*GPL*とソースコード

中島達夫

早稲田大学

tatsuo@dcl.info.waseda.ac.jp

内容

- GPLとLGPL
- **プログラム：技術的な側面**
- **プログラム：ライセンスの側面**
- **組込みシステムとLinux**
- **日本エンベデッド・リナックス・コンソーシアム（Japan Embedded Linux Consortium）**

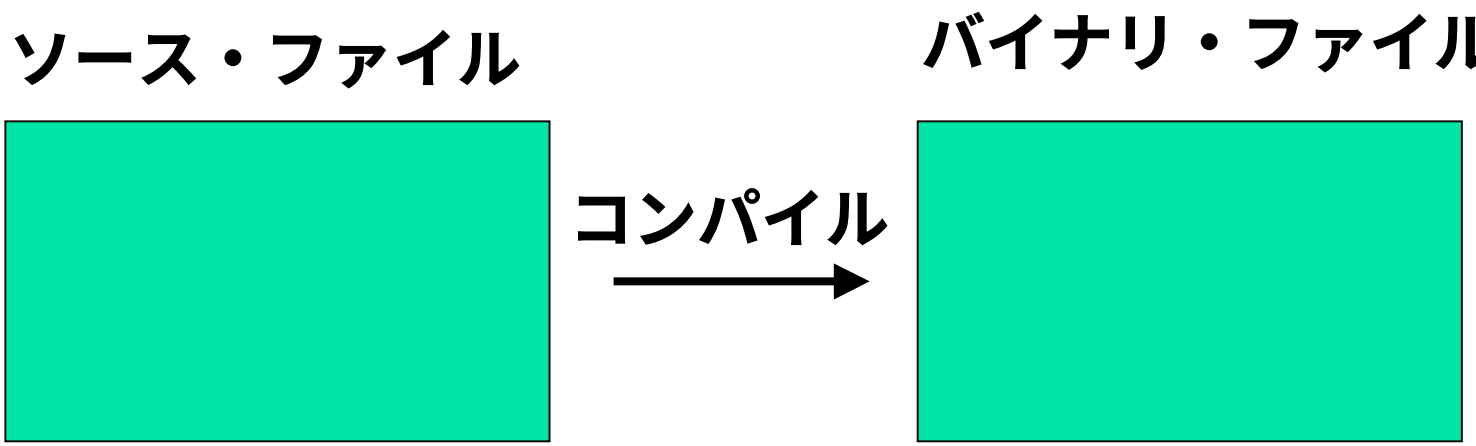
社会問題とコンピューター・サイエンス

- コンピューター・サイエンスにおける様々な社会問題を考慮に入れる必要がある。
  - ライセンス
    - プログラムは通常、自由に使用できるものではない。
  - 信頼
    - 誰を信頼すればよいか。
  - プライバシー
    - 自分に関する情報をどのように保護するか。

GPLとLGPL

- GPL
  - **一般公衆利用許諾契約書**（General Public License）
  - GPL**は通常、アプリケーション、カーネルおよびサーバーに使用される。**
- LGPL
  - **劣等一般公衆利用許諾契約書**（Lesser General Public License）
  - LGPL**は通常、ライブラリに使用される。**

プログラム（1）



プログラム（2）

プログラム

```
#include “header.h”

main(int argc, char **argv)
 {
   libcall();
 }
```

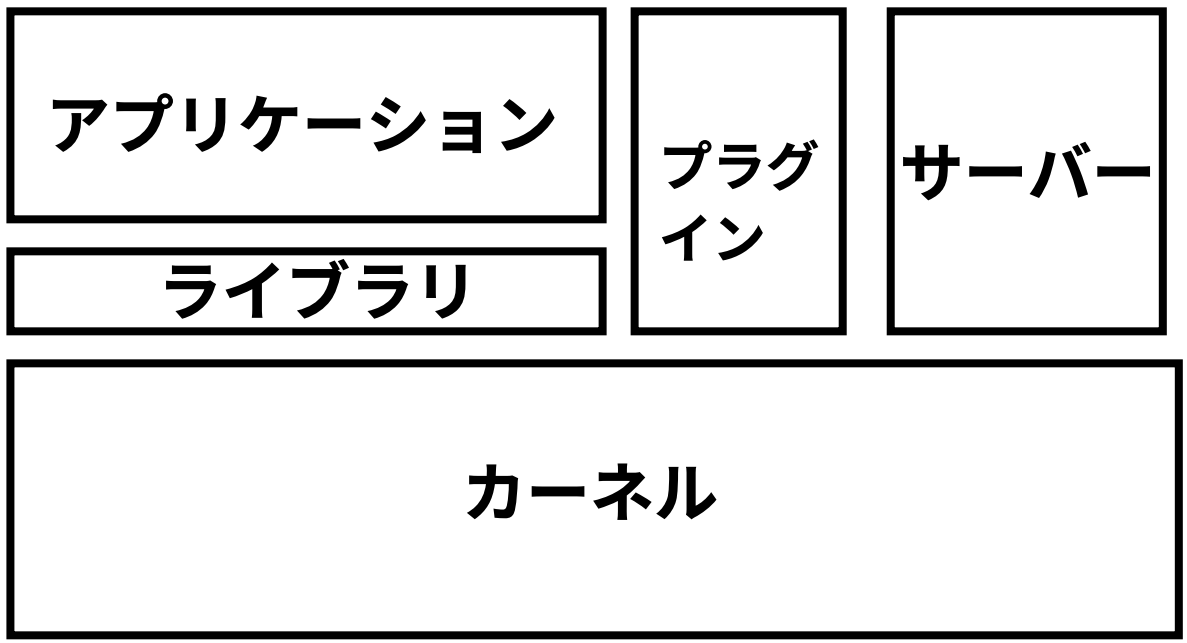
ライブラリ

```
void libcall()
{
  …
}
```

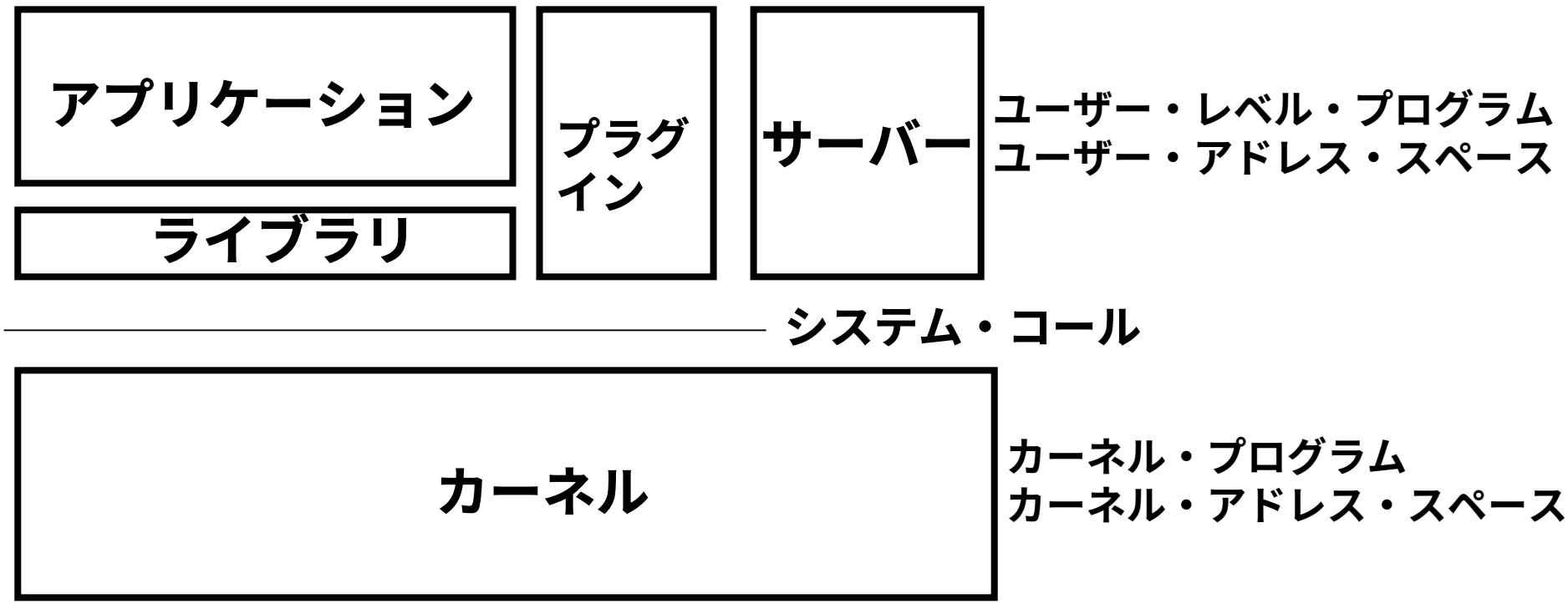
ヘッダー

```
#define MAX 10
#define max(a,b) ((a>b)?a:b)
```

Linuxの構造



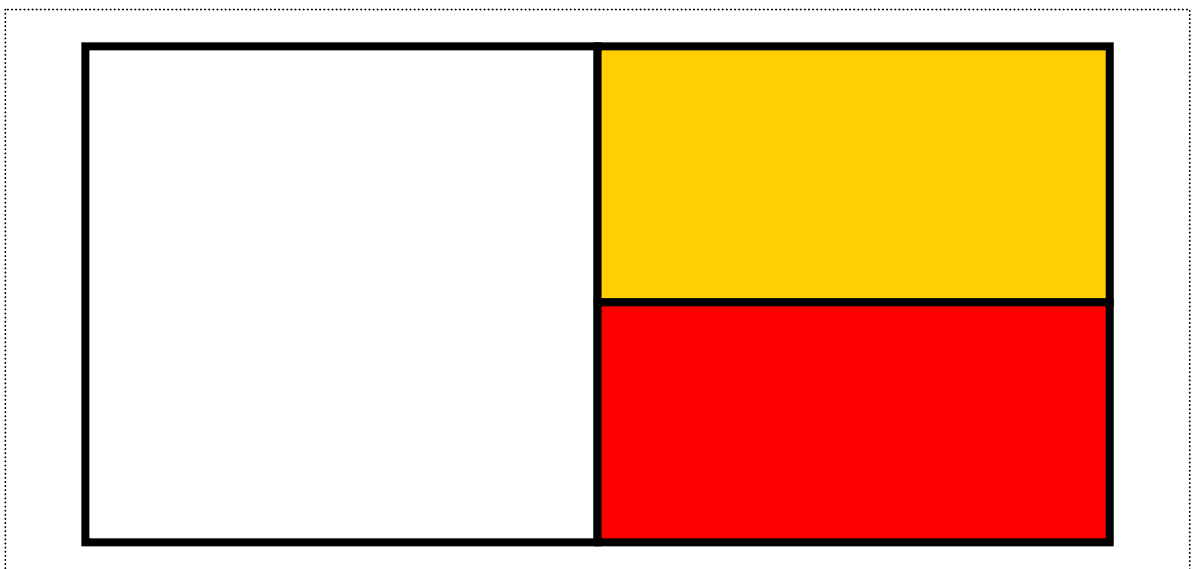
システム・コール



★ システム・コールは、ユーザー・レベル・プログラムとカーネルの間のインターフェース。
★ システム・コールは、トラップ命令をして使用アドレス・スペースの切換え。

ライブラリ

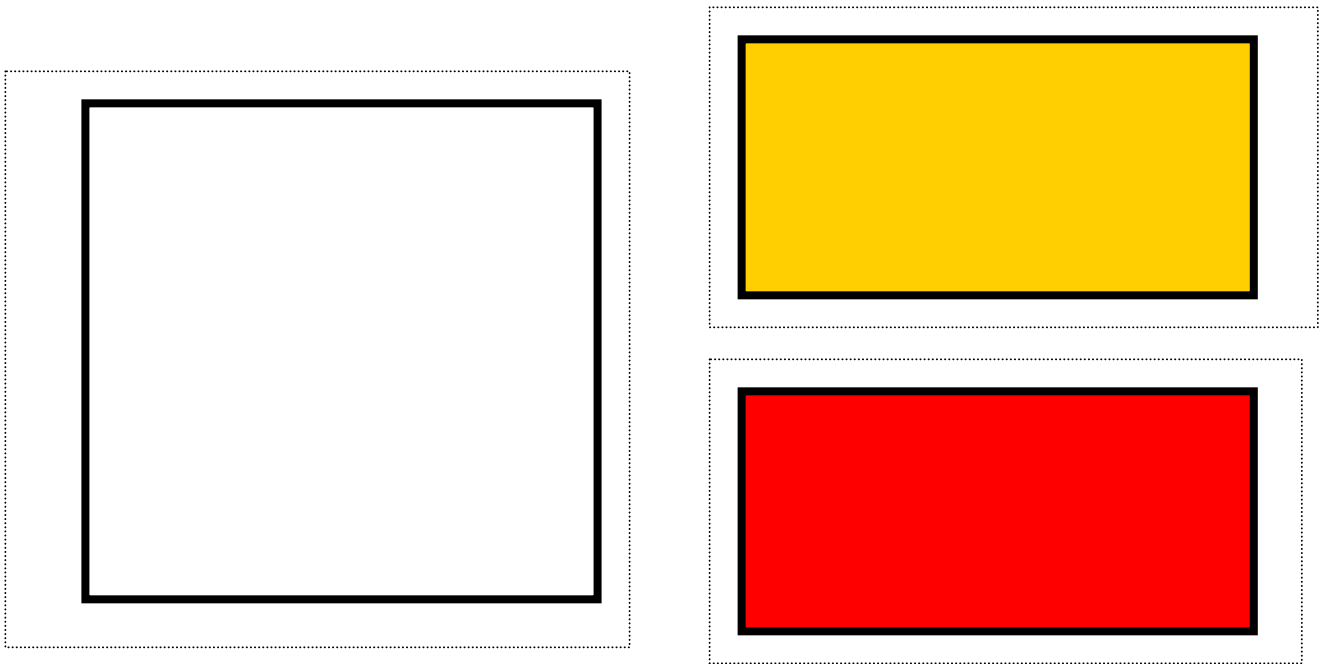
- **アプリケーション・プログラムは、実行可能なコードの作成に、多くのライブラリ・プログラムを使用する。**
  - glibc, GTK++, Qt,
  - cc program.c –o program –llib (liblib.a)



1つのアドレス・スペースにおいて

プラグイン

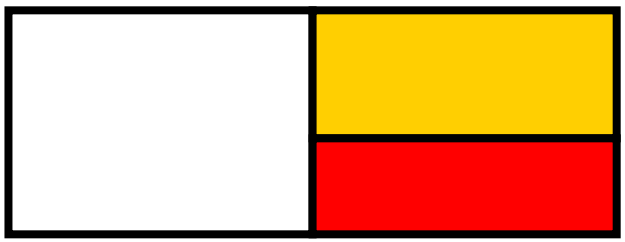
- プラグインは、異なるアドレス・スペースにおいて実行される。



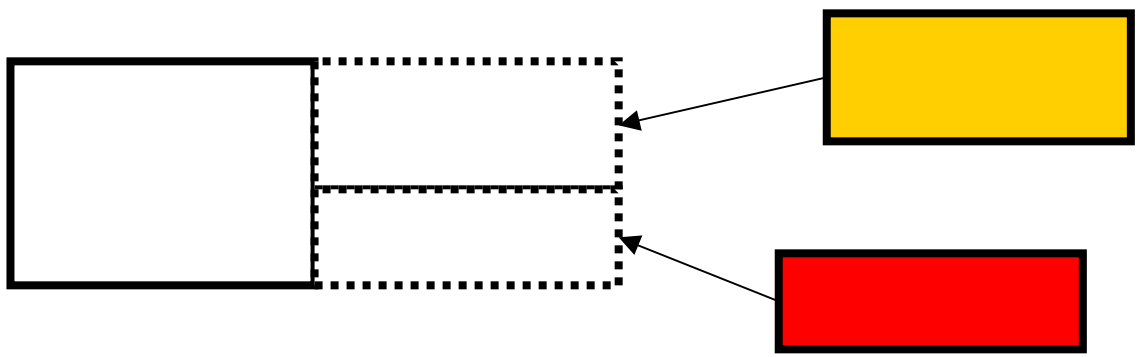
多くのアプリケーションとサーバー

リンク

- 静的リンク
  - ライブラリは、コンパイル時にリンクされる

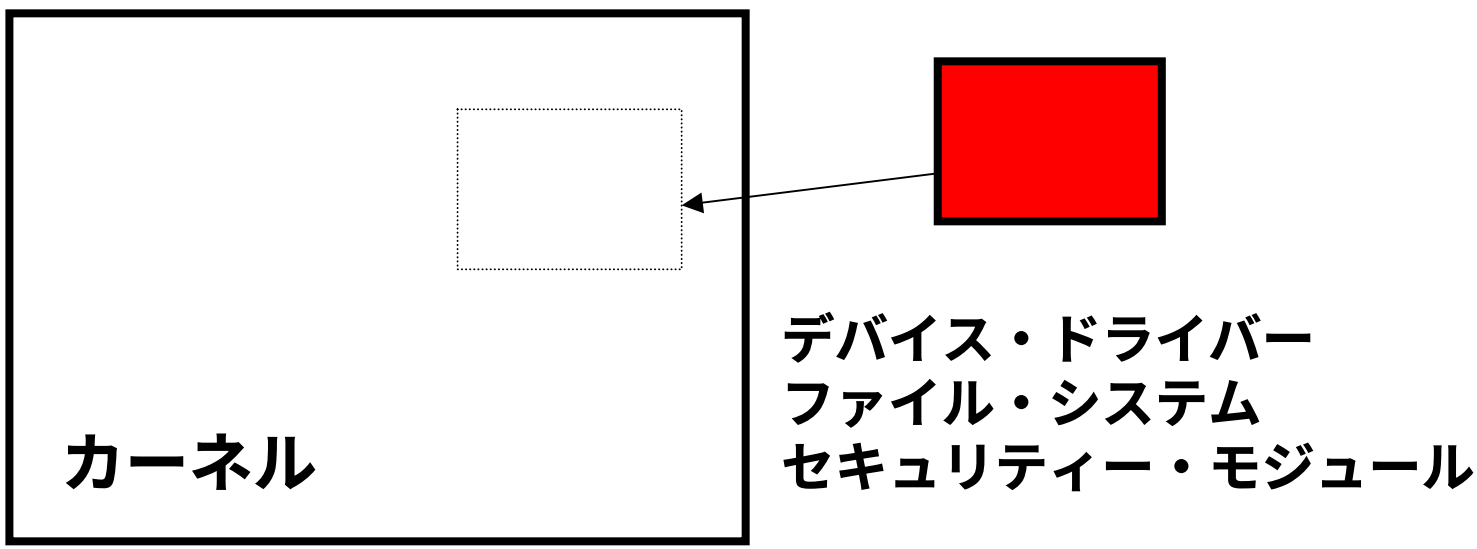


- 動的リンク
  - ライブラリは、実行時にリンクされる。



ローダブル・カーネル・モジュール

- ローダブル・カーネル・モジュールは、実行時にカーネルにリンクされる。
  - 戦略は動的リンキングに類似している。



ライセンスの問題点

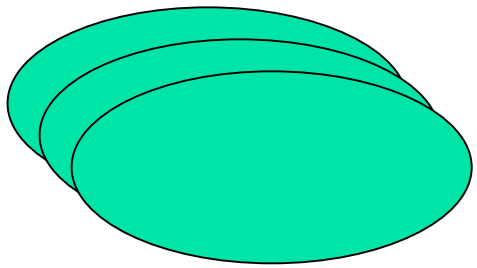
- GPLとソースコード
  - バイナリコードを配布する場合には、他者からの要請に応じてそのソースコードを提供する必要がある。

  → 他者にバイナリコードを提供しない場合には、自作の改変ソースコードを開示する必要はない。

  → 組込みシステムにはバイナリコードが含まれているので、そのソースコードは公開の準備をしておくべきである。

標準インターフェース

- **標準インターフェースは、GPLの効果を停止させる境界線である。**



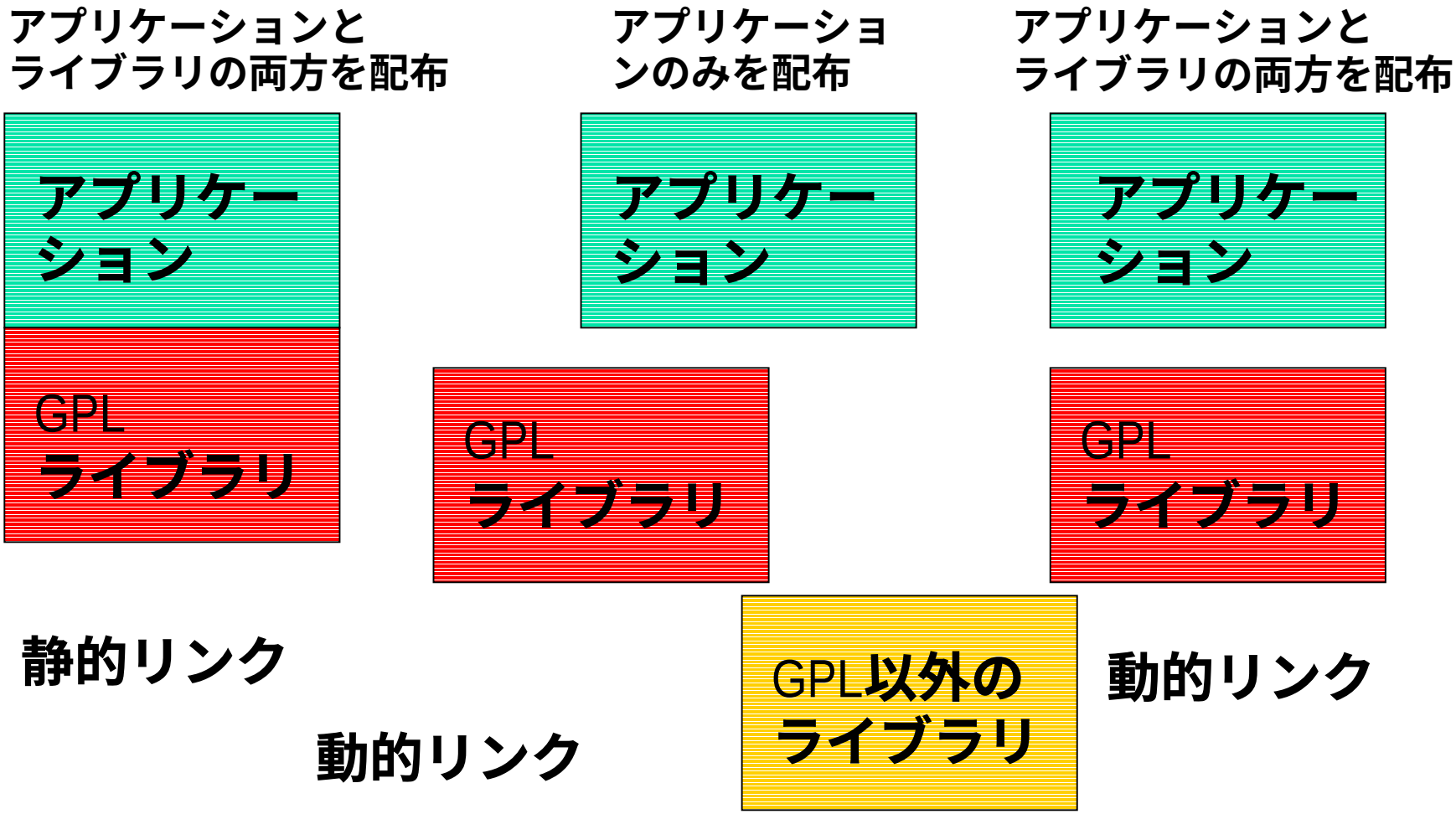
GPLソフトウェア

GPL以外の
ソフトウェア

アプリケーション

- Linuxカーネル・インターフェースが標準インターフェースであれば、すべてのアプリケーションはプロプライエタリとすることができる。
  - 現行のLinuxインターフェースはPOSIXに基づいているが、標準化されてはいない。

派生物とリンク（1）

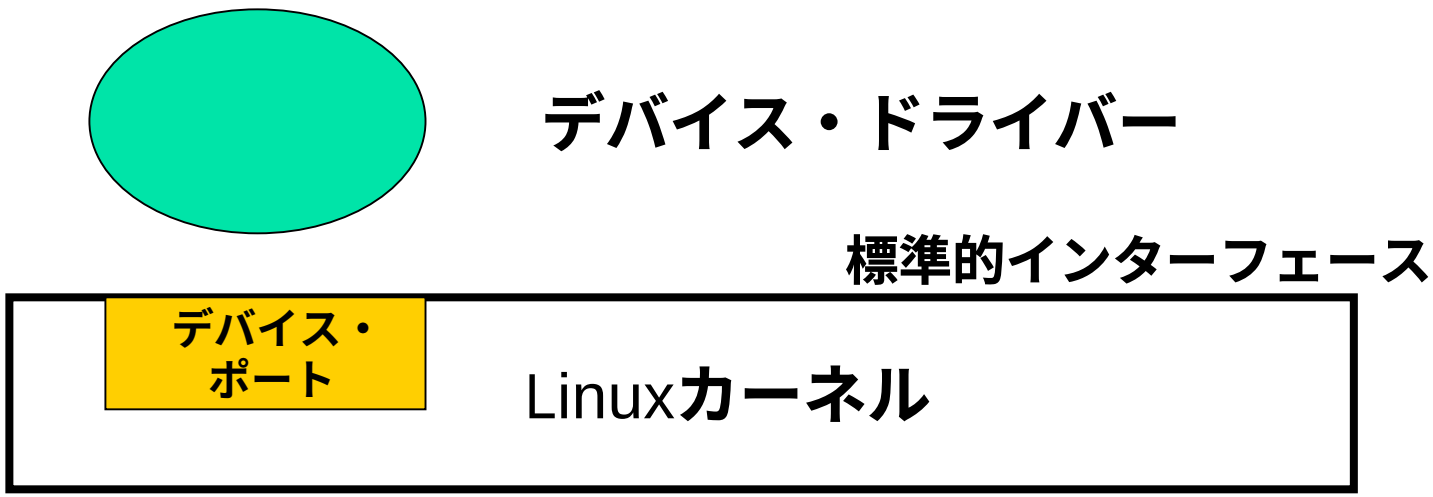

アプリケーションとライブラリの両方を配布
アプリケーション
GPL ライブラリ
静的リンク
アプリケーションのみを配布
アプリケーション
GPL ライブラリ
動的リンク
アプリケーションとライブラリの両方を配布
アプリケーション
GPL ライブラリ
GPL以外のライブラリ
動的リンク

派生物とリンク（2）

- ライブラリが標準インターフェースを提供するのであれば、自作のソースコードはプロプライエタリとすることができる。
  - しかしながら、自作のバイナリコードがGPLライブラリと静的にリンクしているのであれば、自作コードはGPLライセンスによってライセンスされるべきである。
  - ところが、動的リンクを使用しているのであれば、ライブラリをGPL以外のライブラリに置き換えることができるので、自作コードはプロプライエタリとすることができる。
  - しかし、1つのシステム（組込みシステムなど）で自作コードをGPLライブラリとともに配布するのであれば、自作コードはGPLライセンスによってライセンス許諾できる。
  - ライブラリがLGPLライセンスによってライセンス許諾されるのであれば、自作コードはプロプライエタリとすることができる。

ユーザー・レベル・デバイス・ドライバー

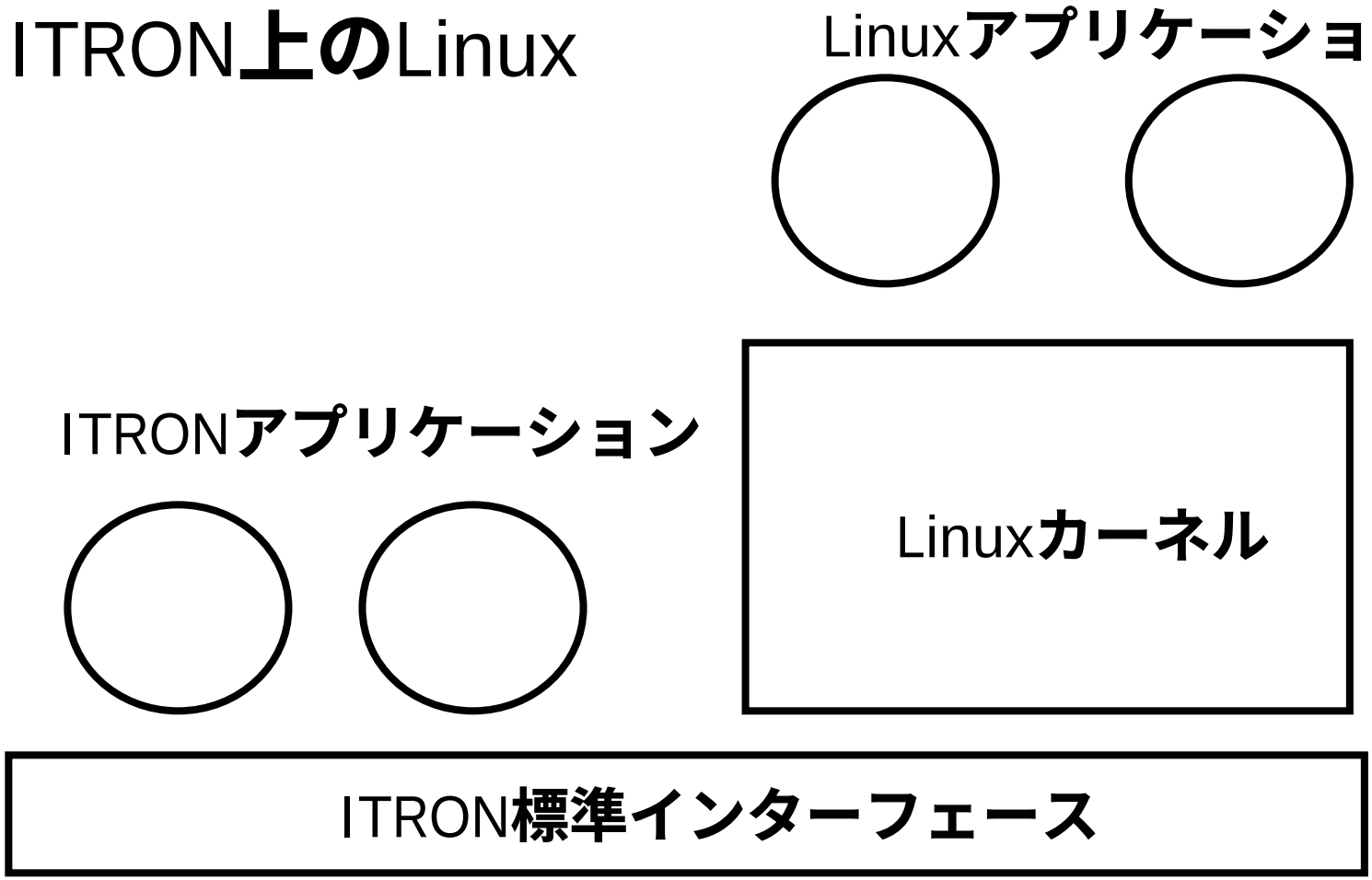
- ユーザー・レベル・デバイス・ドライバーはユーザー・レベルで実行され、GPLライセンスによってライセンス許諾が可能である。



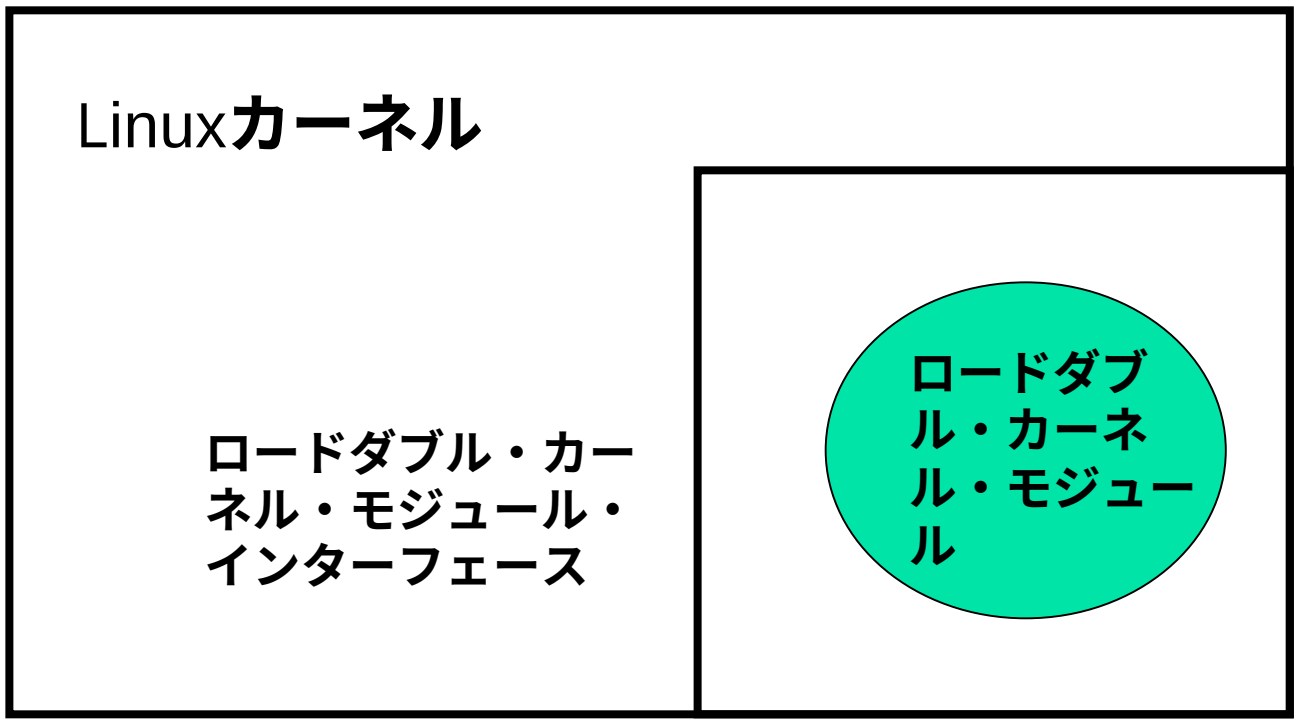
どのように割り込みに対処するか。

ハイブリッド・アーキテクチャー

- ITRON上のLinux

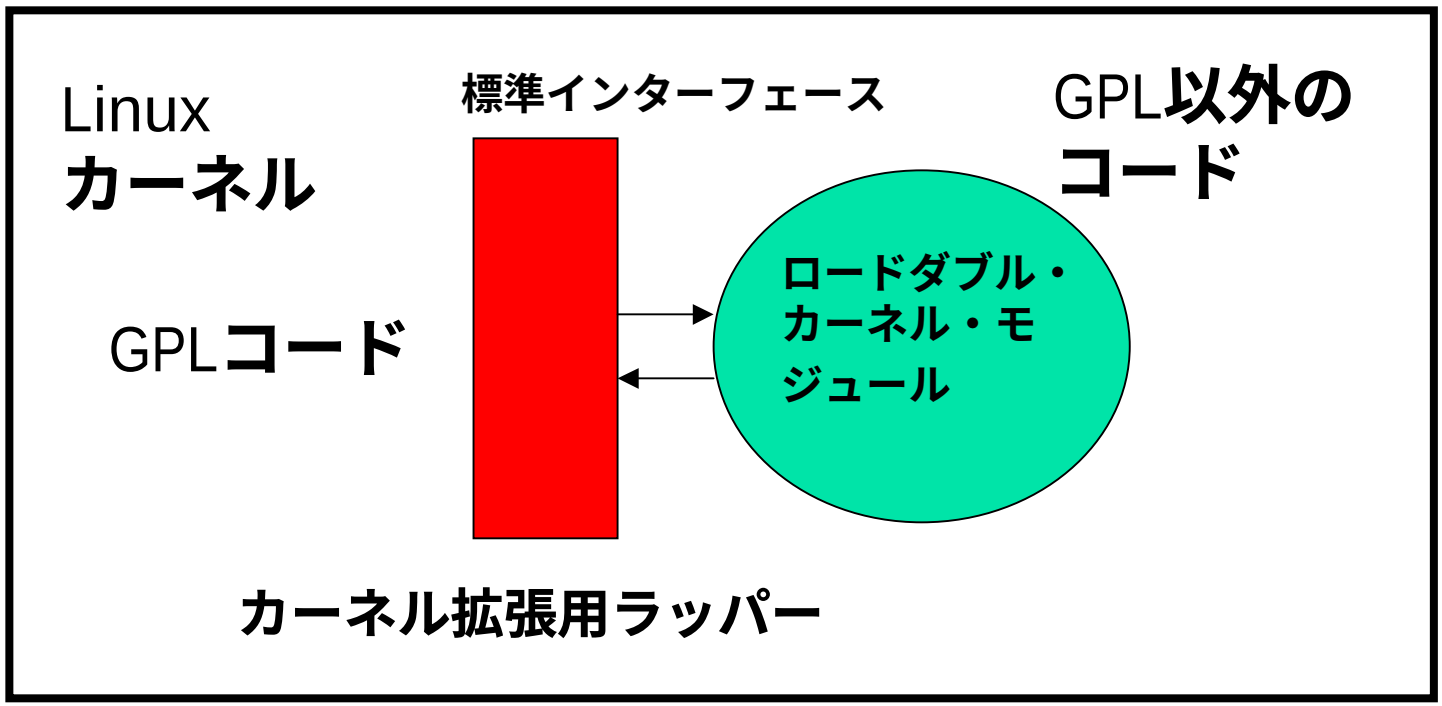


ロードダブル・カーネル・モジュール



ロードダブル・カーネル・モジュール・インターフェースは標準インターフェースではない。

カーネルの拡張



この種のカーネルの拡張は可能であろうか。

論点

- 標準インターフェースの定義が明確でない。
  - デファクトスタンダードが実際の標準なのであろうか。
- 国によって、GPLに関する解釈が異なる場合がある。
- 採用されたオペレーティング・システムによって、論点が変わってくる可能性がある。
  - カーネルの拡張メカニズムに依存
- 採用されたプログラミング言語によって、論点が変わってくる可能性がある。
  - 言語の機能に依存

組込みシステムの現状

- 組込みシステムは、ますます複雑になってきている。
  - 携帯電話、カーナビ、デジタルテレビ放送
- 何が問題か。
  - 複雑さはパーソナル・コンピューターと同レベルである。
  - 従来のソフト・インフラは、複雑なソフトウェア開発に適していない。
  - 日々、新しい要件に直面している。

将来の組込みシステム

- より一層進んだ組込み
  - 単純かつ分散型
  - 厳しいリソースの制約
  - 厳格な信頼性、リアルタイム
- 情報家電
  - 専門的な機能
  - 複雑、構成可能
  - セキュリティー、仮想信頼性
  - 現在のパーソナル・コンピューターに取って代わるであろう。

情報家電

- 自分のPDAからキオスクの端末装置のディスプレイを使用したい。
- キーボードを使用して、自分のPDA上でデータを編集したい。

→　将来のコンピューターは、種々のデバイスを構成するために使用される。

→　情報家電には、プロトコル・スタック、サービス・ディスカバリー、Webサービス、動的プログラム・ローディングなどが必要である。

→　メモリー保護、ネットワーク・サポート、POSIXサポート、種々のミドルウェア。

→　Linuxは、情報家電にとって最適のプラットフォームである。

日本エンベデッド・リナックス・コンソーシアム（Japan Embedded Linux Consortium）（1）

- **組込みLinuxに関する産学協同をサポートするためのコンソーシアム**
  - **会員数はおよそ100**
    - **松下、ソニー、NEC、富士通、東芝、ノキア・ジャパンなど**
  - **組込みLinuxの普及・発展**
  - **組込みLinuxに関連する技術の標準化**
  - **種々のコンソーシアムとの間の情報交換**
    - Embedded Linux Consortium、CE Linux Forum**など**
  - http://www.emblix.org/

日本エンベデッド・リナックス・コンソーシアム（Japan Embedded Linux Consortium）（2）

- オープンソースの問題点
- ライセンスの問題点
- 技術的な問題点
- 教育的な問題点

オープンソースの問題点

- 誰がソースコードを維持するのか。
  - 予算、ライセンス、持続可能な保守など
- 誰がプログラムを書くのか。
  - 予算、ライセンス、教育など
- どのようにソースコードを改変するのか。
  - コミュニティーまたは保守パッチ。
- どのようにプログラムを使用するのか。
  - 教育的な問題点（本、チュートリアル）
- どのようにソースコードの正確性を保証するのか。
  - 試験

ライセンスの問題点

- 組込みシステムでは、ライセンスの問題点がより一層複雑である。
  - ライブラリは、プロプライエタリなアプリケーション・コードとともに配布される。
  - ダイナミック・ローダブル・モジュールは、製品とともに出荷されるべきである。
  - ユーザーにソフトウェアのインストールを要請することは困難である。

結論

- プレゼンテーションでは、GPLに関連する技術的な問題点を提示した。
  - GPLは技術的な問題点を定義してはいない。
    - リンク戦略、プログラム構造など。
  - 個別の状況に合わせてGPLを解釈する必要がある。
  - 現時点では、個人個人がそれぞれ異なる解釈をしている可能性がある。