

**「組込み」って  
何なのか  
考えてみた**

**坂井弘亮  
(KOZOSプロジェクト)**



# 略歴

1997年頃 プログラミングに傾倒

X Window System プログラミング  
オブジェクト指向プログラミング  
OSに興味を持つ

1999年 就職

組込みシステムに興味を持つ

2007年 自作組込みOS「KOZOS」の開発を  
趣味で開始

2009年 OSC出展

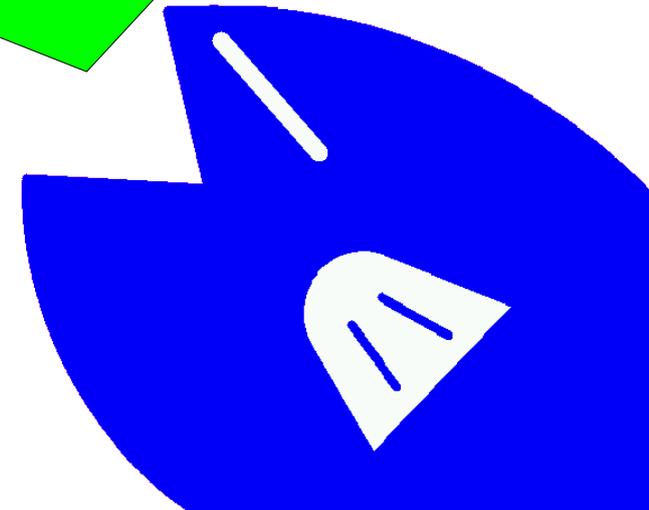
KOZOSを実ボードに移植

# まずはちょっとアンケート

- A. 組込みはまったく知らない分野だが、今日は興味はあって参加した
- B. 組込み初心者で、今日は勉強したくて参加した
- C. 普段から組込みにたずさわっている

# 1つ目の疑問

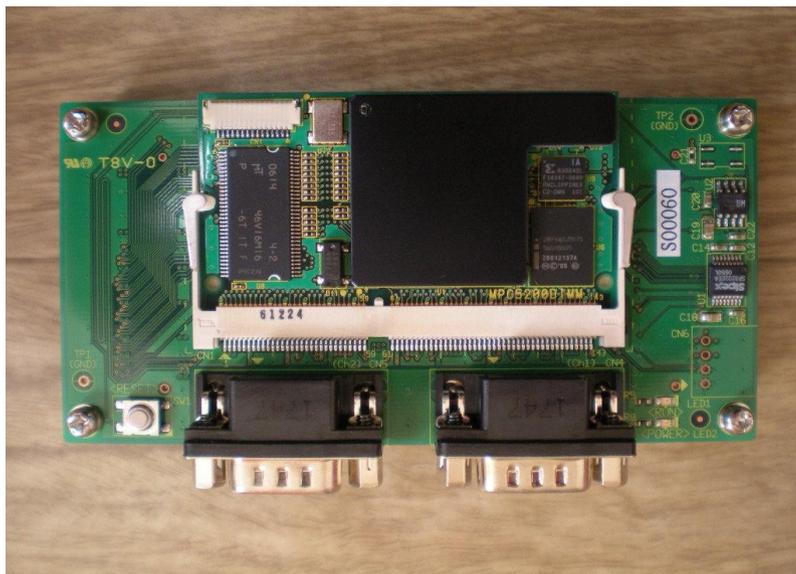
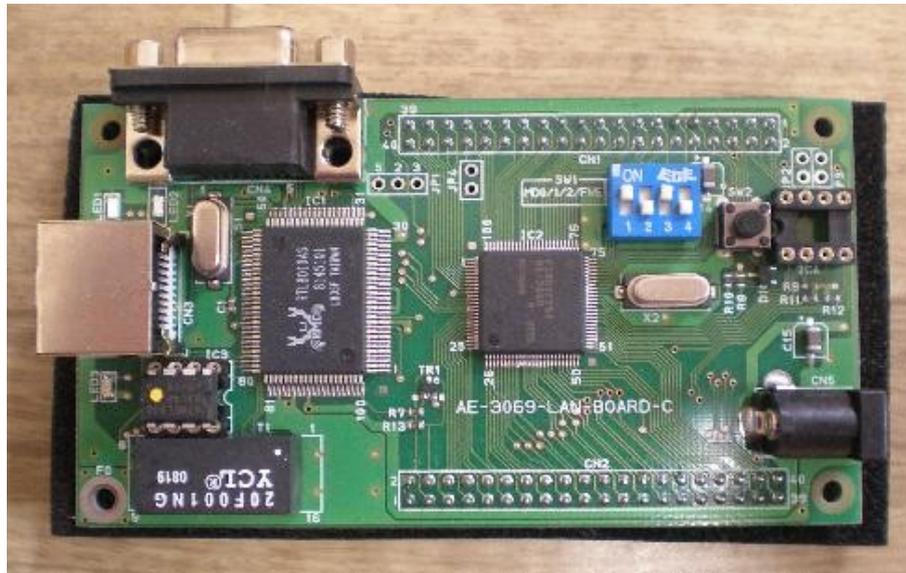
そもそも  
「組込み」って  
何？



# こんなのか



# こんなのか



# 汎用システム

- ・PC, ワークステーション

# 組込みシステム

- ・携帯電話、カーナビ、  
プリンタ、炊飯器、  
自動車、電子レンジ...

# 組込みシステムとは

特定の機能を実現するために、  
家電製品や機械などに組み込ま  
れるコンピュータシステムのこと

(まんまwikipediaですが、  
だいたいどこでもこんな説明です)

**まあざっくりいいうならば**

**PC以外は全部組込み！  
でもいいような気がします  
(乱暴すぎるか)**

# 組込み機器の特徴

CPU...4bit～32bit

メモリ...数KB～数百MB

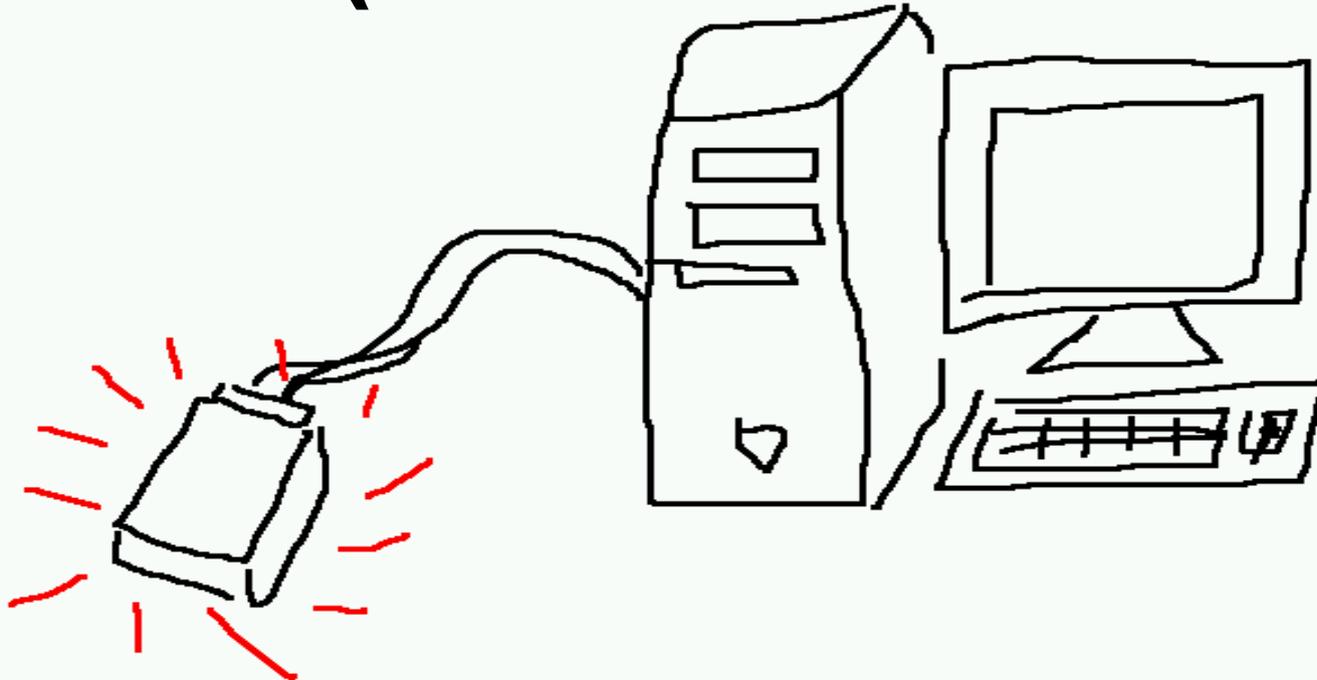
OS...無し、独自OS、

各種RTOS、Linux、...

とにかく幅が広い！

# 組込み開発の特徴

- ・ハードウェアには逆らえない  
(理想のコードを追求できないことも)
- ・クロス開発(デバッグとかたいへん)



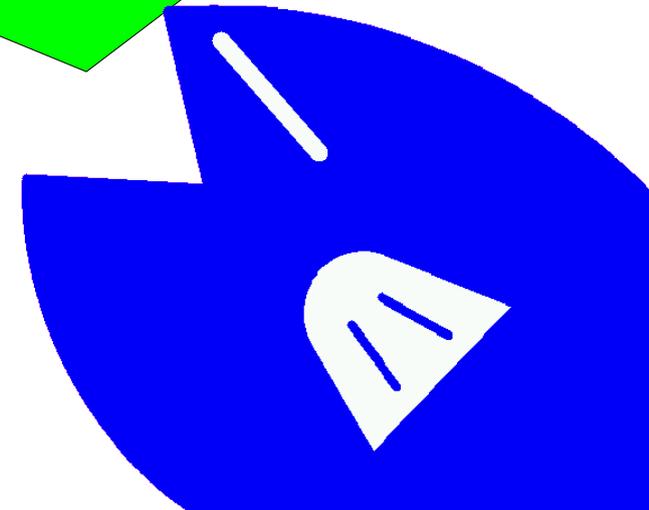
# 組込み界の特徴

- **業界主導**
- **クローズドな開発**
- **考え方もクローズドだったり**
- **理論より経験**
- **職人多し**
- **何でも知らないといけない**

**きちんと勉強する場が  
あまり無かったりしますよね**

# 2つ目の疑問

「OS」って  
何？



ちょっと考えてみてください

「シェル」は  
OSの  
一部でしょうか？

# シェルはOSの一部か？

- ・「OS」を、PCを動作させるためのシステム全体としてとらえるなら...  
シェルはOSの一部

- ・「OS」を、アプリケーションプログラムを動かすためのベースとしてとらえるなら...

シェルはアプリであってOSではない

# Windowsにマイコンピュータが 無かったら？

アプリのインストールすらできない！  
→何もできない！

インストーラを起動する手段が無い  
インストーラをUSBメモリに入れて  
挿しても中身を見れないし、  
インストーラを起動することができない  
でもマウスを動かせばカーソルは動く

# アプリにも2種類ある

- ・ 分類的にはアプリだが、コンピュータがひとつの実用システムとして動作するために必須のもの  
(例: シェル、マイコンピュータなど)  
→ システムアプリ
- ・ 本来、利用したいアプリ  
(例: ブラウザ、メール、エディタなど)  
→ ユーザアプリ

# ・狭義のOS

## ...カーネル

(アプリを動作させるためのシステム)

# ・広義のOS

## ...カーネル+システムアプリ

(「ユーザが利用する実用システム」としてのシステム)

このへんの言葉がごっちゃになって  
混乱するが多い  
(とくに「OS自作」とか言ったとき)

きちんと区別したいときは...

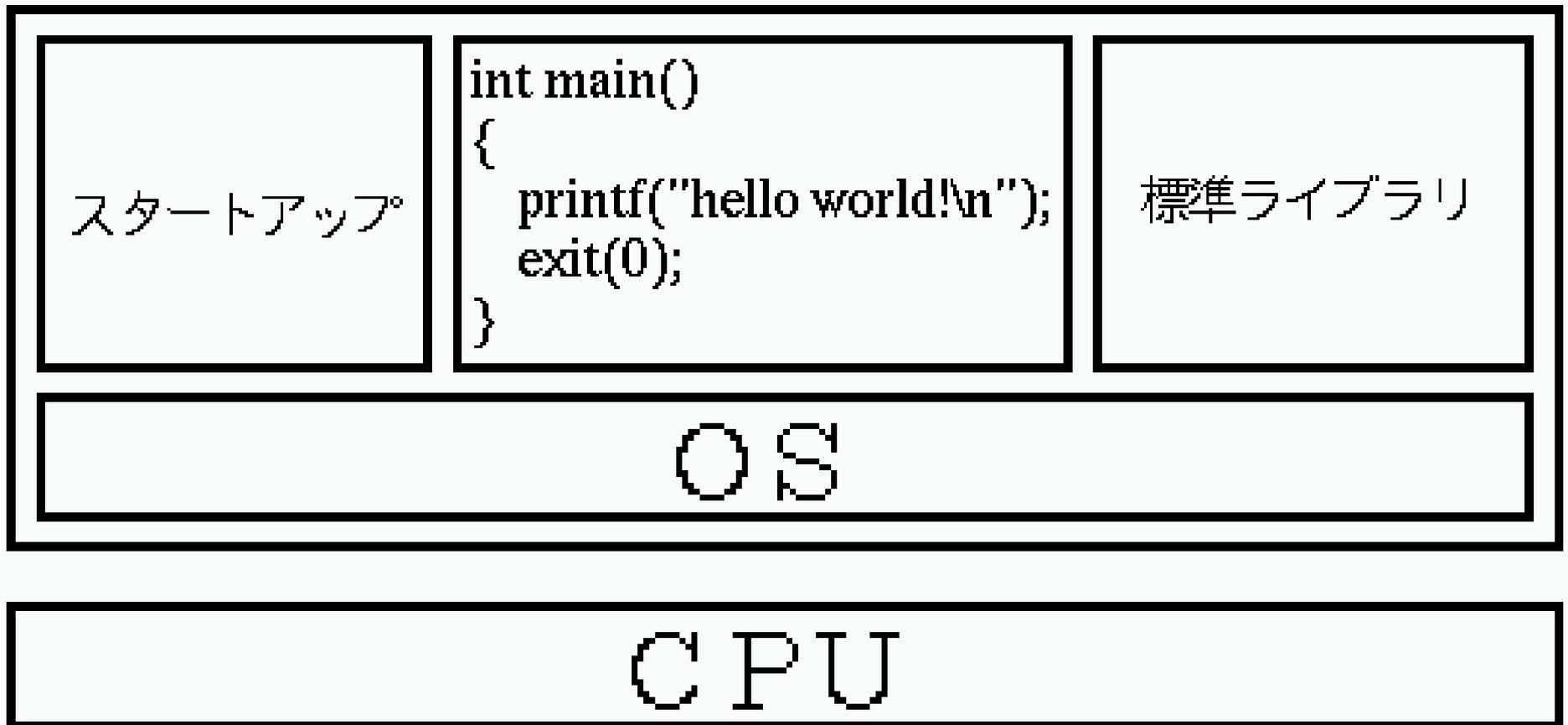
- ・カーネル

- ・システムアプリ(シェルとか)

- ・ユーザアプリ

と、言葉を分けよう！

# 言葉の違いに注意！ たとえば、「プログラム」



**C言語の入門書では...**

**「関数呼び出し時のスタックは、  
勝手に作成・操作されるので、  
プログラム側で気にする必要は無い」**

**→C言語ユーザとしての視点**

**CPUのマニュアルでは...**

**「関数呼び出し時のスタックは、  
プログラム側で作成・操作する  
必要がある」**

**→CPU利用者としての視点**

**こういうので混乱しないよう注意！**

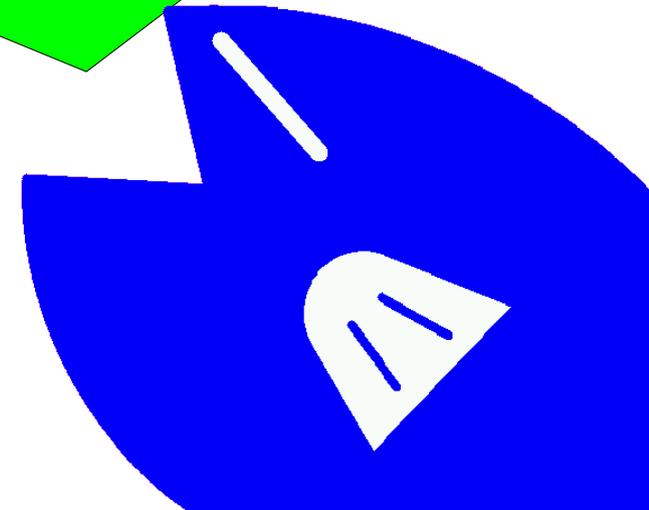
**OSの定義は？**

# 何をしていればOSと言えるのか？

- ・リソース管理という観点ならば  
...CPU管理、メモリ管理、I/O管理が  
できていれば「OS」
- ・アプリを動かすベースという観点ならば  
...アプリにサービスを提供するのが「OS」
- ・ソフトウェアの階層構造という観点ならば  
...その下に何も無い最下層が「OS」

# 3つ目の疑問

組込みでも  
「OS」は必要か？



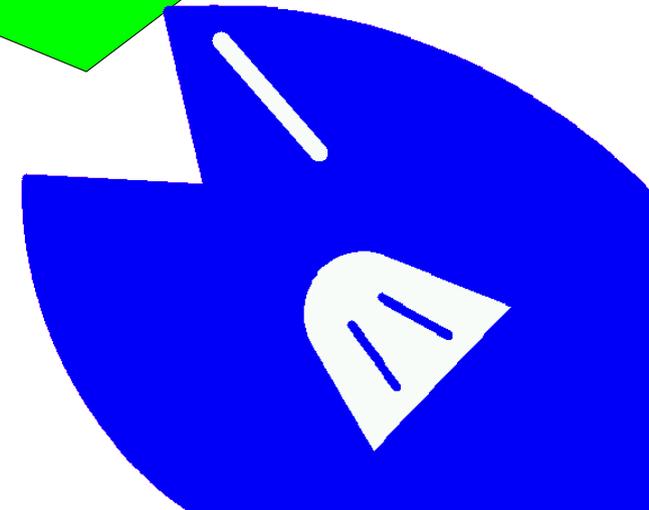
炊飯器にも  
OSは必要か？

**(実演)**

**OSが無いとは  
どういうことか？**

# 4つ目の疑問

「汎用OS」と  
「組込みOS」の  
違いは何か？



- **汎用機器**

ワークステーション、PC

(そのままでは役立たず、  
アプリを入れて使う)

- **組み込み機器**

携帯電話、カーナビ、炊飯器、  
電子レンジ、プリンタ、自動車...

(アプリは製品の一部であり  
ユーザは意識しない)

- **汎用OS**

**Windows、Mac、Linux、  
\* BSD、Solaris...**

- **組込みOS**

**μITRON、VxWorks、  
uClinux、OSEK...**

# 汎用OSは...

アプリケーションをインストールする  
ベース環境としての役割

→カーネルとしてのOSでなく、  
実用システムとしてのOSがほしい

→いろんなアプリをインストールして  
動かしたい

# 組込みOSは...

複数のI/Oを制御したいが、  
デバイスごとにマイコンを  
割り当ててはコスト高

- 複数のデバイスをひとつのCPUで  
制御したい
- マルチタスク、リアルタイム制御、  
タスクの優先度制御

**なので汎用OSは  
「隣のひとと同じものを  
使ったほうがなにかと便利」**

**組込みOSは  
「隣の人と同じものを使う必要は無い」**

**→汎用OSは淘汰の方向に、  
組込みOSは多様化の方向に  
進みます**

# 組込みでは、複数のI/Oを操作したい

...時分割で操作する

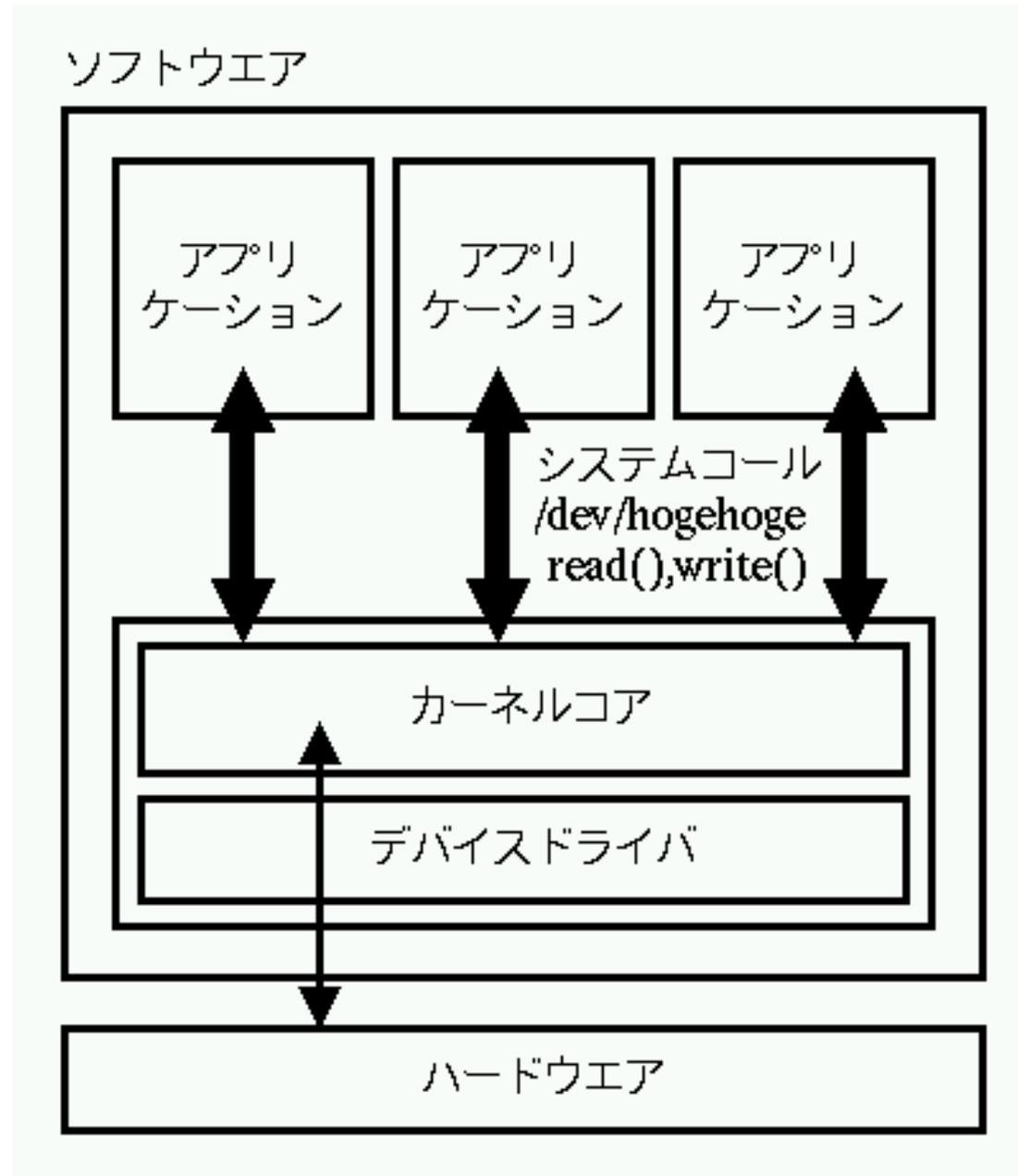
→タスクに優先度という考えが必要

(あくまで例ですが)例えば自動車なら、  
どのような優先順位にすべきか？

- ・ワイパー制御
- ・ハンドル制御
- ・エアバッグ制御

# 汎用OSと 組込みOSの 構造的な違い

# UNIXの(ていうかBSDの)構造



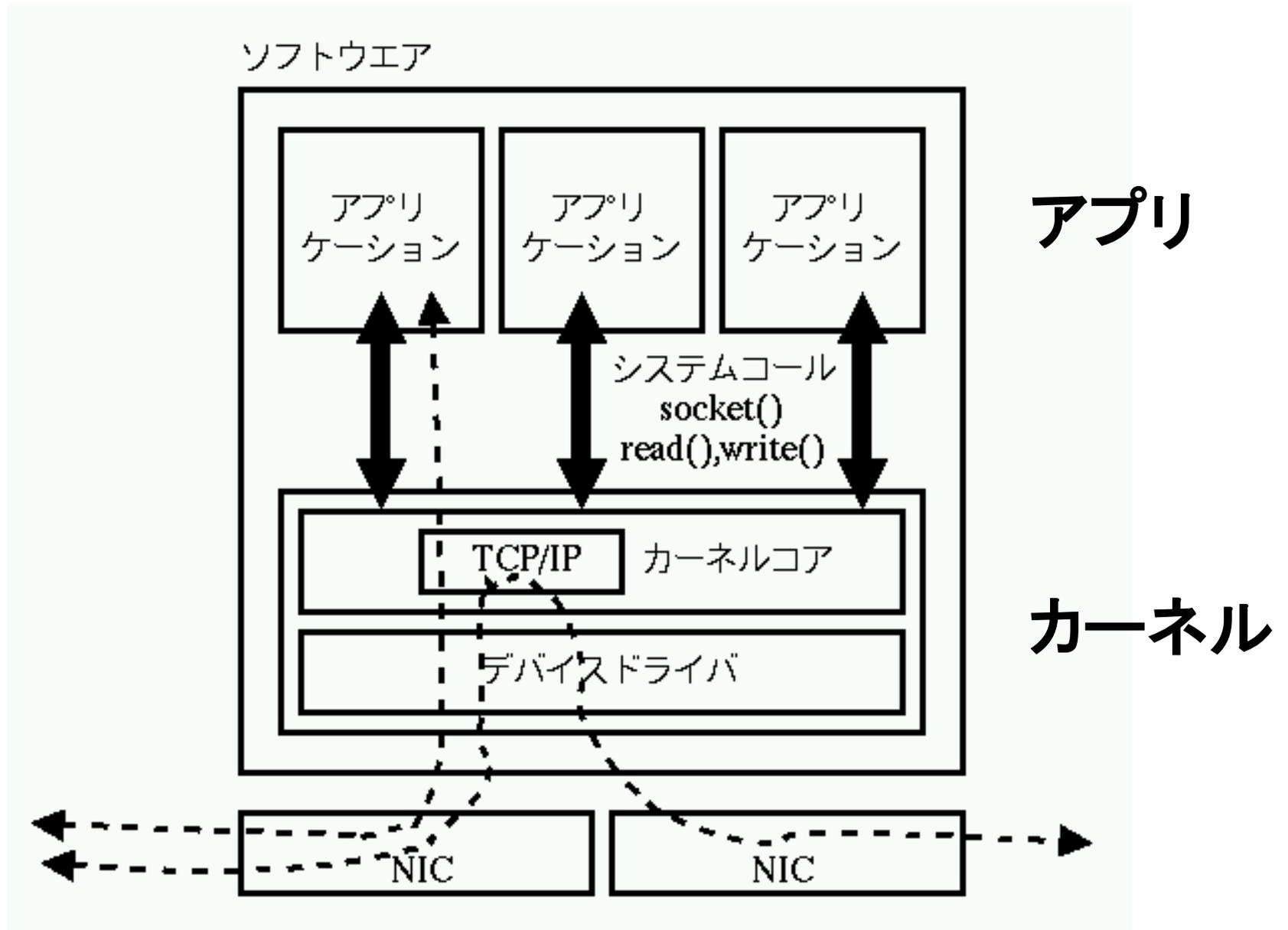
アプリ

カーネル

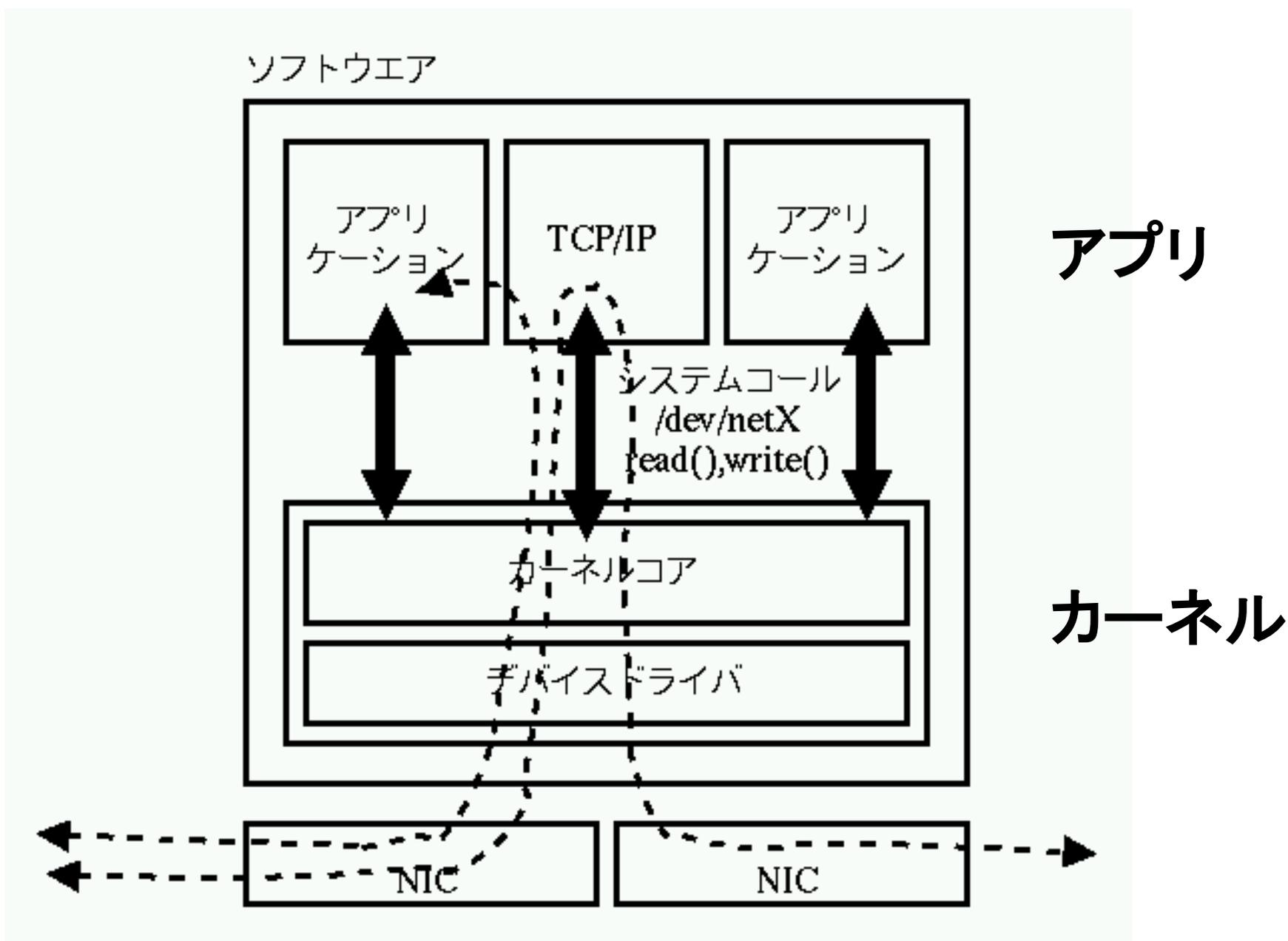
# UNIXは 「ファイルシステムOS」

I/Oを「デバイスファイル」  
として抽象化し(/dev/XXX)、  
ファイルのread()/write()で  
すべて操作できるという  
コンセプト

# TCP/IPも内蔵



# こんななんでもよかったんじゃあ



# なぜ、そうしなかったのか

ネットワーク性能(パケット転送性能)  
を優先したため

→BSDが生き残った大きな理由の  
ひとつになった

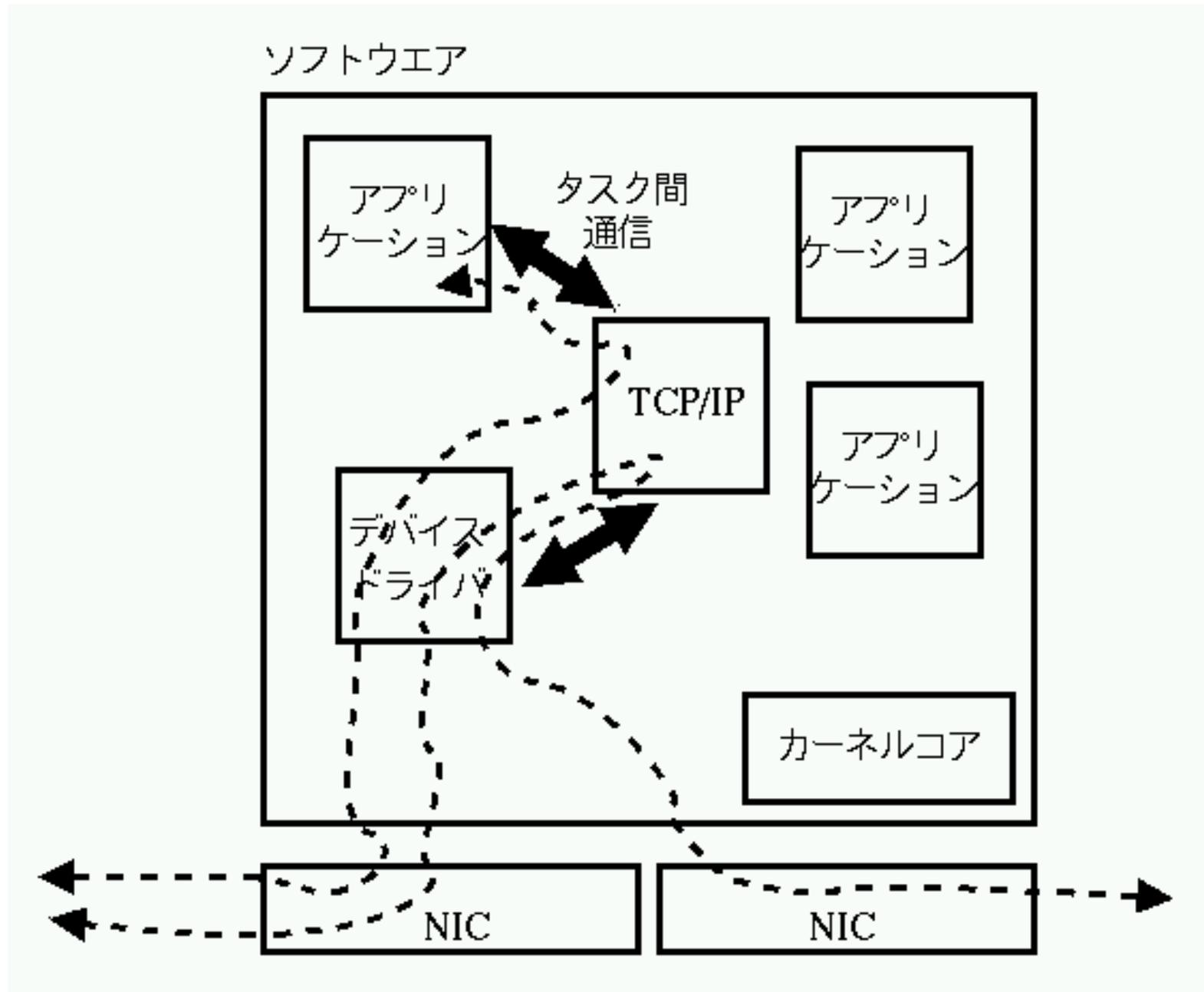
※もともとIPv4プロトコルは、

32ビットCPUで処理しやすい

ように設計されている

(int型のマスク計算で処理できる)

# 組み込みOSの構造(主にRTOS)



# 組込みOSでは

- ・カーネルはマルチタスクを実現するための最低限の機能にしぼる
- ・ネットワークやファイルシステム、デバイスドライバはアプリケーションタスクとして実装する

...という場合が多い  
(主にRTOS)

# なぜ、このようにするのか

組み込みシステムでは、ネットワークを削りたい場合もあれば、優先度を調整したい場合もある

- ・ルータなどのネットワーク機器なら  
...ネットワークの優先度は高  
(何か操作したらパケット転送性能がガタ落ち、では困る)
- ・DVDプレイヤーとかでweb設定のために  
ネットワークが必要なだけなら  
...ネットワークの優先度は低  
(ブラウザのリロード押しまくったら動画がコマ落ち、では困る)

# まとめ

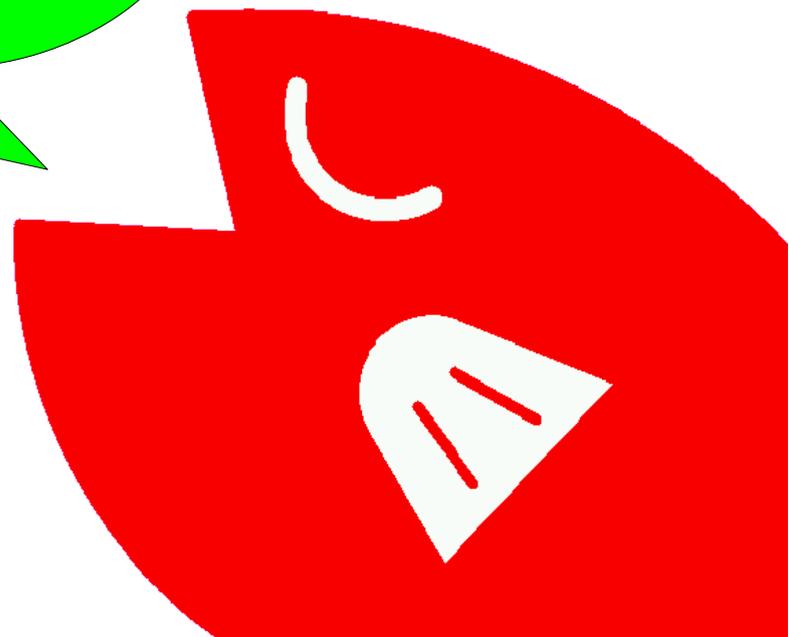
汎用と組み込みではOSの用途も目的も異なるので  
比較することはナンセンス  
両方を知ってたほうがいい

でもまあ現在は、境界もはっきりしなかったり  
(汎用OSを組み込みに利用したりとか、組み込み機器  
なんだけど中身は実は汎用PCだったりとか)

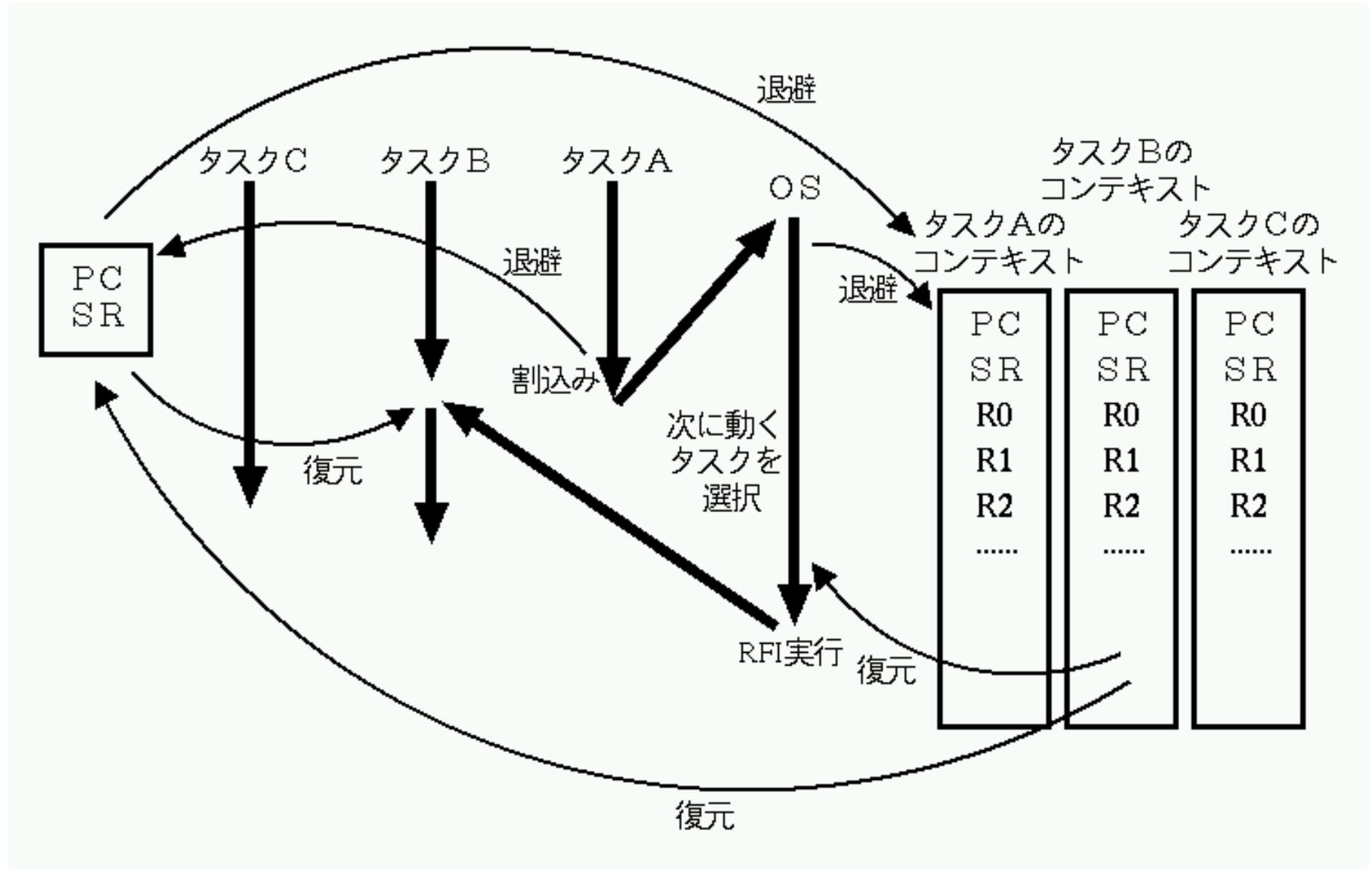
組み込みは混沌とした世界、多様な世界  
「どちらかに統一すべき」という考えに意味は無い  
「どちらも知ること」「取捨選択できること」が重要

※汎用は一神教、組み込みは多神教といえるかなあ

次回予告！

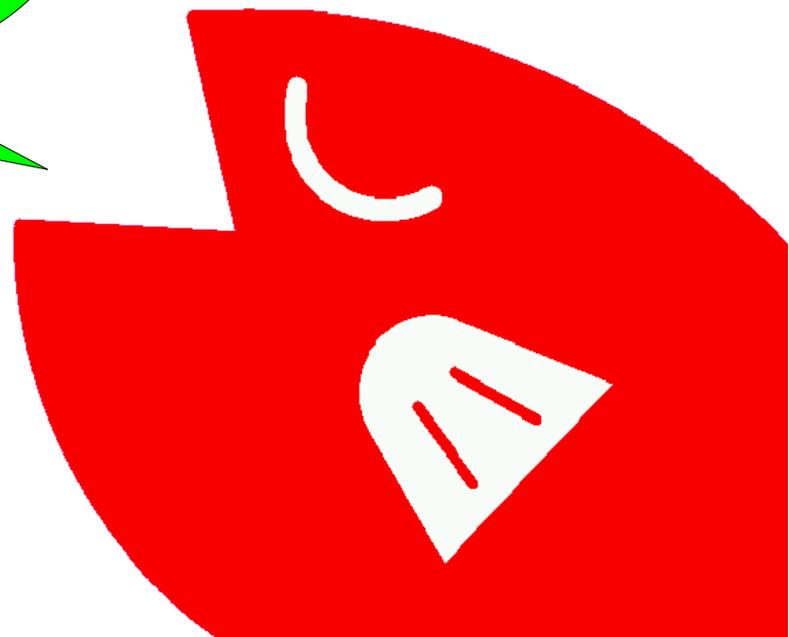


# KOZOSを利用して、マルチタスクの原理とか説明したい



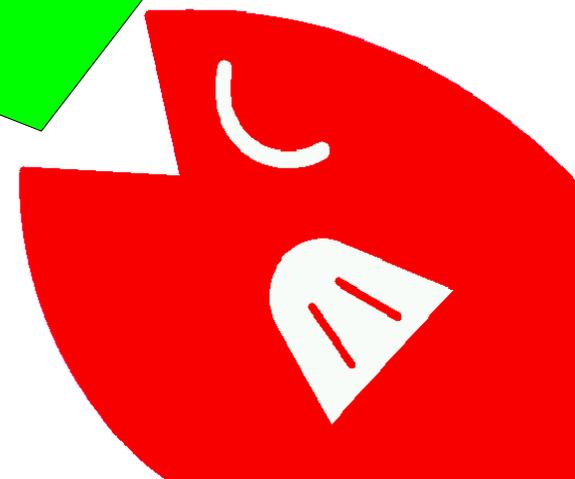


おしまい



# 汎用機器と組込み機器の 決定的な違い

ユーザがアプリを  
自由にインストール  
するかどうか



# 汎用機器に必要なこと

「バグのあるアプリ」

「悪意のあるアプリ」

「わがままなアプリ」が  
インストールされるかも  
しれない!

# 汎用機器に必要なこと

「他のアプリに悪さをせず」

「すべて平等に」

そのために...

- ・仮想メモリ、メモリ保護
- ・TSSによるスケジューリング
- ・ファイルシステム



# 組込み機器に必要なこと

ある処理の最中なのに、  
突然別の処理が必要に  
なるかもしれない！

# 組込み機器に必要なこと

「応答性を保証したい」

「タスクを優先度づけたい」

「リアルタイム制御を行いたい」

そのために...

- ・優先度によるスケジューリング
- ・リアルタイム性の提供

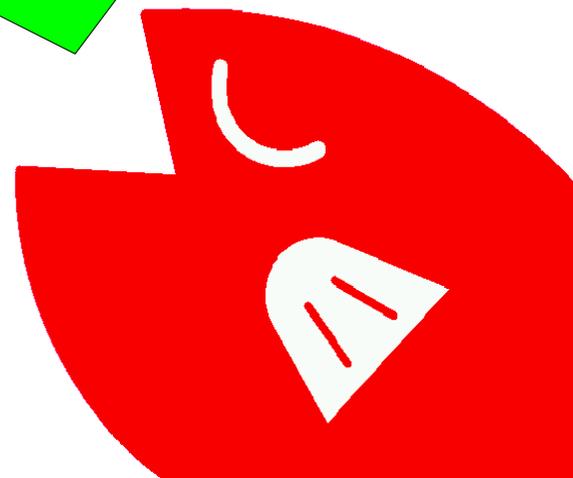


# ただし組込み機器では

アプリも含め、ひとつのシステムとして閉じられるので

- ・おかしなアプリがインストールされることはない!
- ・おかしなアプリがいたら、アプリ側を直すべき!
- ・システム全体に渡っての調整が可能(優先度など)

汎用OSは「性悪説」に  
組込OSは「性善説」に  
もとづくOSです



おしまい

