製権	著作物を印刷、写真、複写、録音、録画などの方法によって有形的に再製する権利
上演権・演奏権	著作物を公に上演したり、演奏したりする(上演、演奏の録音物を再生することを含む)権利
上映権	著作物を公に上映する権利
公衆送信権・公の伝達権	著作物を自動公衆送信したり、放送したり、有線放送したり、また、それらの公衆送信された著作物を受信装置を使って公に伝達する権利 ＊自動公衆送信とは、サーバーなどに蓄積された情報を公衆からのアクセスに応じ自動的に送信することをいう。また、そのサーバーに蓄積された段階を送信可能化という。
口述権	言語の著作物を朗読などの方法により口頭で公に伝える(口述の録音物を再生することを含む)権利
展示権	美術の著作物と未発行の写真の著作物の原作品を公に展示する権利
頒布権	映画の著作物の複製物を頒布（販売・貸与など）する権利
譲渡権	映画以外の著作物の原作品又は複製物を公衆へ譲渡する権利
貸与権	映画以外の著作物の複製物を公衆へ貸与する権利
翻訳権・翻案権など	著作物を翻訳、編曲、変形、翻案等する権利（二次的著作物を創作する権利）
二次的著作物の利用権	自分の著作物を原作品とする二次的著作物を利用（上記の各権利に係る行為）することについて、二次的著作物の著作権者が持つものと同じ権利

参考条文...著作権法第21条～第28条

https://www.cric.or.jp/qa/hajime/hajime2.html

著作者人格権 = 著作者の心情（モラール）を保護する権利

著作者人格権は 第三者によるコードの改変や再配布を制約 する

- 公表権（著作権法 18 条 1 項）
  - 無断で公表されない権利、すなわち未だ公表されていない自分の著作物について、公表するかどうか、いつ、どういう方法及び条件で公表するかを決定する権利
- 氏名表示権（著作権法 19 条 1 項）
  - 自分の著作物を公表する際に、著作者名を表示するかどうか、どのように表示するか（実名で表示するのか、ペンネームなどの変名で表示するのか）を決定できる権利
- 同一性保持権（著作権法 20 条 1 項）
  - 自分の 著作物の内容、題号を著作者の意に反して無断で改変させない権利 です。

著作者人格権は財産権と異なり、第三者に譲渡したり相続したりはできない

オープンソースであると主張するためのミニマム条件

OSI (Open Source Initiative) が OSD (Open Source Definition) を規定した

- 1. 再頒布の自由
- 2. ソースコード (を入手できること)
- 3. 派生ソフトウェア（を作れて、同じライセンスを付与すること）
- 4. 作者のソースコードの完全性 (integrity)（が担保されること）
- 5. 個人やグループに対する差別の禁止
- 6. 利用する分野 (fields of endeavor) に対する差別の禁止
- 7. ライセンスの分配 (distribution)（に際し追加ライセンスが不要）
- 8. 特定製品でのみ有効なライセンスの禁止
- 9. 他のソフトウエアを制限するライセンスの禁止
- 10. ライセンスは技術中立的でなければならない
- http://www.opensource.jp/osd/osd-japanese.html

Netscape のブラウザー ソースコード公開のタイミングで、このルールが整備された

OSI が認定したオープンソース ライセンスの例

Open Source Licenses by Category

License Index

- License Approval Process
- License Information
- Origins and definitions of categories from the License Proliferation Committee report

In the lists below, a parenthesized expression following a license name is its SPDX short identifier, if one exists, except for two items in the first list (GNU General Public License and GNU Lesser General Public License). For these, the parenthesized expressions ("GPL" and "LGPL" respectively) are the common non-version-specific names of these licenses today (note also that the full name of the first version (2.0) of the LGPL is the GNU Library General Public License). There is no non-version-specific SPDX short identifier for the GPL and LGPL.

Licenses that are "popular and widely-used or with strong communities"

The below list is based on publicly available statistics obtained at the time of the Report of License Proliferation Committee.

- Apache License 2.0 (Apache-2.0)
- 3-clause BSD license (BSD-3-Clause)
- 2-clause BSD license (BSD-2-Clause)
- GNU General Public License (GPL)
- GNU Lesser General Public License (LGPL)
- MIT license (MIT)
- Mozilla Public License 2.0 (MPL-2.0)
- Common Development and Distribution License 1.0 (CDDL-1.0)
- Eclipse Public License 2.0 (EPL-2.0)

https://opensource.org/licenses/category

コードの開示義務 に関しては、OSS ライセンスに 2 つの類型 がある

コピーレフト型ライセンス

- 代表は GNU GPL/LGPL
- （共通）改変、再配布、利用を許諾
- ソースコードの開示義務あり
- Linux kernel など
- OSS コミュニティの活性化に寄与
- GPL は他のライセンスと結合できない

パーミッシブ型ライセンス

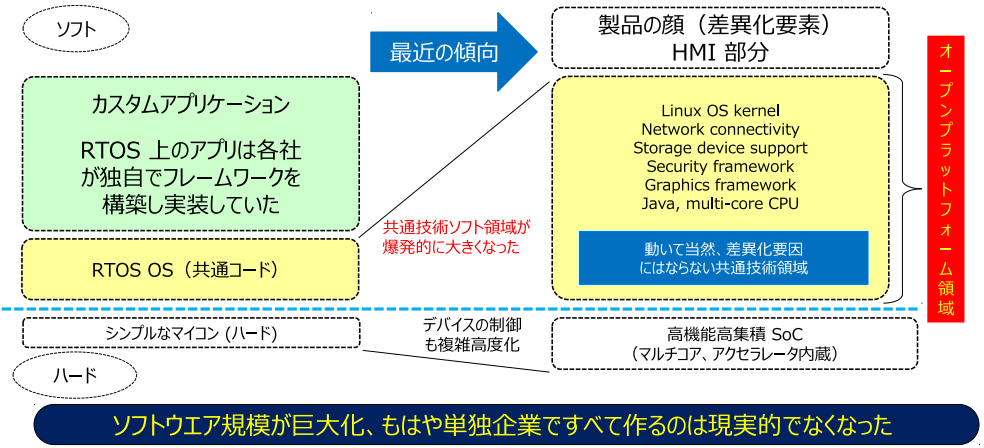
- 代表は MIT、Apache2
- （共通）改変、再配布、利用を許諾
- 改変したコードの開示を義務づけない
- Android、AI 関連ツールなど
- OSS の商用製品への適用を促進
- 一方で分岐（fork）が発生しやすい

近年は、産業界で使いやすい パーミッシブ型ライセンス が好まれる傾向がある

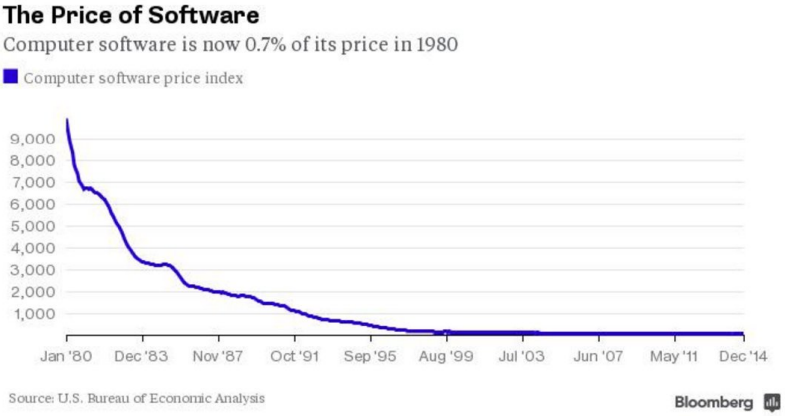
「共創開発」が産業界のトレンドに

何故今、世界のトップ企業が 「共創開発」 に注目しているのか？

商品差異化につながらない 非競争領域のソフトウエア比率が拡大



同時に ソフトウエアの低価格化 (無償化) も急速に進行している



本来、企業の研究開発は 自社の競争優位性を獲得 を目指した活動

研究開発（R&D = Research & Development）の目的は「競争優位性」の獲得

- アカデミア（学術界）
  - 基礎技術領域 では純粋な技術探求（= 何に役立つかが明確でない活動）も重視する
  - 研究成果は論文などを通じて公開され、成果は共有される
  - 産学連携活動 として産業界の意向を反映させた開発もある（増えている）
- インダストリー（産業界＝企業）
  - 企業の 競争優位性の源泉 としての 差異化技術 の獲得が動機
  - 明確な開発ロードマップと比較的短期間での開発成果の取り組みが期待される
  - 研究開発成果は 特許化して公開 するケースを除き 原則として非公開

各企業は自社 R&D は競争差異化の源泉だと考えるので、成果は「非公開」が常識

なのに 何故、共創開発（＝オープンな R&D 活動） に取り組むのか

多くの先端企業が「make or buy」から 「make or buy or collaborate」 に転換

- make 戦略 = 差異化技術については自社内でクローズに技術開発
- buy 戦略 = 非差異化領域の技術、時間が求められる技術は社外から技術導入
- collaborate 戦略 = 他社と連携して差異化技術のベースを共同開発
  - 開発規模が大きく自社だけでは必要なリソース（資金、開発者等）を集められない
  - オープンな活動とすることで、技術の中立性担保や開発スピードアップ を図る
  - 企業間の連携による マジョリティ確保でデファクトを獲得する
  - 中長期的にメンテナンス可能な サスティナブルなソリューションの構築
  - 企業同士で 似て非なる技術を重複開発するリスクを回避する

技術進化の速度、規模の拡大、グローバル化等のトレンドを反映した技術の新潮流

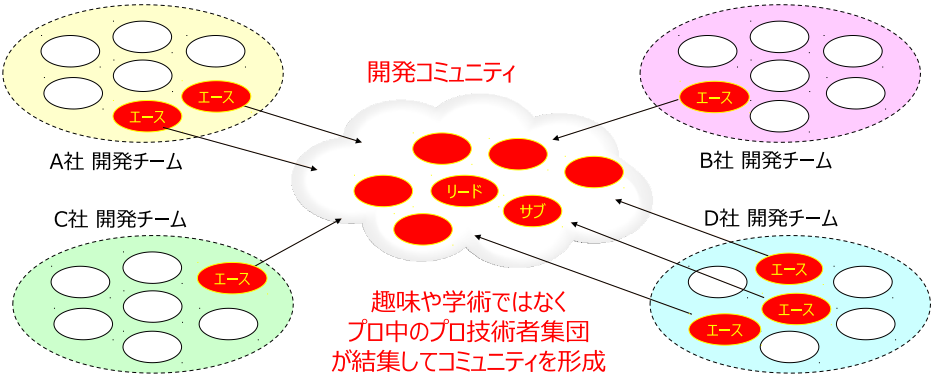
「デジュール スタンダード」から「デファクト スタンダード」の流れ

IT 業界では「リーダーシップ企業」の戦略によって「デファクト」が形成 されている

- デジュール（dejure = 原則上の）スタンダード
  - 公的な 技術標準化団体 であらかじめ決められた手順を踏んで方式を策定
  - 各企業の思惑などにより合意形成にはとても時間がかかる
  - ISO/IEC、IEEE、JIS などの標準化団体がある
- デファクト（de facto = 事実上の）スタンダード
  - ユーザーの選択によって結果的に主流となる
  - 公式な技術標準化よりも早くトレンドが決まることも多い（流行り廃りは激しい）
  - 企業はマーケティング活動、プロモーション活動を通じてデファクトを取りにいく

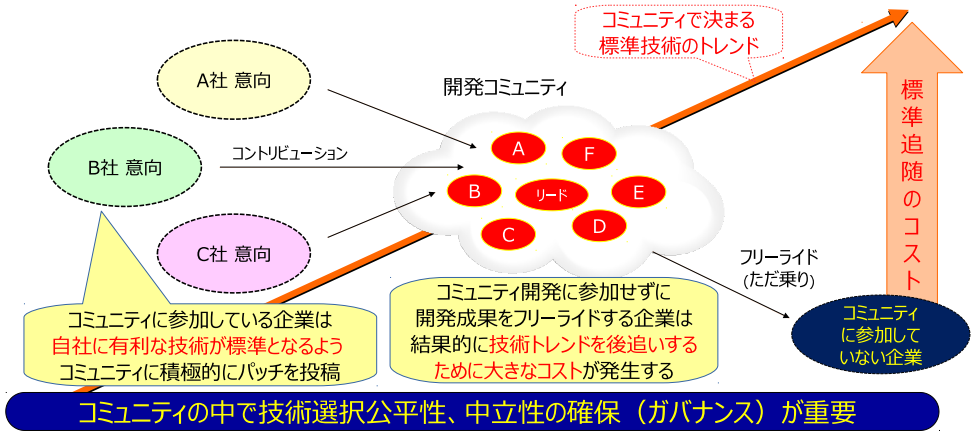
ハードウエアは「デジュール」が主流だが、ソフトウエアでは「デファクト」が拡大中

先行企業は 開発コミュニティに自社のエースを送りこんでいる



各社がエース級開発者をコミュニティに送り込んで共同で共通技術の開発を推進

何故なら Contribution の本質は「陣取り合戦」 だから



Founder 企業の産業コンソーシアム創設の企図 は何なのだろうか？

テクニカル面の期待 (本筋)

- 企業間の実装フラグメントを解消し、デファクトを確立 させ易くする
- 各社に分散しているエース開発者を結集させ 開発リソースを強化拡充 する
- クロスレビューの実施によってコード品質を向上 させる
- 長期的に維持可能なソフトウエア資産を構築する

ビジネス面の期待 (不純？、でも本質)

- 特定企業のブランドを脱色 して他の企業/団体に 共創への参加を促す
- 類似の 他の（先行する）活動に対する対抗策 として打ち出す
- 活動をリーディングすることで 業界内でのポジションを確立 する
- 公知化によって独占禁止法などによる訴訟リスクを回避 したい

コンソーシアム活動の Founder 企業の期待値は、ケースバイケースで異なるだろう

活動に 参加する企業 にも、それぞれの期待値と振る舞い がある

テクニカル面の期待

- 共創開発 への参加（先端技術の取得）
- 自社技術の デファクト化 を狙う
- 業界動向など、技術情報 の取得
- 攻めの参画
  - 自社エースエンジニアの投入
  - 積極的な開発貢献
- 受け身
  - 開発者会議等への参加
  - 情報収集モード に徹して発言せず

ビジネス面の期待

- Founder 企業に 参加を要請された
- ビジネス獲得の前提 として参加
- 企業間の 会話チャンネル 確保
- 共創活動参加を 社会にアピール する
- 積極活用
  - マーケティング機会 として有効活用
  - プレスリリース等で活動をアピール
- 消極的な対応
  - 最低ランク会員 で参加実績作り

企業による「共創開発」の場となる「産業コンソーシアム」

産業コンソーシアムは 個人ベースの OSS 開発コミュニティではない

OSS 開発コミュニティ = 個人ベース活動

- コミュニティの形成は自然発生的 で、明確な創始者が居ない場合もある
- 活動参加は任意（通常、契約は不要）
- 原則年会費などの 費用負担は無い
- 積極的な貢献が期待される
- 規模が大きくなると、コードリリースサイクル管理、マージルール等の運用ルール の整備が必要になる

産業コンソーシアム = 企業ベース活動

- 創設メンバー (Founding Member)（多くの場合企業）が明確である
- 明文化された メンバーシップ契約 に基づいて活動に参加する
- 参加費や年会費 などの義務がある
- 開発参加には CLA (Contributor License Agreement) が必要な場合も
- 開発リソースコミット (=FTE) の要請

産業コンソーシアムは、明確な目的や意思を持った企業による共創活動での舞台である

産業コンソーシアムの運営スタイル にはバリエーションがある

Code First 型の共創活動

- Founder のリーダーシップが強い場合
  - 自社で開発済みのコード を持ち込む
  - コードの品質や要件適合性 はケースバイケース
- コンソーシアム内で コードを新規開発
  - 参加メンバーが コードの共創開発 を行う（調停 は大変困難）
  - メンバー会費で 新規に開発委託
  - メンバー企業が推奨する 既存ソリューションを採用 する

Spec First 型の共創活動

- 参加メンバーの 要求仕様（要件定義）の収集 を行う
- 各メンバー企業が それぞれ持ち帰って実開発を実施 し、成果を持ち寄る
- コンソーシアム自体はで行うのは
  - 重複開発摘出とロードマップ共有
  - 外部への広報活動（更なるフラグメントの回避）
  - コンソーシアム成果の一般化活動

「共創開発」では必ずしも Code First 型の方が良いとは言えないケースもある

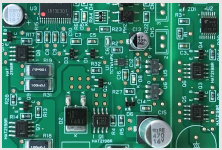
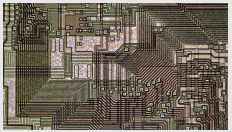
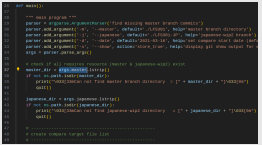
自動車業界のオープン化戦略

『ソフトウエアファースト』 by 及川卓也

ソフトウエアファースト（=SF） の命名者

- 及川氏 = IT 技術の急激な進化の現場にいた当事者
  - Dec → Microsoft → Google
  - Increments (Qiita 運営母体)
  - 2017 から個人としてコンサルタント活動をスタート
  - Denso とも技術顧問契約を締結
- 及川氏がこの本で問題提起しているのは
  - 日本企業は IT 技術を軽視（= 社外に丸投げ）してきた
  - 各種 IT サービスのクラウド化への対応遅れ が顕著な例
  - その結果、日本の企業は GAFA などの先端 IT 企業から遅れをとってしまっているようにみえる
  - 先端 IT 技術はデジタルトランスフォーメーションの担い手 なので早急なキャッチアップが必要である

何故今、自動車で「SW First」がメガトレンドに

	2010 年以前	最近まで	今後
主な **差異化要因**	PCB基盤上の回路	SoC	SW
差異化技術の **帰属**	最終製品開発メーカー	半導体ベンダー	SW プラットフォーマ
主な **技術要件**	堅牢性	機能の集積	移植性（再利用性）
イメージ			
解決すべき **課題**	機器間ネットワーク 経年劣化対策	処理性能 消費電力（発熱対策）	SW 更新 サイバーセキュリティ

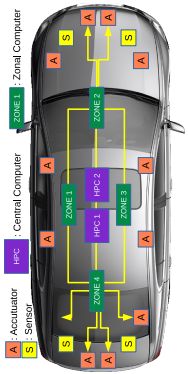
背景には、製品に差異化をもたらす主要因が SW に移行してきた事実がある

SDV (=Software Defined Vehicle) の概念

BEV（Battery EV) は SDV（SW 定義型車両） 実現を目指す

- これまでは、機能別の制御ユニット（ECU） が多数配置
- 機能追加（変更）は 物理 ECU の増設（置換） が必要
- 更に、ECU 連携のための大量の配線（ワイヤーハーネス）

- BEV では制御が単純なので、制御 ECU 数が減らせる
- HW 機能の統合化、階層化 によって ECU 機能が統合できる
- 機能追加（変更）は SW の変更によって実現
- 通信機能を使った OTA (=Over The Air) で SW を変更する



頻繁な SW 更新で端末の魅力が維持する「スマートフォンの流儀」を自動車にも適用

SOAFEE by Arm ･･･ 共創活動の成果 とアピールしている

arm

Armの最新テクノロジーが、自動車業界におけるソフトウェア定義型の未来を変革

September 16, 2021

2021年9月15日、英国ケンブリッジ発 ― 英Arm（本社：英国ケンブリッジ、日本法人：神奈川県横浜市、以下Arm）は本日、自動車サプライチェーンにおけるリーダー各社との協業を通じて、最新のソフトウェア・アーキテクチャとオープンソースのリファレンス実装である「Scalable Open Architecture for Embedded Edge（SOAFEE）」、ならびに自動車業界におけるソフトウェア定義型（software-defined）の未来を推進する2種類の新たなリファレンスハードウェア・プラットフォームを提供することを発表しました。

成功へのロードマップ

SOAFEEは、自動車メーカーやシステム・インテグレーター、半導体、ソフトウェア、クラウドのテクノロジーリーダー各社が協力し、ソフトウェア定義型自動車向けのオープンスタンダードに準拠した最新のアーキテクチャを定義するという一連の取り組みの成果です。さらに、こうしたリーダー企業の分科会（SIG）によって定義されたアーキテクチャの実装であるSOAFEEのリファレンス実装が今後、広範なプロトタイプ作成やワークロード検証、早期開発のための無償のオープンソース・ソフトウェアとして提供されます。互換性を最大限に高めて、機能安全に優れた、より迅速な設計手段を可能にするため、Armは業界をリードする商用ソリューション・プロバイダー各社と協力しています。

https://www.arm.com/ja/company/news/2021/09/16-9-2021-new-arm-technologies-to-transform-the-software-defined-future-f

「ソフトウエア ファースト」に関する「よくある誤解」

「ソフトウエア ファースト」で何が変わるのか？

- 組み込み機器開発では 常にプログラムを実行するターゲット が先に決まっている
  - プログラマーの仕事は ハードウエア性能を最大限に引き出す制御手順 を考えること
  - 時には ハードウエアの制約（不具合）回避 もソフトウエアの役割とされた
  - 開発予算配分ではソフトウエア開発費が削減対象コストになることも多い

- 「ソフトウエア ファースト」になっても、以下のようなことは起こりません
  - どんなハードウエアでも動く（= 移植可能な）ソフトウエア が開発できる
  - 機器の優位性はソフトウエアによって生み出されるので ハードは何でもよくなる
  - 特殊なコーディングが必要な HW アクセラレータ活用ではなく、全て CPU 処理 としたい
  - ハードウエア開発よりも ソフトウエア開発費用に多くの費用を使うのは当然

「ソフトウエアがハードウエアに優先する」とは誰も言っていない点に注意

まとめ

- SoC (=System On Chip) は、これまでディスクリート半導体の組合せで構成されていた システム回路を１チップ上に集約 したもの。専用のソフトウエアが無ければ動かせない ので、半導体メーカーはソフトウエアの開発にも力を入れている。
- SoC 制御用のソフトウエアに OSS (=Open Source Software) を活用 されている。Linux OS は組み込み機器の制御用にも幅広く利用されるようになった。
- 半導体メーカーが OSS 開発コミュニティの活動に参加するケース も増えている。OSS にはコードの再利用を促進する 「配布ライセンス」 や「多拠点同時開発を行うためのコード管理システム」 が整備されている。
- ライバル関係のメーカーが OSS 領域で協力する「共創開発」は新しいトレンド。自動車業界では共創開発を推進する「産業コンソーシアム活動」 が盛んである。

ユニット③映像系技術が実現する未来社会

JEITA

以下、3回の講義を聞いて、次ページ以降に記載の共通課題について、提出してください。

【提出〆切】1月17日（水）14:00

第10回 12/13	「半導体産業の動向」 「DRAM」	マイクロンメモリジャパン㈱永島 靖 氏 マイクロンメモリジャパン㈱田桑 哲也 氏
第11回 12/20	「イメージセンサー」 「マイコン・システムLSI」	ソニーセミコンダクタソリューションズ㈱ 山田 英史 氏 ルネサスエレクトロニクス㈱ 宗像 尚郎 氏
第12回 1/10	「メモリ」 「パワーデバイス」	キオクシア㈱ 内川 浩典 氏 ローム㈱ 愛宕 崇之 氏

JSIA JEITA 一般社団法人 電子情報技術産業協会 半導体部会

「半導体が創り出す未来社会①～⑥」 共通課題

Common challenges for “Semiconductors to create a better future ①～⑥”

あなたが現在取り組んでいる（あるいは今後取り組んでみたい）研究テーマ に関して、以下の設問に具体的に、A4用紙 1 枚以内で回答してください。
（回答には、①②③すべての要素を含んでください）

Please describe your thoughts as specific as possible on your current (future) research for each question below in one sheet (A4 size paper).

① どのような世界を目指すのか
What kind of future do you want to create?

② その実現に向けて、どのような課題を認識し、どうアプローチしようとしているか
To make it happen, what is your challenge and how do you address it?

③ その中で、半導体がどう関与し、どう活かせるか
How can semiconductors get involved and contribute to building your future?

JSIA　JEITA 一般社団法人 電子情報技術産業協会 半導体部会

「半導体が創り出す未来社会①～⑥」 共通課題　採点ガイドライン

あなたが、現在取り組んでいる（あるいは今後取り組んでみたい）研究テーマ に関して、
以下の設問に具体的に、A4用紙 1 枚以内で回答してください。
（回答には、①②③すべての要素を含んでください）

① どのような世界を目指すのか、**（20点）**
社会や技術の動向を把握し、自らの考えを加えて具体的に書かれている**（20点）**、
社会や技術の動向を把握し、具体的に書かれている**（15点）**、書かれている**（10点）**、書かれていない**（0点）**

② その実現に向けて、どのような課題を認識し、どうアプローチしようとしているか、**（30点）**
社会や技術の動向をふまえて課題を認識し、自らの考えにもとづいたアプローチをとろうとしている**（30点）**、
社会や技術の動向をふまえて課題を認識し、アプローチしようとしている**（25点）**、
課題とアプローチが書かれている**（20点）**、書かれていない**（0点）**

③ その中で、半導体がどう関与し、どう活かせるか、**（50点）**
半導体技術の動向を理解し、自らの考えにもとづいて活用しようとしている**（50点）**、
半導体技術の動向を理解し、活用しようとしている**（40点）**、
活用方法が書かれている**（30点）**、書かれていない**（0点）**

合計点
S（90点以上）、A（80～89点）、B（70～79点）、C（60～69点）、D（59点以下）

JSIA JEITA 一般社団法人 電子情報技術産業協会 半導体部会

Grading Guidelines for “Semiconductors to create a better future ①～⑥”

Please describe your thoughts as specific as possible on your current (future) research for each question below in one sheet (A4 size paper).

① What kind of future do you want to create? **（20pt）**

・Written specifically with an understanding of social and technological trends, with unique ideas added. **（20pt）**
・Written specifically with an understanding of social and technological trends.**（15pt）**
・Written. **（10pt）**
・Not written.**（0pt）**

② To make it happen, what is your challenge and how do you address it? **（30pt）**

・Recognized the challenges based on social and technological trends and tries to take an approach based on own ideas. **（30pt）**
・ Recognized the challenges based on social and technological trends and tries to take an approach. **（25pt）**
・Described the challenges and the approaches. **（20pt）**、
・Not Described.**（0pt）**

JSIA JEITA 一般社団法人 電子情報技術産業協会 半導体部会

Grading Guidelines for “Semiconductors to create a better future ①～⑥”

Please describe your thoughts as specific as possible on your current (future) research for each question below in one sheet (A4 size paper).

③ How can semiconductors get involved and contribute to building your future? **（50pt）**

・Understands trends in semiconductor technology and tries to apply them based on own ideas. **（50pt）**
・Understands trends in semiconductor technology and tries to apply them. **（40pt）**、
・Written. **（30pt）**
・Not Written. **（0pt）**

Grading by total points

S (90pt or more), A (80～89pt), B (70～79pt), C (60～69pt), D (59pt or less)