

連携システムのためのソフトウェアバス(分散OS) “LP49”

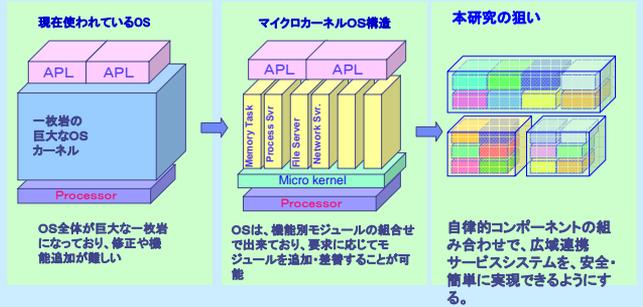
丸山勝巳・佐藤好秀(日立製作所)・児玉和也・日高宗一郎

問題認識

- ◆ ソフトウェアは、人間の管理能力を超えて巨大化・複雑化。
 - ◆ ソフトウェアバグは、決して無くならない。
 - ◆ ソフトウェアの信頼性・頑強性の向上は、ソフトウェアの最大課題。
- ↓
- ◆ 障害が生じて、そのコンポーネントだけを停止・再立ち上げできるコンポーネント構成が肝要。
 - ◆ コンポーネント構成をOSで実証することに意義。
 - ◆ ソフトウェアの進歩には、ソースコードの公開が重要。

防護された部品によるシステム構築

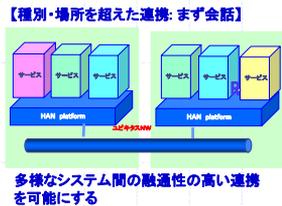
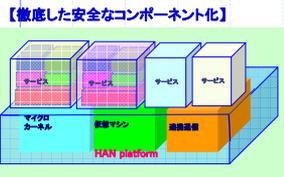
シンプル化こそ、ソフトウェアの信頼性・頑強性強化の王道



研究の必要性と方針

必要性

- 各種システム間の連携動作が必要 → 初めに会話ありき
- システムの巨大化 → 保護されたモジュール・コンポーネント化の徹底
- パソコン以外(組込み系、通信系、ロボット、社会基盤システム等)が益々重要になる → カスタマイズ化、拡張性、実時間性が必要
- 高信頼、頑強性、安心・安全への希求 → 検証可能なシステム
- 現用OSは複雑巨大化しすぎ → いずれ揺り戻しが来よう。

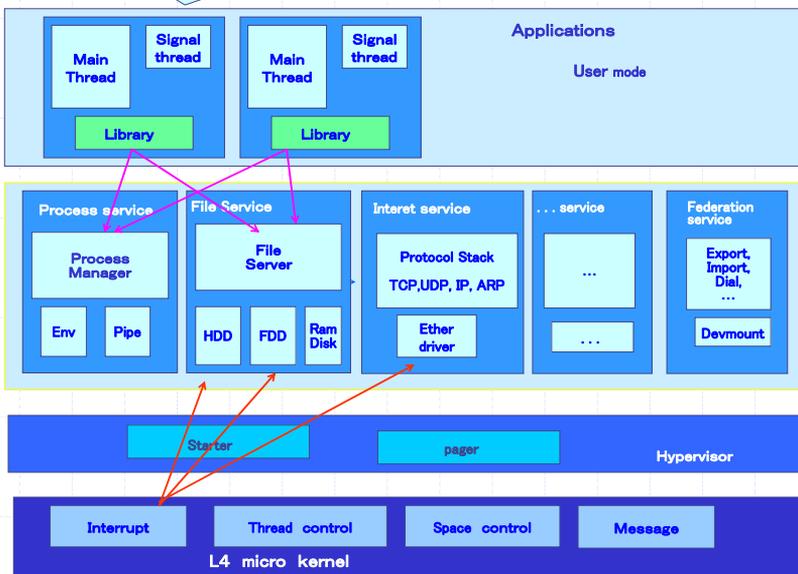


研究方針

- ◆ 制御システムやサーバーに適したシンプルなOSを作り、ソース公開。
・制御システムには、数千万行の太ったOSは不要。
- ◆ 障害に対する頑強化 ⇒ マイクロカーネル+マルチサーバ構成。
- ◆ 拡張性 ⇒ 単機能サーバの自在な追加、置き換え。
- ◆ システム連携 ⇒ 名前空間(リソース管理簿)のexport/importによる融通連携。
- ◆ プログラム開発の容易化 ⇒ サーバは勿論、ドライバもユーザー空間で実行。
- ◆ モノリシックからマルチサーバへ ⇒ 少しのオーバーヘッドで大きな安心。
・マルチサーバ型OSは、まだまだ研究課題が多い。

研究内容(連携OS: LP49)

- マイクロカーネル機能を使って、ハードウェア制御も含め、各サービスをユーザーモードで実現
- OSサービスを提供するサーバーも、普通のAPLも、同じユーザーモードプログラム。



○ ソースコードと(詳しい)資料はWEBで公開中
<http://research.nii.ac.jp/H20/LP49>

- 手頃なサイズ(ソースで100K行程度)
- 分散OSの学習素材としても、効果的。
- 自前の制御用OSの開発ベースとしても便利。

【Universal Federation】

- リソース類の広域マウント
- リソースの広域連携
- 9Pベースのプロトコル

9P (QP) Protocol

【マイクロカーネル以外は、すべて保護されたユーザーモードで実行される】

- User mode
- 障害を波及させない
 - プログラム開発・デバッグの容易化
 - 拡張性強化
 - ドライバプログラムも、ユーザーレベルで実行

【ハイパーマネジャー】

- User mode
- 論理メモリの制御
 - ユーザーレベルで多様なメモリ操作
 - メモリ状態のスナップショット
 - 処理のやり戻し
 - 過去の再現

【L4 マイクロカーネル】

- Kernel mode
- 論理空間
 - ユーザーが仮想メモリを定義可能
 - 効率のよいスレッド
 - 高性能メッセージ通信

