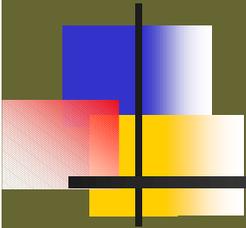


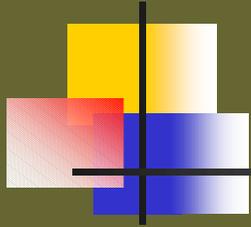
リアルタイム Linux 上の 「ユーザ空間デバイス・ドライバ」



KEKB 制御グループ

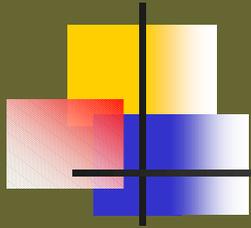
小田切 淳一、 山本 昇、 加藤 直彦

@技術交流会 平成 14 年 2 月 15 日



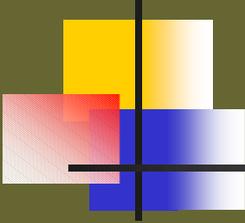
内 容

- 背景
- なぜ Linux はリアルタイムでないのか
- Linux のリアルタイム 拡張にはどのようなものがあるか
- ユーザ空間を使うデバイス・ドライバ
- まとめ



背景

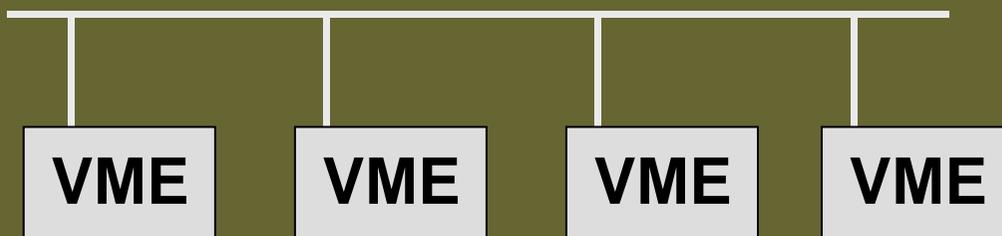
- EPICS iocCore
 - VME 計算機 (VxWorks リアルタイム OS)
- EPICS 3.14 から Linux をサポート
 - POSIX 1003.1c (pthread)
 - POSIX 1003.1b (real-time extension)
- Linux にはリアルタイム応答性がない



EPICS のプラットフォーム



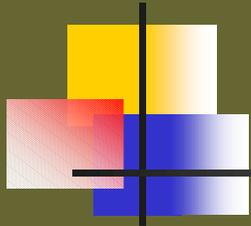
Work
Station



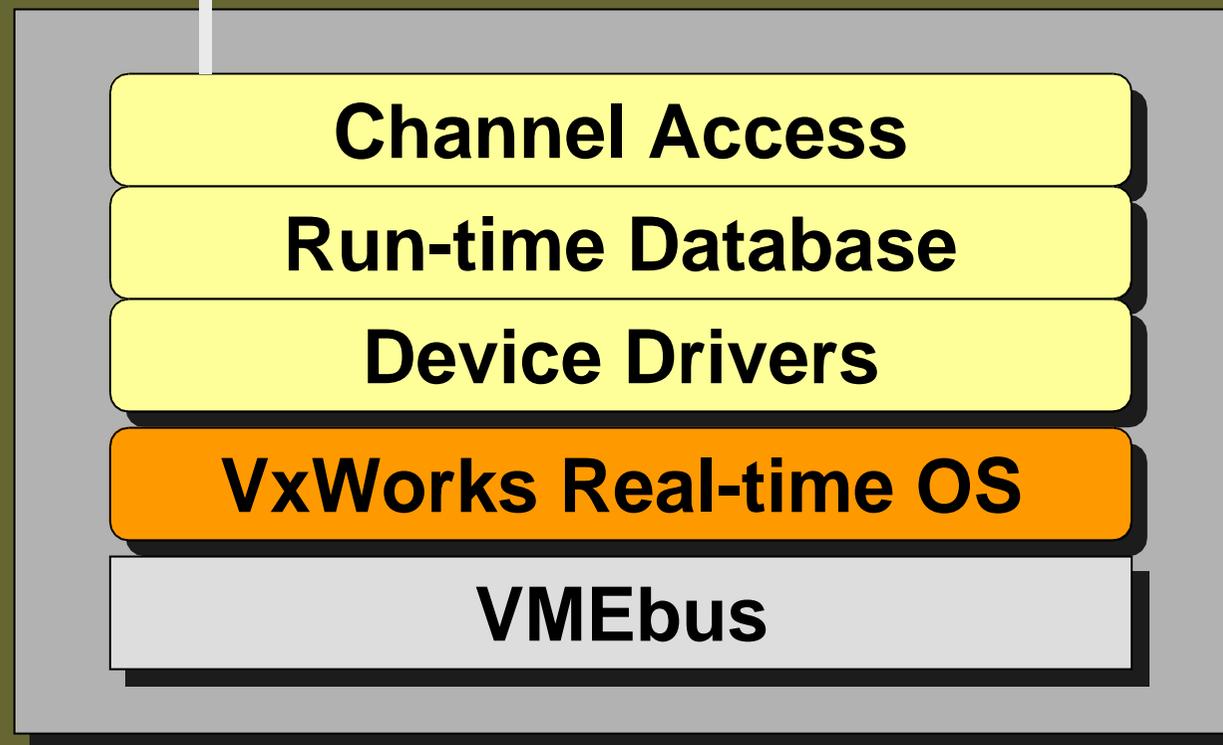
Input / Output Controllers (IOCs)

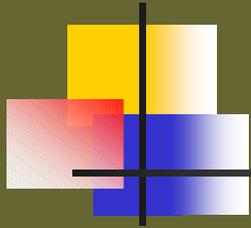
Field Buses (CAMAC, GPIB, etc.)

IOC ソフトウェア (iocCore)



Network

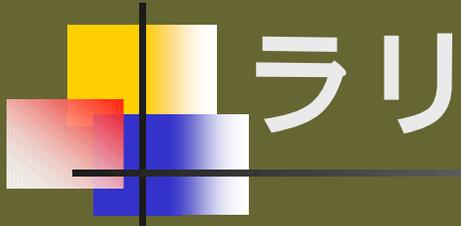




背景

- EPICS iocCore
 - VME 計算機 (VxWorks リアルタイム OS)
- EPICS 3.14 から Linux をサポート
 - POSIX 1003.1c (pthread)
 - POSIX 1003.1b (real-time extension)
- Linux にはリアルタイム応答性がない

OS インタフェース・ライブラリ



EPICS iocCore

OS Interface Lib

VxWorks, Linux, ...

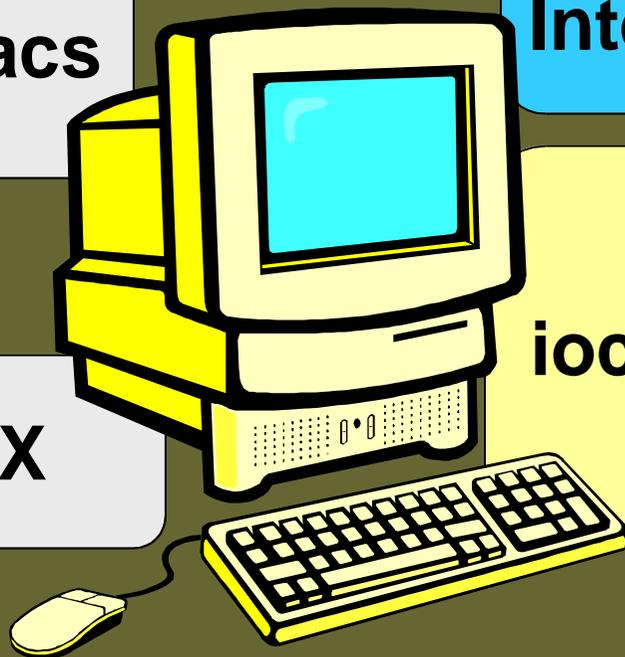
一台の PC 上で EPICS

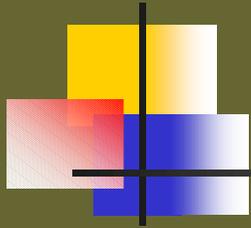
Emacs

Operator
Interface

iocCore

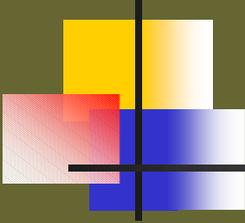
X





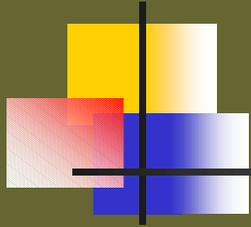
背景

- EPICS iocCore
 - VME 計算機 (VxWorks リアルタイム OS)
- EPICS 3.14 から Linux をサポート
 - POSIX 1003.1c (pthread)
 - POSIX 1003.1b (real-time extension)
- Linux にはリアルタイム応答性がない



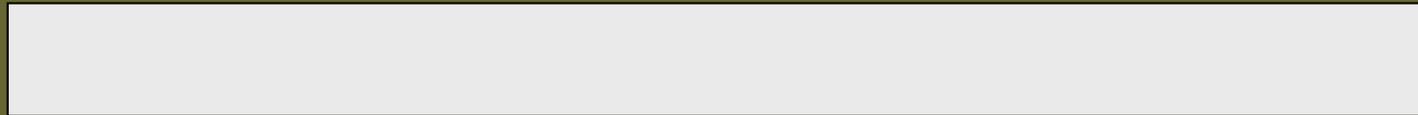
なぜリアルタイムでないか？

- カーネルがプリエンプティブでない
 - 数十ミリ秒
- 割込みの禁止
 - 数百マイクロ秒
- アドレス空間の切替
 - 数十マイクロ秒



予測不能な遅延

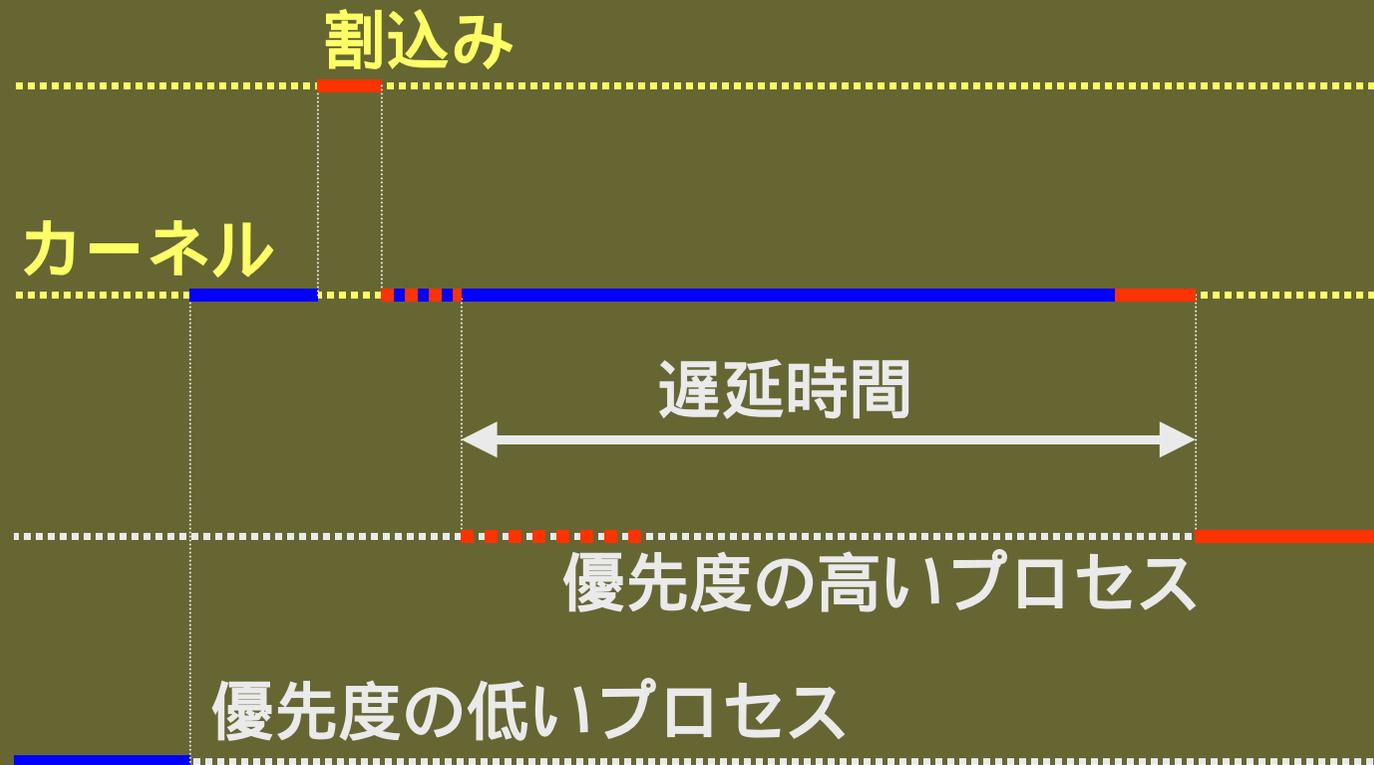
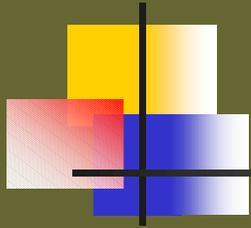
カーネルがプリエンプティブでない



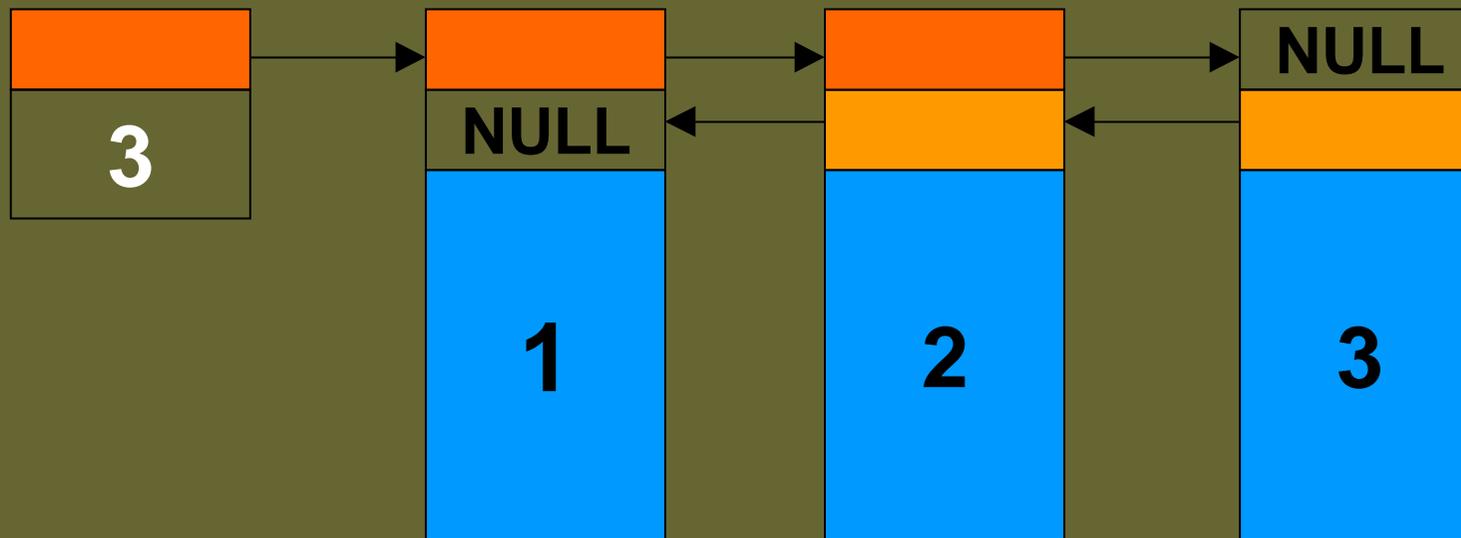
■ 割込みの禁止

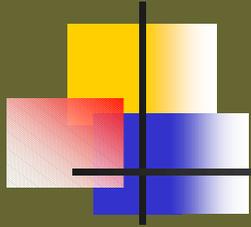
| アドレス空間の切替

プリエンプティブでないカー ネル



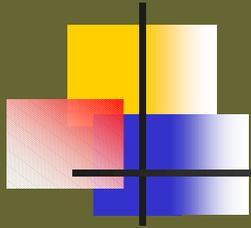
カーネル内での排他制御





排他制御の方法

- プロセスとプロセス
 - プリエンプションしない
- プロセスと割り込みハンドラ
 - 割り込み禁止
- 割り込みハンドラと割り込みハンドラ
 - 割り込み禁止



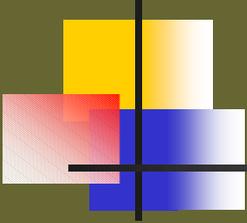
割込みの禁止

```
unsigned long flags;
```

```
save_flags(flags);  
cli();
```

```
/* critical section */
```

```
restore_flags(flags);
```

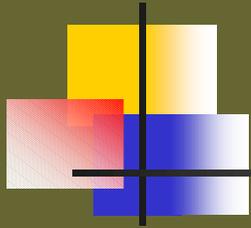


予測不能な遅延時間

プリエンプティブでない（**排他制御**）

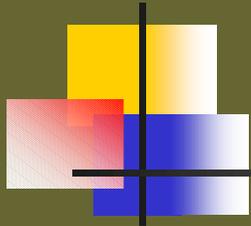
■ 割り込みの禁止（**排他制御**）

| アドレス空間の切替（**仮想記憶**）

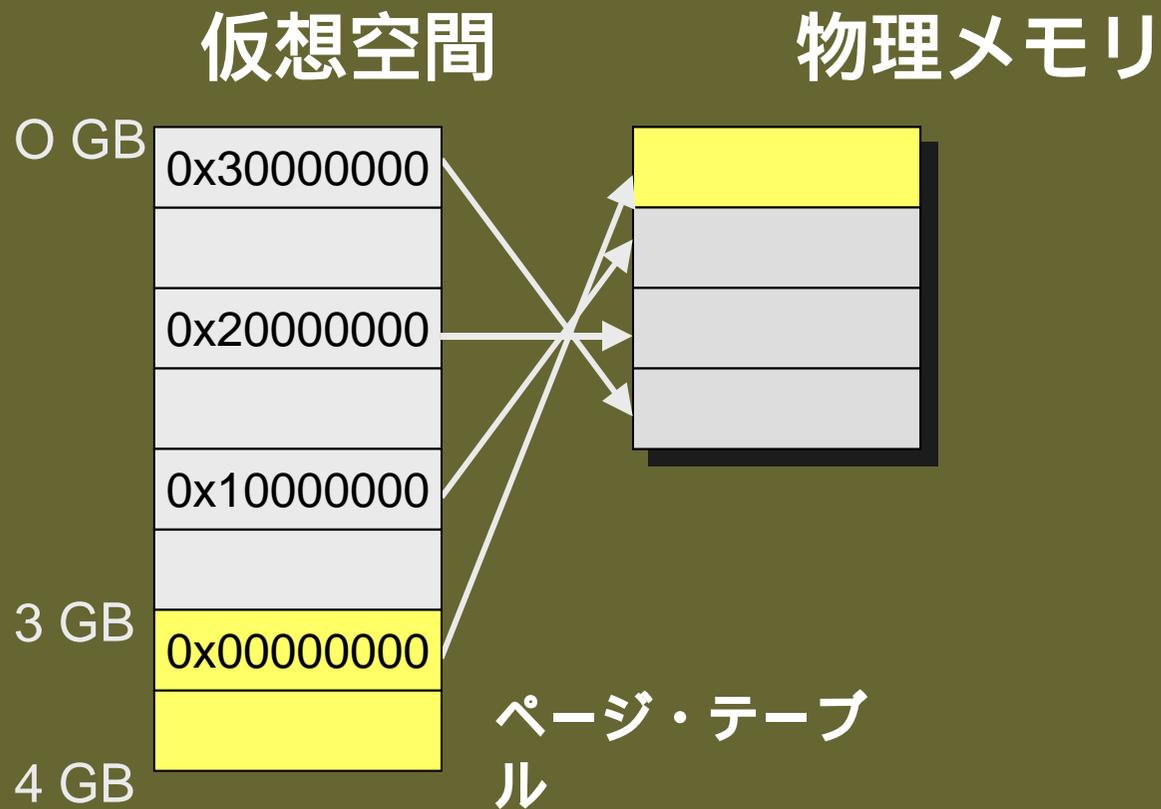


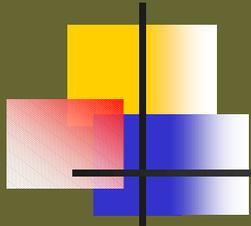
アドレス空間

- 多くのリアルタイム OS は全てをフラットな物理アドレスで実行
 - タスクはスレッドに対応
- Linux は仮想記憶をサポート
 - 仮想アドレスから物理アドレスへの変換
 - カーネル空間
 - ユーザ空間

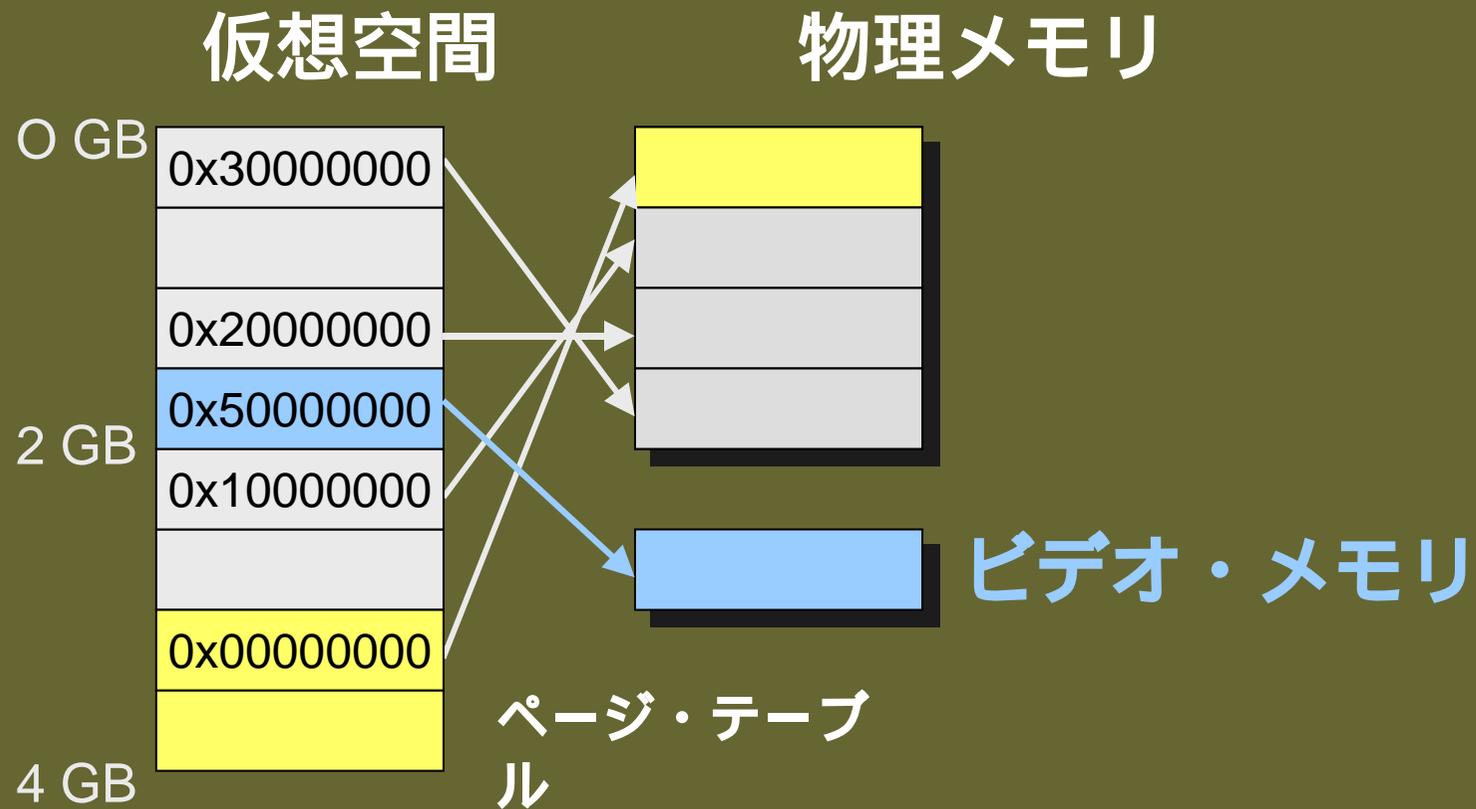


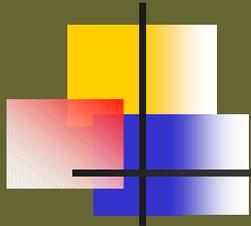
仮想空間とページ・テーブル



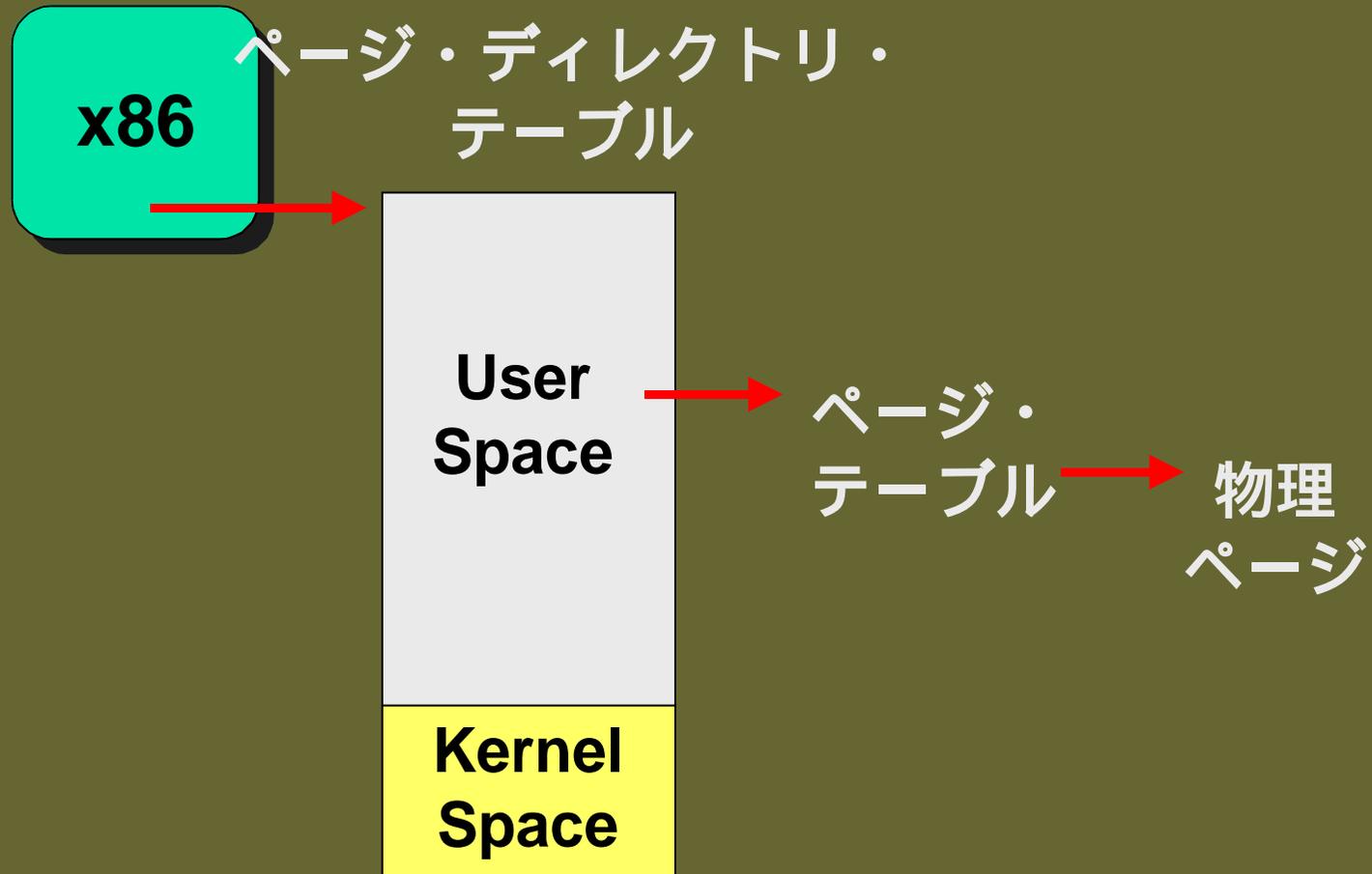


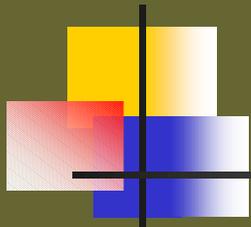
mmap() システム・コール



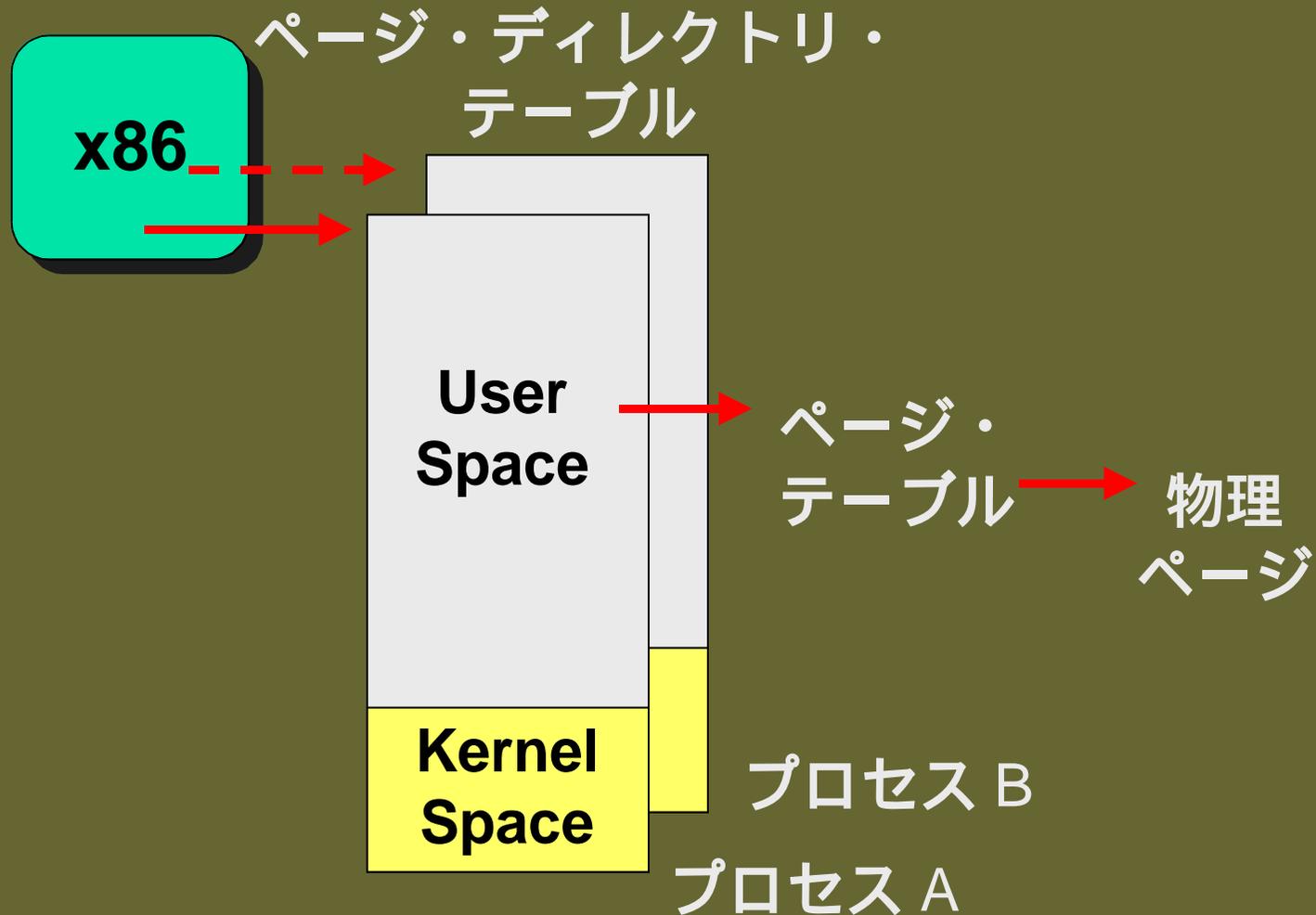


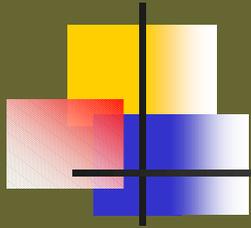
2レベルのページング





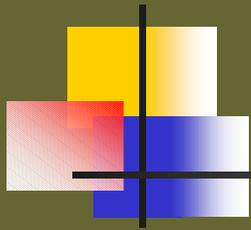
アドレス空間の切替



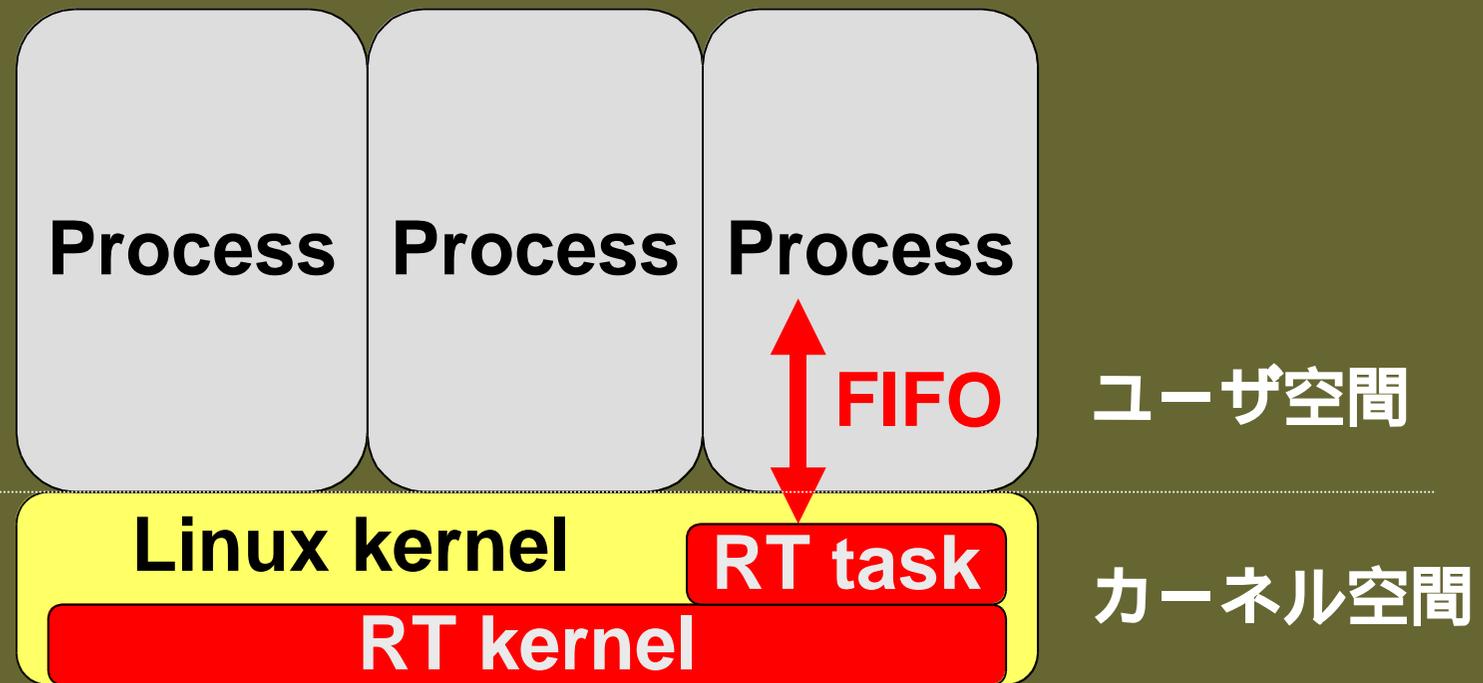


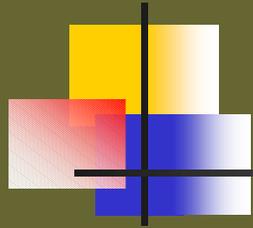
2つのアプローチ

- カーネル空間のリアルタイム・タスクを使う
 - RTLinux
 - リアルタイム・カーネルを Linux のロードダブル・モジュールとして組み込む
- ユーザ空間の Linux プロセスを使う
 - プリエンプティブな Linux カーネル
 - プリエンプティブに改造された Linux の下でのユーザ・プロセスを利用する



RTLinux のアーキテクチャ

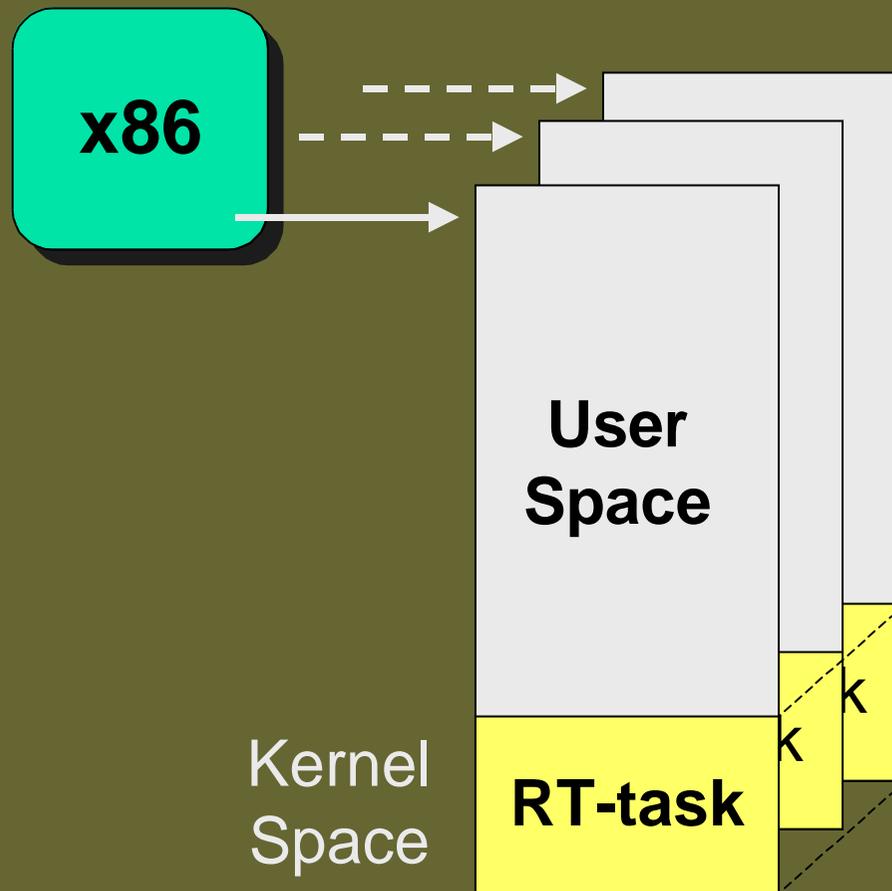




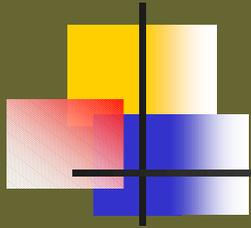
RTLinux

- リアルタイム応答性は抜群
- 3つの遅延要因の全てからフリー
 - RT タスクは Linux の機能を利用しない
 - Linux カーネルに割り込みを禁止させない
 - RT タスクの起動にアドレス空間の切替不要

RT タスクはカーネル空間に

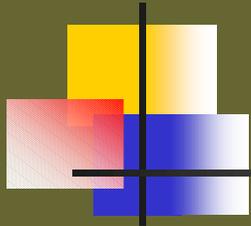


全てのプロセスの
アドレス空間には
カーネル空間が
マッピング済み



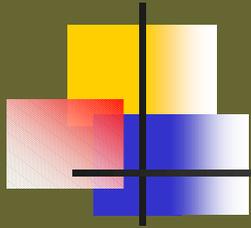
RTLinux の問題点

- RT タスクは Linux のシステム・コールを使えない（ポリシーとして使わない）
- リアルタイム・アプリケーションのバグから Linux カーネルが保護されない
- 高レベル（複雑）なアプリケーションには不向き



2つのアプローチ

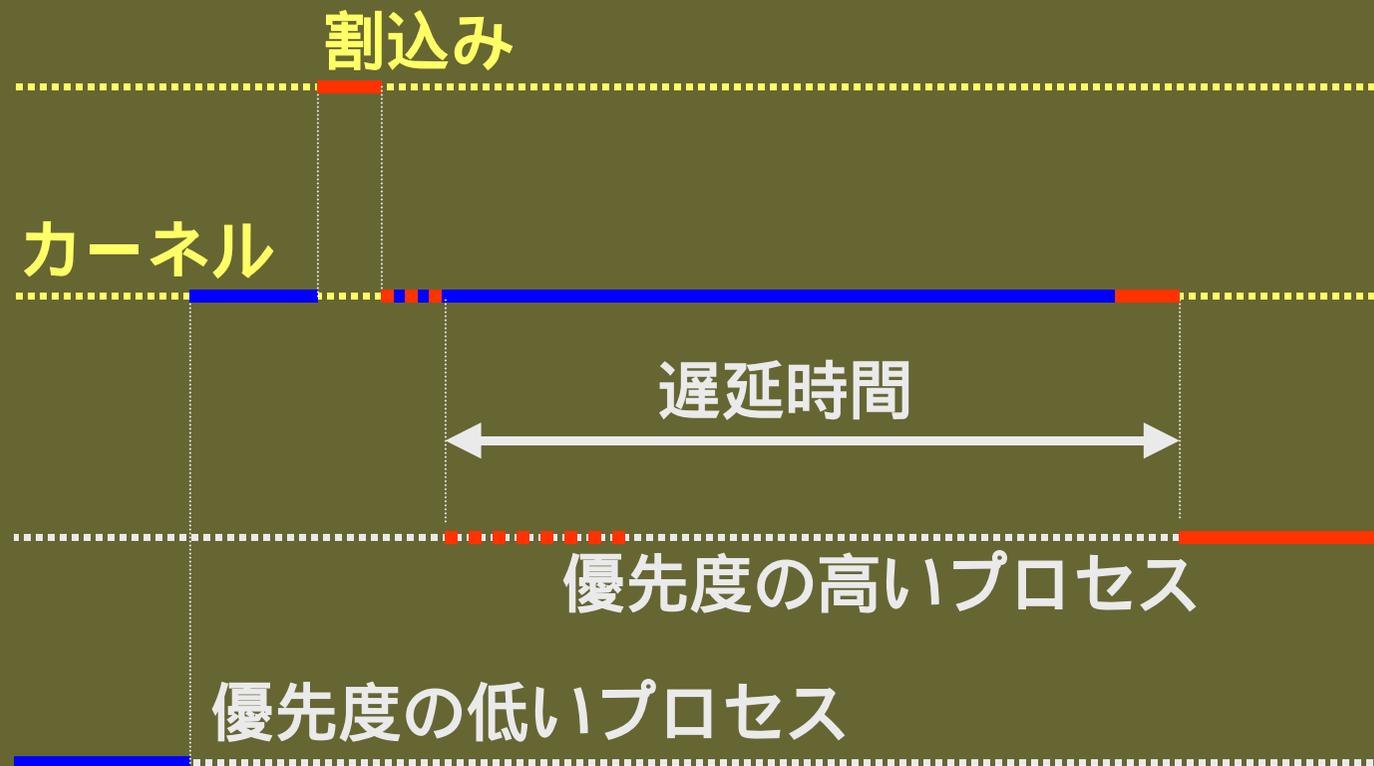
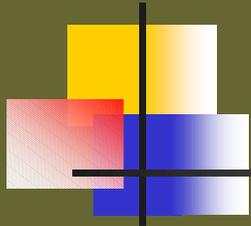
- カーネル空間のリアルタイム・タスクを使う
 - RTLinux
 - リアルタイム・カーネルを Linux のロードダブル・モジュールとして組み込む
- ユーザ空間の Linux プロセスを使う
 - プリエンプティブな Linux カーネル
 - プリエンプティブに改造された Linux の下でのユーザ・プロセスを利用する

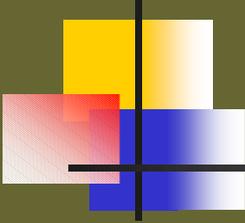


プリエンプティブな Linux

- SMP に対応した Linux カーネルには、複数の CPU に割当てられたプロセスの間での排他制御機構（スピン・ロック）が備わっている
- これをシングル・プロセッサ上で走る複数のプロセス間での排他制御に置き換えれば、プリエンプティブなカーネルができる

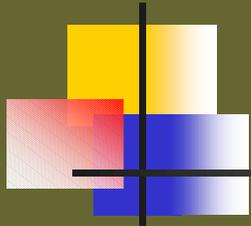
プリエンプティブでないカーネル





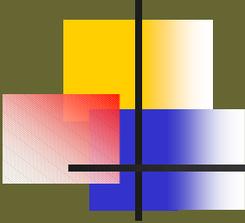
それで本当に話は済むの？

- SMP は本質的にスループットを上げる技術
- カーネル・コードの一部がスピン・ロックを「長持ち」しても、スルー・プットへの影響は少ない
- が、遅延の最悪値には直接的な影響がある
- そのような「問題児」がないか、カーネル全体を検査する必要がある



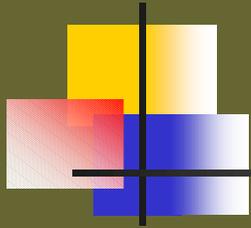
“ 第三の道 ”

- ユーザ空間で実行
 - Linux カーネルをアプリケーションから保護
 - Linux のシステム・コールを利用可能にする
- リアルタイム・カーネルでスケジューリング
 - リアルタイム・タスクの起動に Linux カーネル内の排他制御の問題を持ち込まない

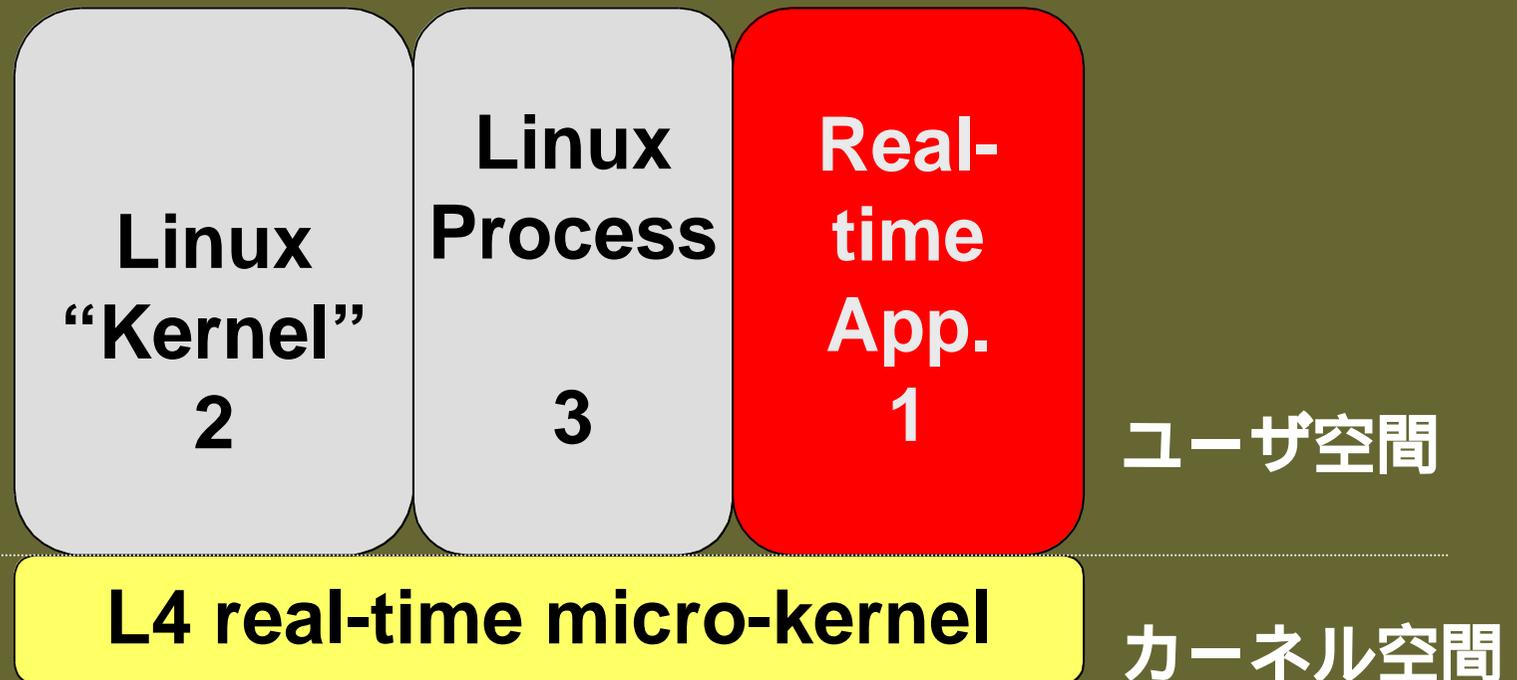


使えそうなもの

- L4-Linux
 - Linux カーネルをリアルタイム・マイクロカーネルの上にサーバ・タスクとして実現
 - Linux のプロセス自体はリアルタイムではない
- LXRT
 - RTAI (RTLinux から派生) のオプションな機能
 - ユーザ空間からカーネル内のリアルタイム API を呼べる

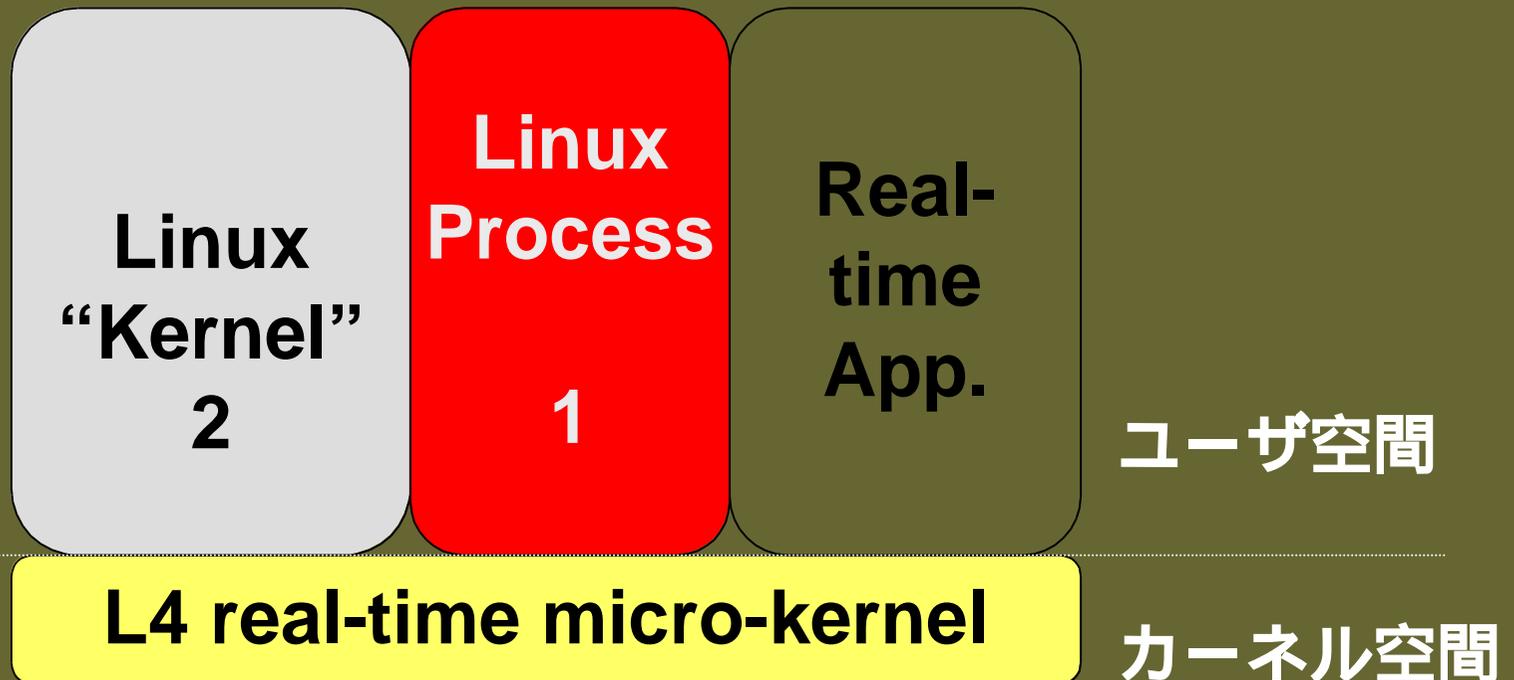
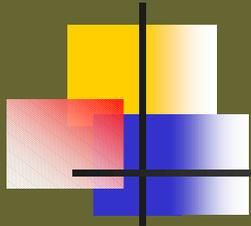


L4-Linux のアーキテクチャ



プロセスをリアルタイム処理

に



プロセスが“Linux”から横取

り

割込み

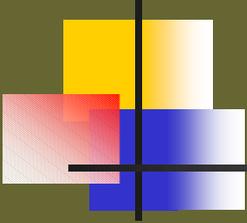
L4 カーネル

RT プロセス

Linux

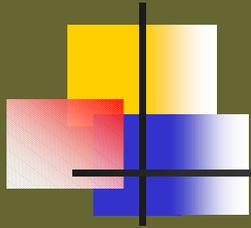
プロセス





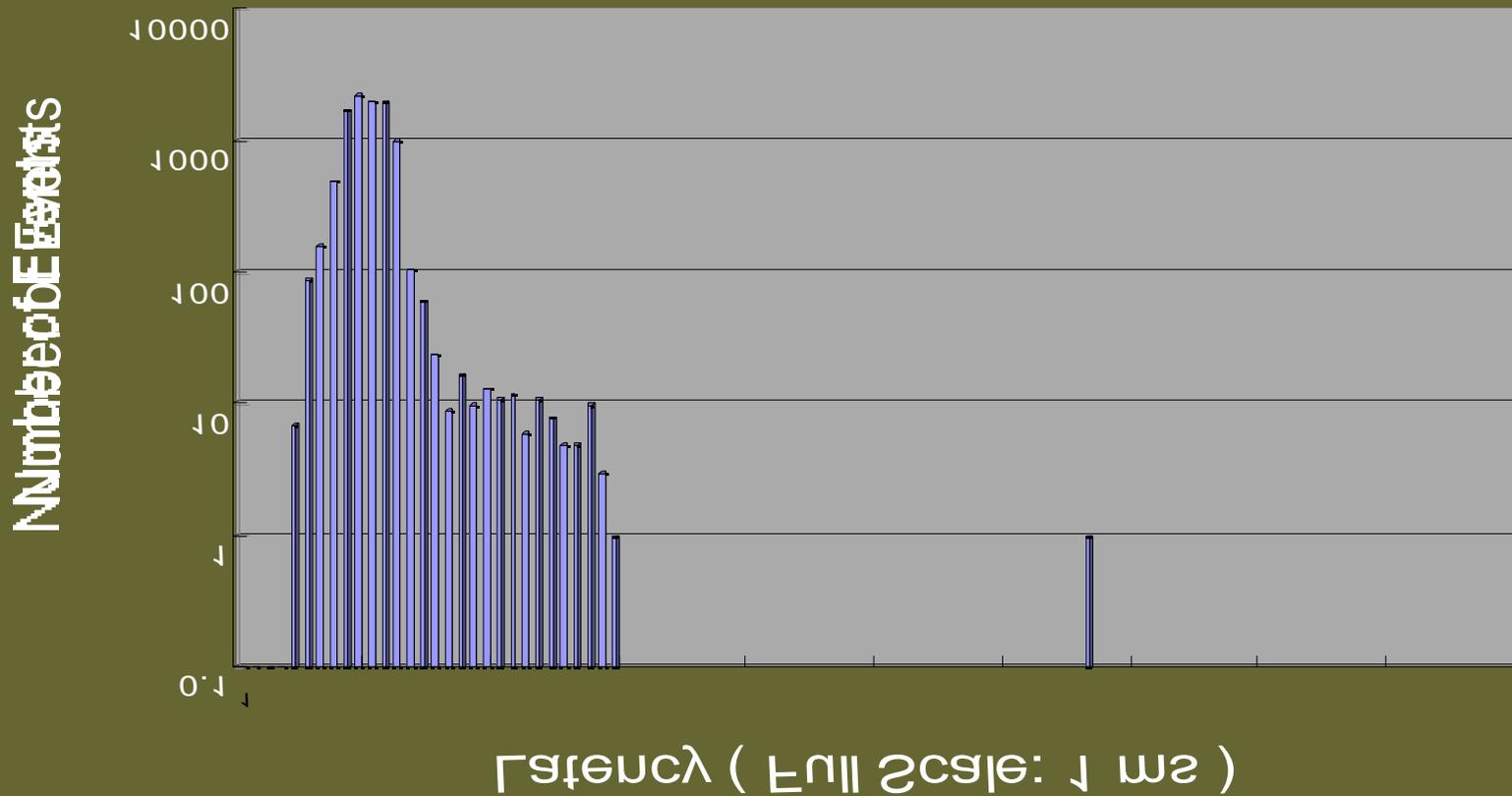
割り込みへの応答時間の測定

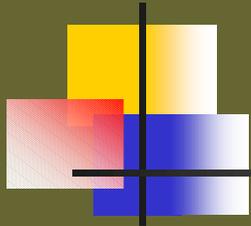
- VME CPU ボード (Celeron 300MHz)
- ディスク I/O をバックグラウンドで実行
- VME モジュールからの割り込みによりリアルタイム・スレッドを起動
- その遅延時間を繰り返し測定



遅延時間の測定結果

Thread Priority > Linux Server Priority





予測不能な遅延時間

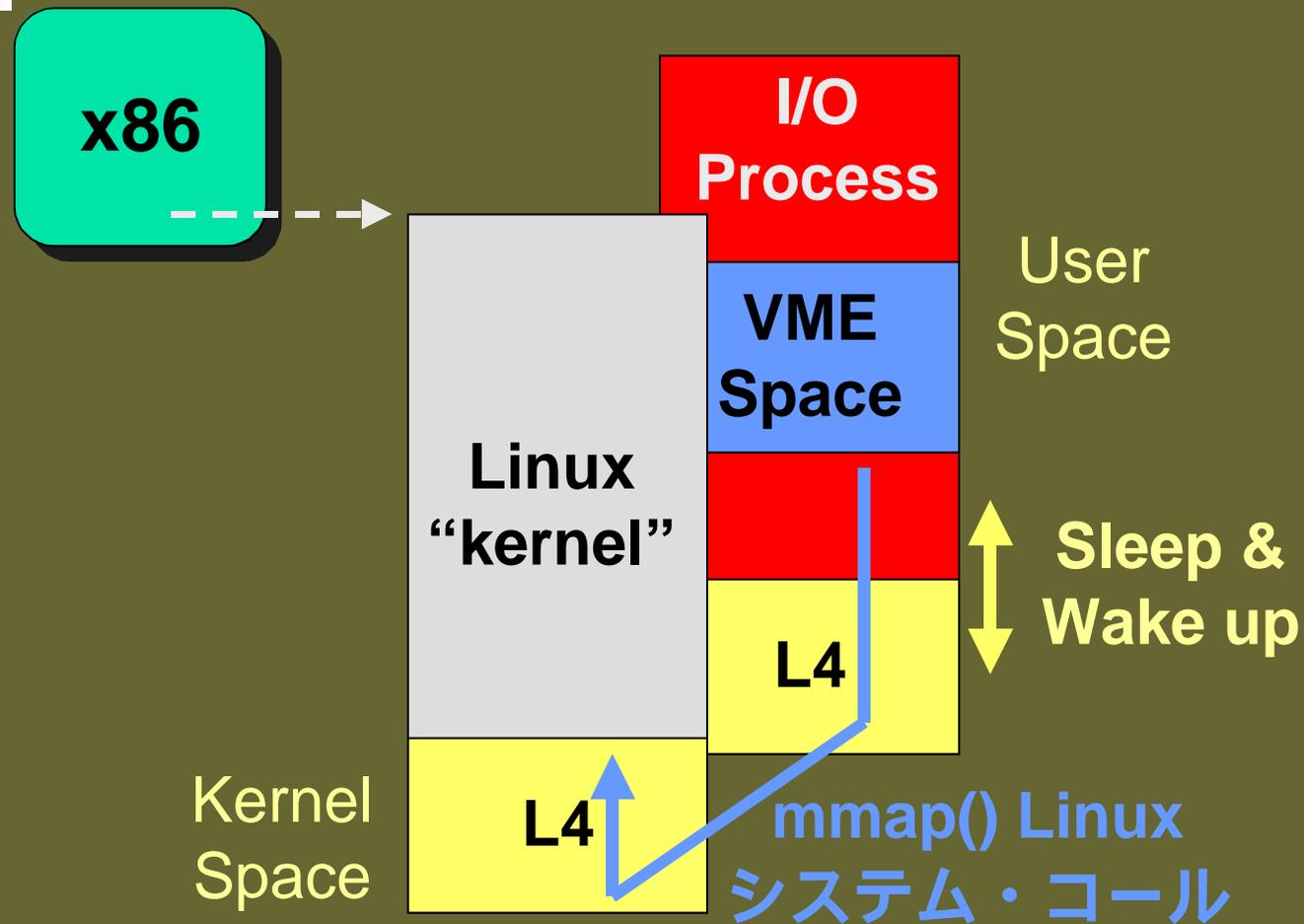
カーネル (L4) がプリエンプティブ

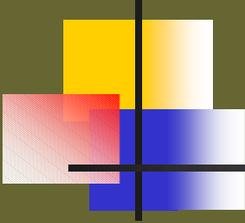


■ 割込みの禁止

| アドレス空間の切替

I/O ドライバ・プロセス



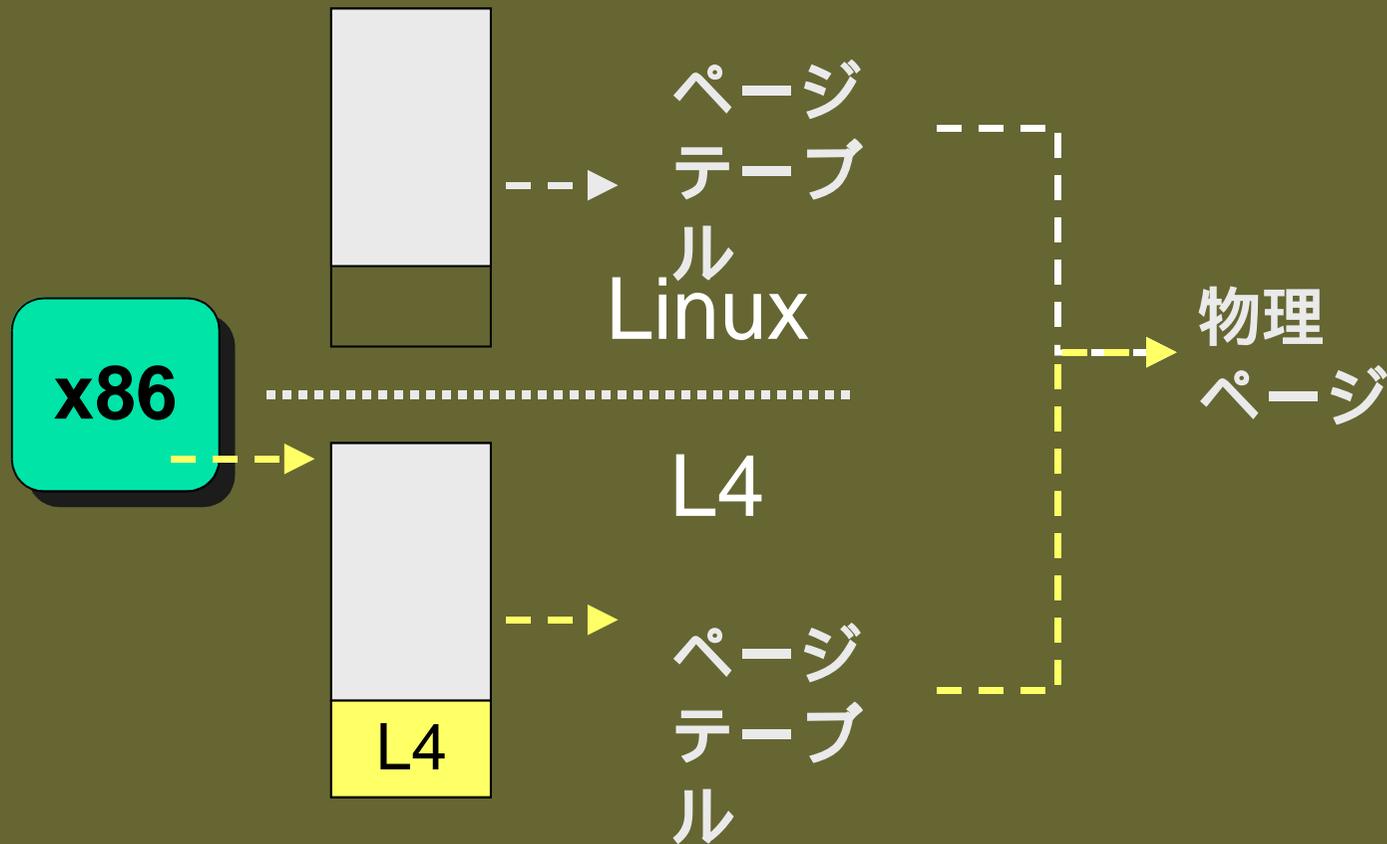


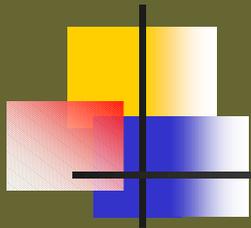
L4-Linux の問題点

- プロセスに高い優先度を与えるための変更は簡単
 - 変更を要したファイルは3つ
 - 追加したコードは数十行
- 仮想記憶機構の扱いが特殊
 - 素朴な疑問
 - そもそも、ページ・ディレクトリやページ・テーブルはどこにある？

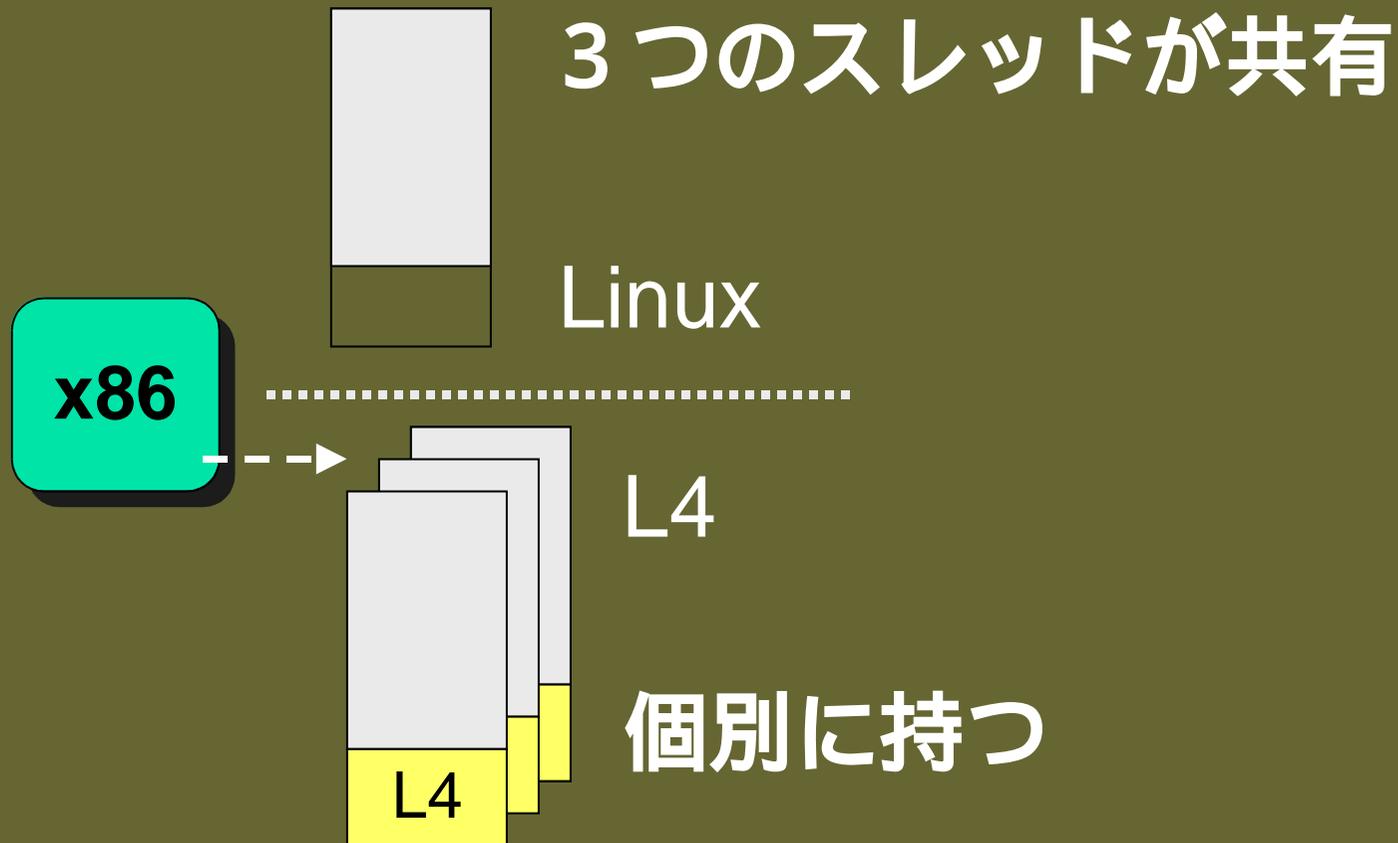
1つのプロセスに2セットあ

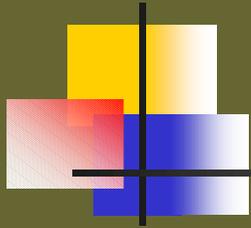
る



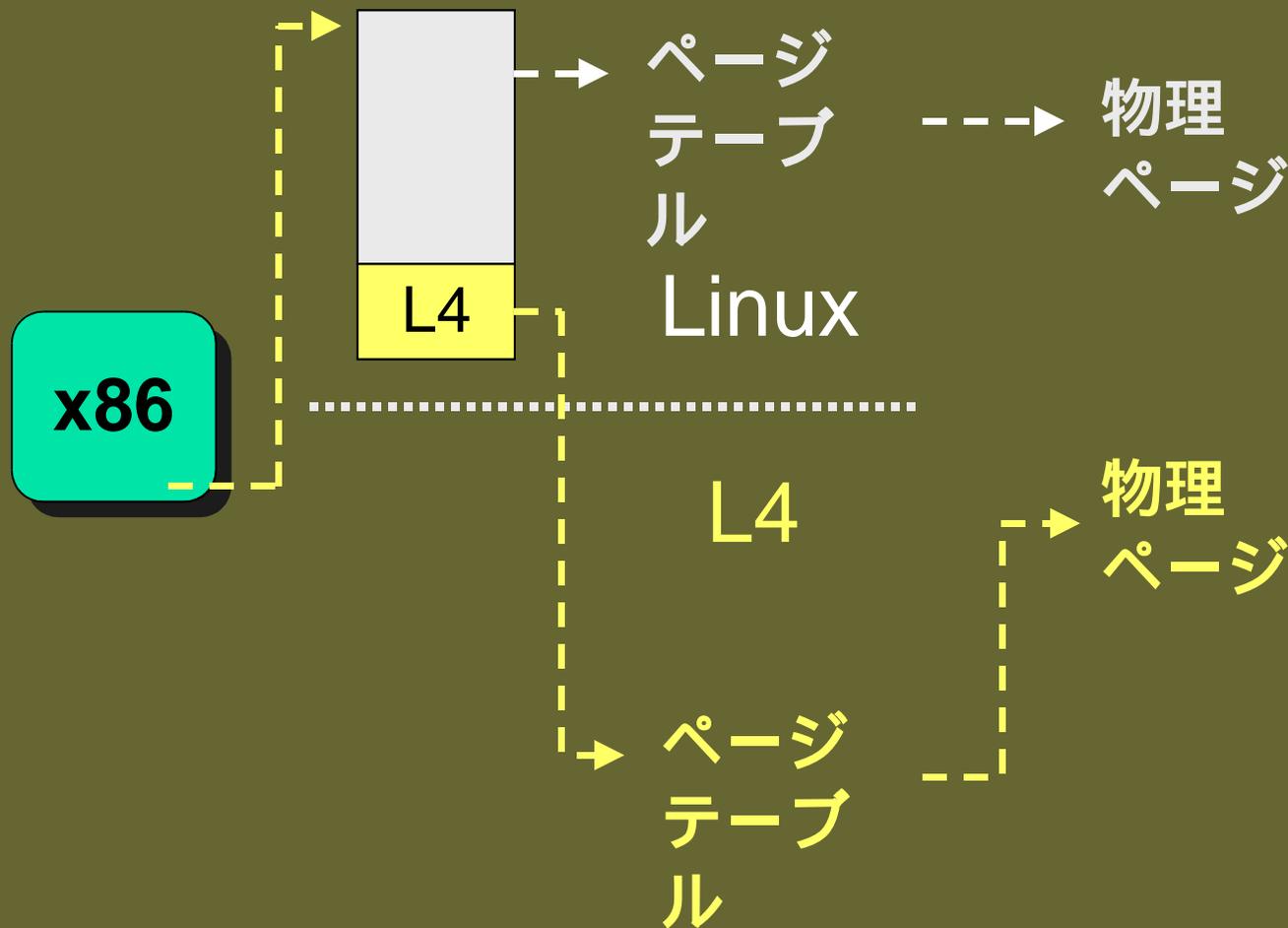


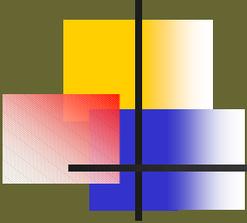
スレッドを生成した場合





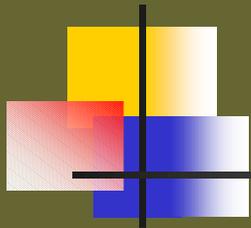
できれば一元化したいが



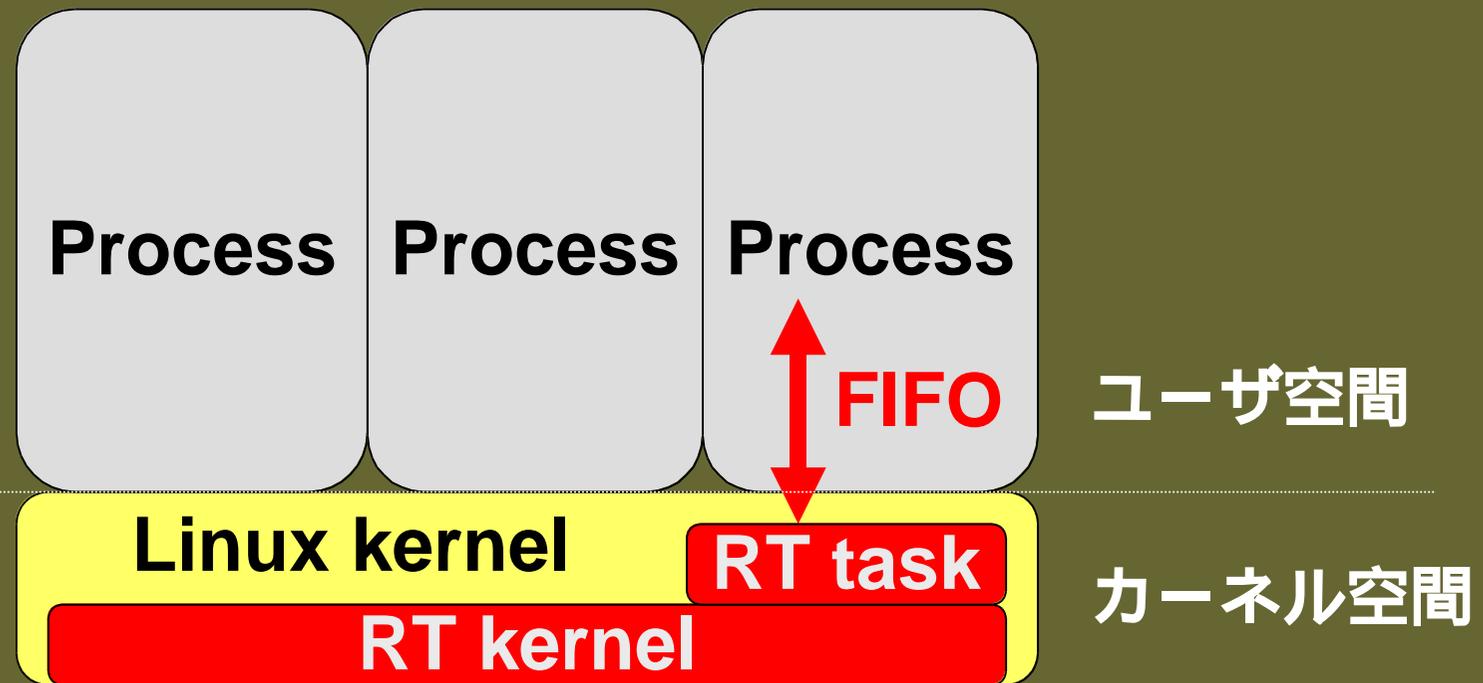


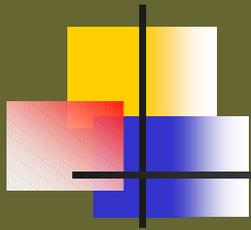
大きく改造すると

- 派生品の派生品になる
 - メンテナンスが大変
 - 「市場」の信頼を得ることが困難
 - カレントからの後れが大きくなる
 - EPICS も Linux も猛烈な勢いで進化している
- もう1つの可能性
 - LXRT (RTAI のオプション)
 - カーネルに手を加える必要はない

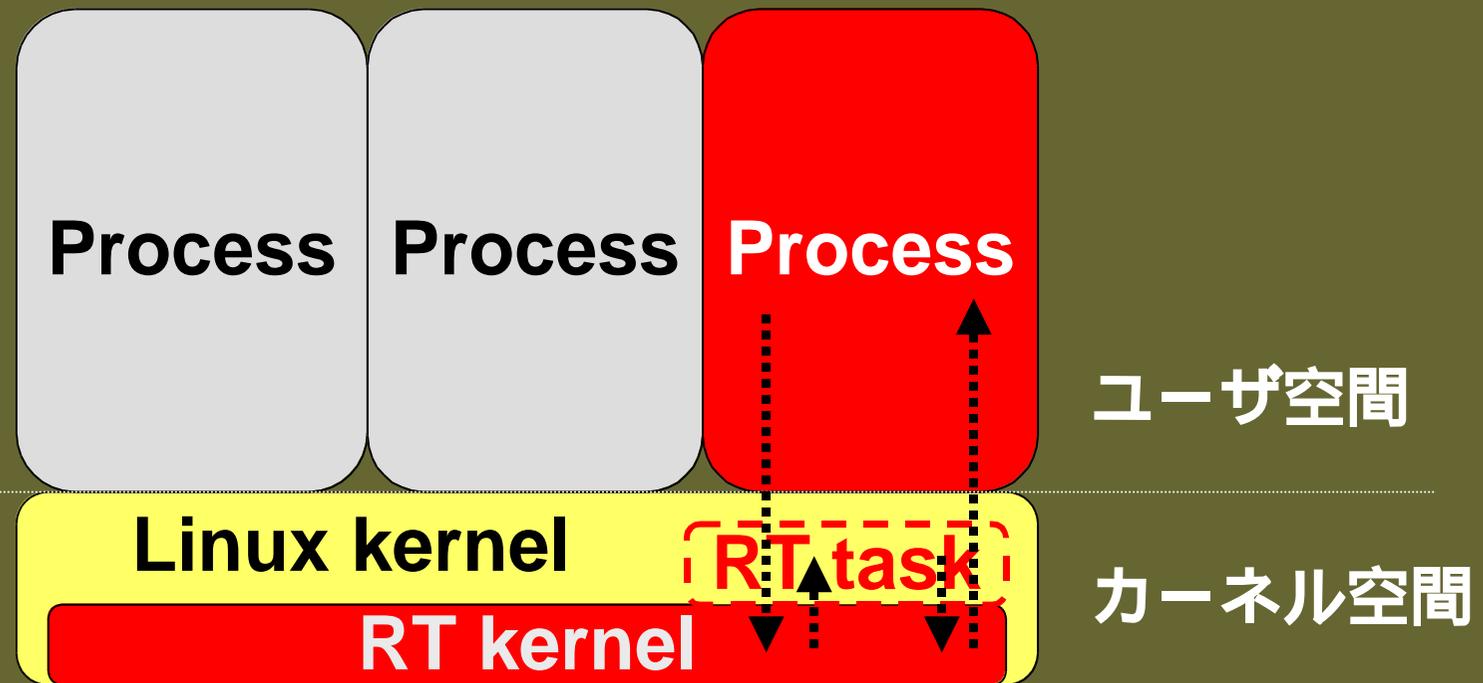


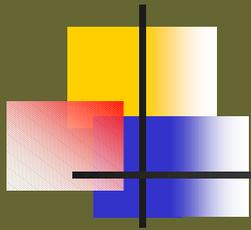
RTAI のアーキテクチャ



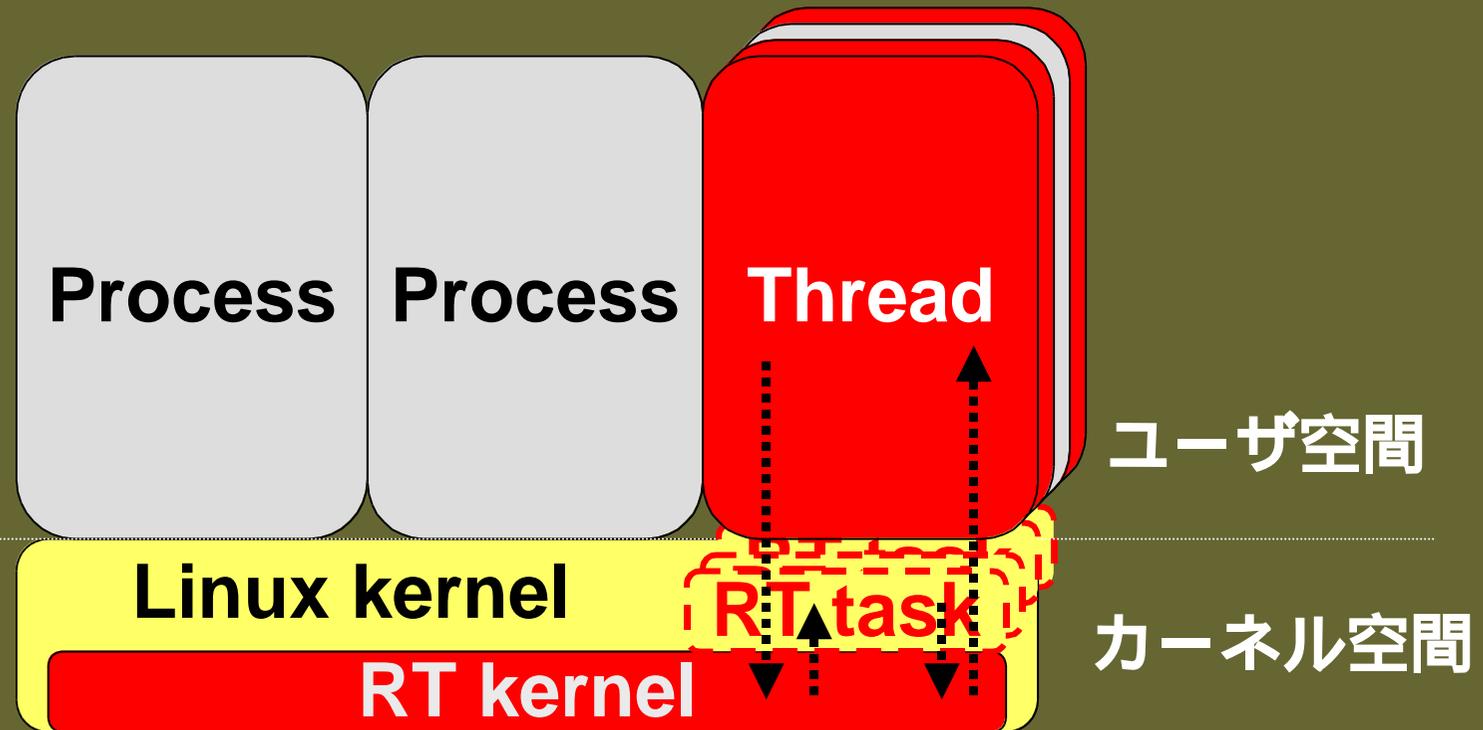


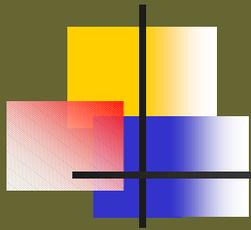
LXRT (RTAI のオプション)



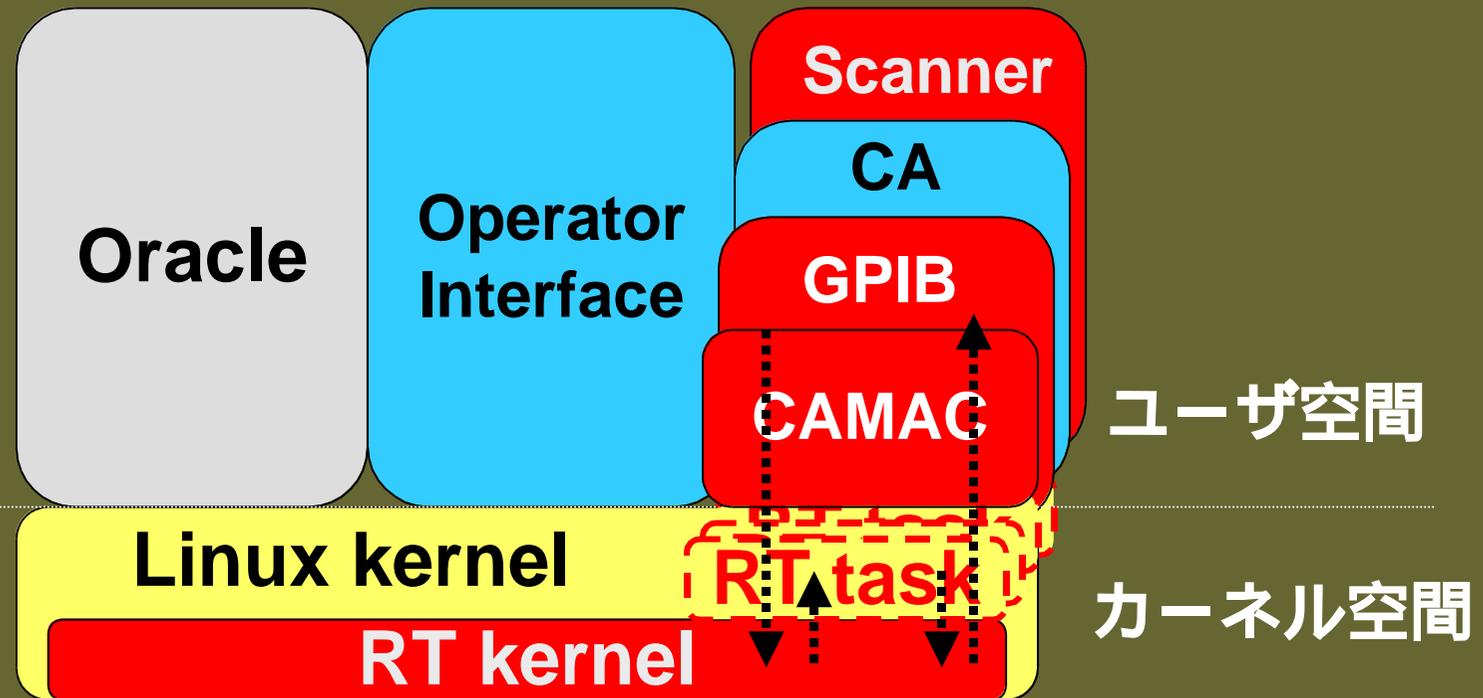


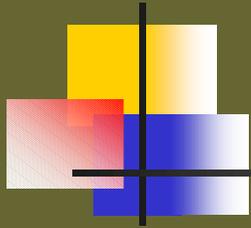
スレッドを使って役割分担





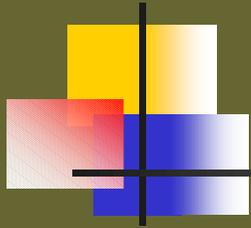
EPICS on Real-time Linux





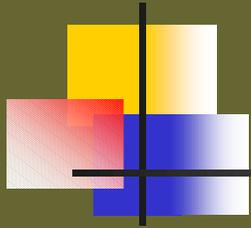
まとめ (1)

- EPICS iocCore の移植のための要件
 - マルチ・スレッドをサポートすること
 - スレッドはユーザ空間で走ること
 - スレッドはリアルタイムであること
- Linux のスレッドにはリアルタイム性がない
 - プリエンプティブでない
 - 割込みの禁止
 - アドレス空間の切替



まとめ (2)

- RTLinux (RTAI) は複雑なアプリケーションには不向き
- プリエンプティブなカーネルはソフト・リアルタイムのレベルを超えるのは困難
- あくまでもユーザ空間の Linux プロセスでハード・リアルタイムを目指す
 - L4-Linux : 大掛かりな改造が必要
 - RTAI-LXRT : 期待が持てそう(?)



謝 辞

- L4-Linux 上で VME バスにアクセスするためのライブラリを開発するに当り、安 芳次氏には数々の有益なアドバイスを頂きました。
- また、仲吉 一男 氏には、Linux のリアルタイム応答性に関する諸情報、およびその測定方法について色々とお教え頂きました。
- 両氏のご協力に感謝いたします。