Operating Systems & Compiler IA2

Embedded Real-Time Operating Systems based on Smart Chips

By Ketaki Mahajan (16014022050)

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

What is this Research About?

- This research paper by **Xiaofeng Shang** looks at how to create **special operating systems** for **small computers**.
- The focus is on making tiny computers work **efficiently** using **smart chips**.
- The goal is to make these small systems better by **improving** how they **handle tasks**.

What are Embedded Systems?

- **Embedded Systems** are small computers built into other devices - like the computer inside your microwave or car.

- Key features include:
  - Made for **specific purposes** (not general use like your laptop)
  - Must be **reliable** (can't crash while controlling medical equipment)
  - Need to be **small** and use **little power**
  - Should be **low-cost** for mass production

Examples:
Smart watches, digital cameras, car systems, industrial controllers

How Were These Systems Built Before?

- Old Way:
  - First, engineers would **build** all the **hardware.**
  - Then, programmers would **write software** for it.
  - This took a **long time** and was **expensive**.
  - If something needed to change, they often had to **start over**.

Problems:

- Hardware and software teams worked separately
- Difficult to make changes later
- Took too long to finish products
- Hard to fix mistakes

The New Way: Smart Chip Design

- **What's different now?**
  - Engineers work on hardware AND software at the **same time**.
  - They use **special chips** called **FPGAs** that can be **reprogrammed**.
  - These chips can do **many functions** that used to require multiple chips.

- **System-On-Programmable-Chip** (SOPC System):
  - **Everything is on one chip -** processor, memory, and special functions.
  - Can be **redesigned** without **changing physical parts**.
  - Makes devices **smaller** and **cheaper**.

What is a Micro-Kernel?

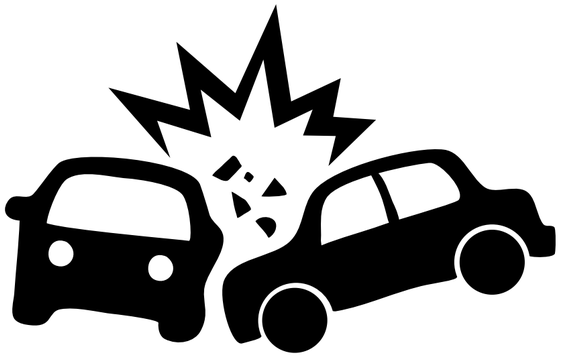
The **micro-kernel** is like the **essential core** of the operating system. It only keeps the most **important functions inside** it. Everything else is **moved outside** to **save space** and **improve speed**.

How it works?

- Handles basic task **switching** (letting different programs take turns).
- Manages **communication** between tasks
- **Controls interrupts** (urgent signals that need immediate attention).

Real-Time Operating Systems Explained

- **What makes it “Real-Time”?**
  - It must **respond** to events within **strict time limits**.
  - If a sensor **detects danger**, the system must **react immediately**.
  - Regular operating systems (like **Windows**) don't guarantee **quick responses**.



Example: In a car's airbag system, even a tiny delay could be dangerous

How the System Manages Task?

Task Scheduling

- The system decides which task runs when.
- Important tasks get priority over less important ones.
- Each task gets its own special area in memory called a "stack".

Communication Between Tasks

- Tasks need to share information and coordinate their work.
- They use three main ways to communicate:
  - Semaphores
  - Message Queues
  - Mailboxes

Memory Management

- System can give memory to tasks when they need it.
- When tasks are done, they give the memory back.
- This helps save limited resources.

System Optimizations for Better Performance

- **Network Communication:**
  - **Problem:** Standard protocols use too **many resources**.
  - **Solution: Simplified protocols** with **unnecessary steps removed**.
  - Small messages handled **more efficiently** through combined function modules.

- **System Timing Management:**
  - Regular timer signals (10-100ms) create system "heartbeat".
  - **Coordinates** when tasks run and tracks execution times.
  - Ensures time-sensitive operations happen **precisely** when needed.

Benefits:
- Faster overall system response
- Reduced memory footprint
- Lower power consumption
- More reliable timing for critical tasks

Research Outcomes & Why It Matters

- **Key Achievements:**
  - **Successfully implemented** entire **system** on a **single smart chip**.
  - Built a real-time OS that handles **multiple time-sensitive tasks**.
  - Developed **reliable application** software for real-world conditions.

- **Benefits:**
  - More **compact**
  - More **efficient**
  - More **affordable**
  - **Widely applicable** - medical devices, home appliances etc.

Major Impact: Extended CPU life and improved overall performance

Thank you!

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST